Overview

Notation

Matrices are written in uppercase: A, vectors are written in lowercase: a. A_{ij} denotes the element of A in position (i, j), A_i denotes the ith column of A (it's a vector!) and a_i denotes the ith entry of a.

1 Sparsity

In the following sections we describe some applications of sparse models in signal and image processing, statistics and inverse problems. The goal is to provide a high-level picture of how to leverage sparsity assumptions in these domains.

1.1 Denoising via thresholding

In data processing, data are often described as a combination between a **signal** component, which contains the information that we are interested in, and a **noise** component, which accounts for perturbations that corrupt the signal. The problem of **denoising** data consists of teasing apart the signal and the noise, using prior information about their structure.

Let us consider an additive noise model

$$\text{data} = \text{signal} + \text{noise}. \tag{1}$$

If we expect the signal to be sparse, then an intuitive denoising method is to set to zero any entry that is below a certain value. This is known as **hard thresholding** (as opposed to soft thresholding, which we will learn about later in the course).

Definition 1.1 (Hard thresholding)**.** *Let $x \in \mathbb{R}^n$. The hard-thresholding operator $\mathcal{H}_\eta : \mathbb{R}^n \to \mathbb{R}^n$ sets to zero any entries in x with magnitude smaller than a predefined real-valued threshold $\eta > 0$,*

$$\mathcal{H}_\eta(x)_i := \begin{cases} x_i & \text{if } |x_i| > \eta, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Figure 1 shows an example in which a sparse signal is denoised by applying hard thresholding.

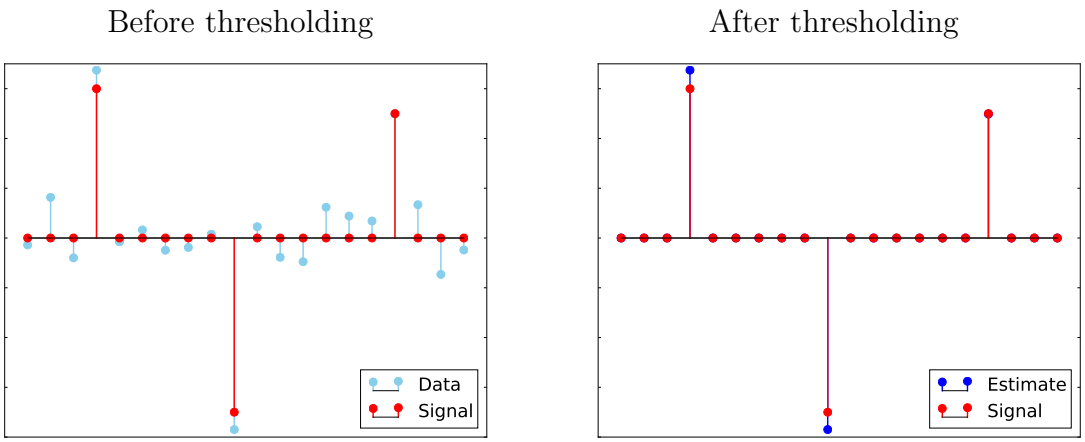


Figure 1: Denoising via hard thresholding.

1.2 Sparsity in a basis

Even though signals of interest in applications are often not sparse at all, they are often well modeled as a linear combination of a small number of predefined **atoms**. This can be conveniently expressed mathematically by grouping the atoms as columns of a certain matrix D, which is often known as a **dictionary**. Recall that for any matrix $D \in \mathbb{R}^{m \times n}$ and any vector $c \in \mathbb{R}^n$, the matrix-vector product

$$Dc = \sum_{i=1}^{n} D_i \, c_i, \tag{3}$$

is a linear combination of the columns of D weighted by the entries in c. If the signal $x = Dc$ can be represented by a combination of a few atoms in the dictionary, then its corresponding coefficient vector c will be sparse.

Sinusoids are an important example of atoms that allow to obtain sparse representations. Figure 2 shows an example of a signal that is sparse in such a dictionary. Another very popular sparsifying transform in signal and image processing is the **wavelet** transform. Figure 3 shows a natural image and its corresponding wavelet coefficients, most of which are extremely small.[1]

If the atoms form a basis of the ambient space, then the coefficients $c \in \mathbb{R}^n$ in the representation of x,

$$x = Bc, \tag{4}$$

[1]The numerical simulations in Figures 3 and 5 were implemented using Gabriel Peyré's Wavelet Toolbox which is available online at http://www.ceremade.dauphine.fr/~peyre/matlab/wavelets/content.html

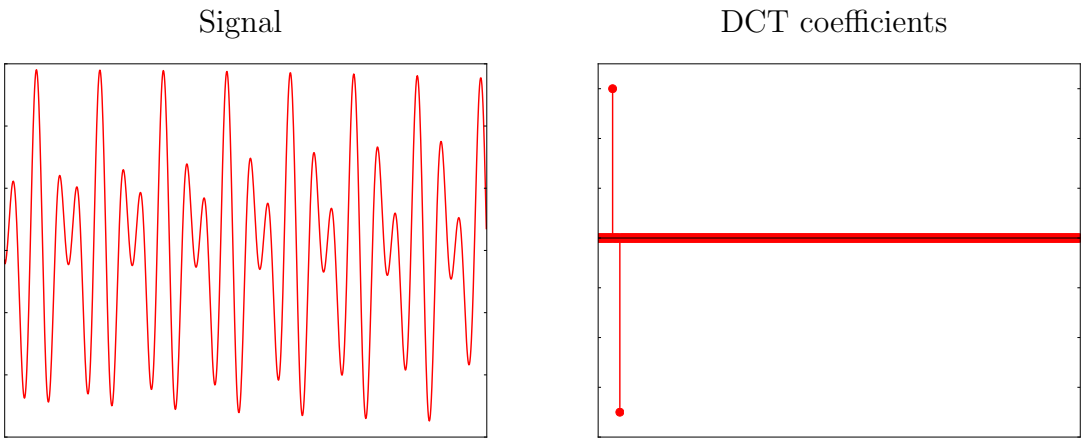


Figure 2: The signal on the left is sparse after applying a discrete cosine transform (DCT), which expresses the signal in terms of cosine atoms. The corresponding coefficients are shown on the right.

can be retrieved by applying the inverse of the dictionary B. This is the case for both examples in Figures 2 and 3.

If we consider data corrupted with additive noise,

$$y = Bc + z, \tag{5}$$

it is straightforward to exploit the sparsity assumption through thresholding. We only need to apply the inverse of the dictionary and then threshold. The estimated coefficients are of the form,

$$\hat{c} = \mathcal{H}_\eta \left(B^{-1} y \right), \tag{6}$$

which yields the signal estimate

$$\hat{y} = B\hat{c} = B\, \mathcal{H}_\eta \left(B^{-1} y \right). \tag{7}$$

Figures 4 and 5 show the results of applying this denoising method to the signals in Figures 2 and 3. Both signals are corrupted by additive Gaussian noise. In both cases, exploiting the sparse decomposition allows us to denoise the data very effectively.

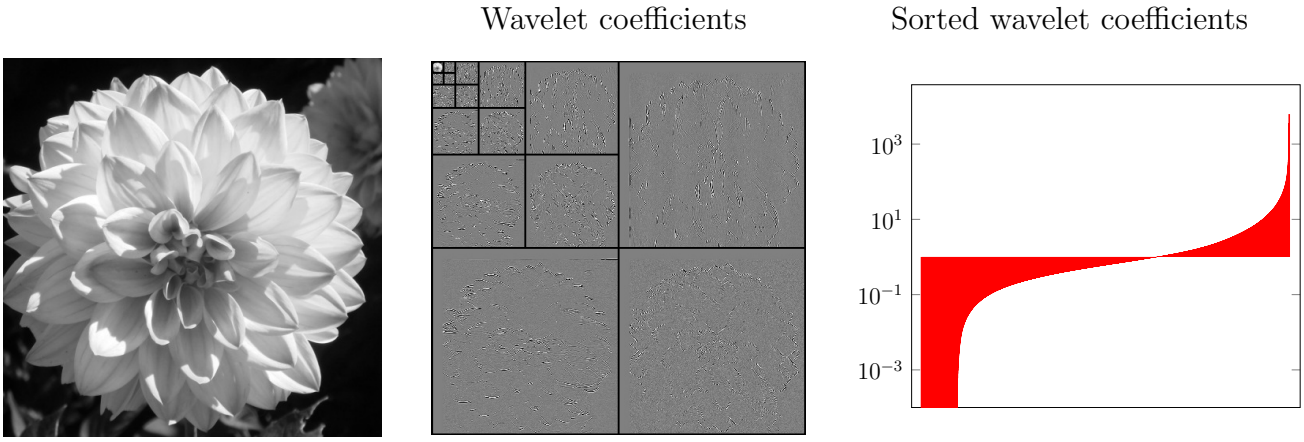


Figure 3: Coefficients in a biorthogonal wavelet basis (center) of a natural image (left). Plotting the ordered coefficients reveals that the image is highly compressible in the wavelet domain (right).

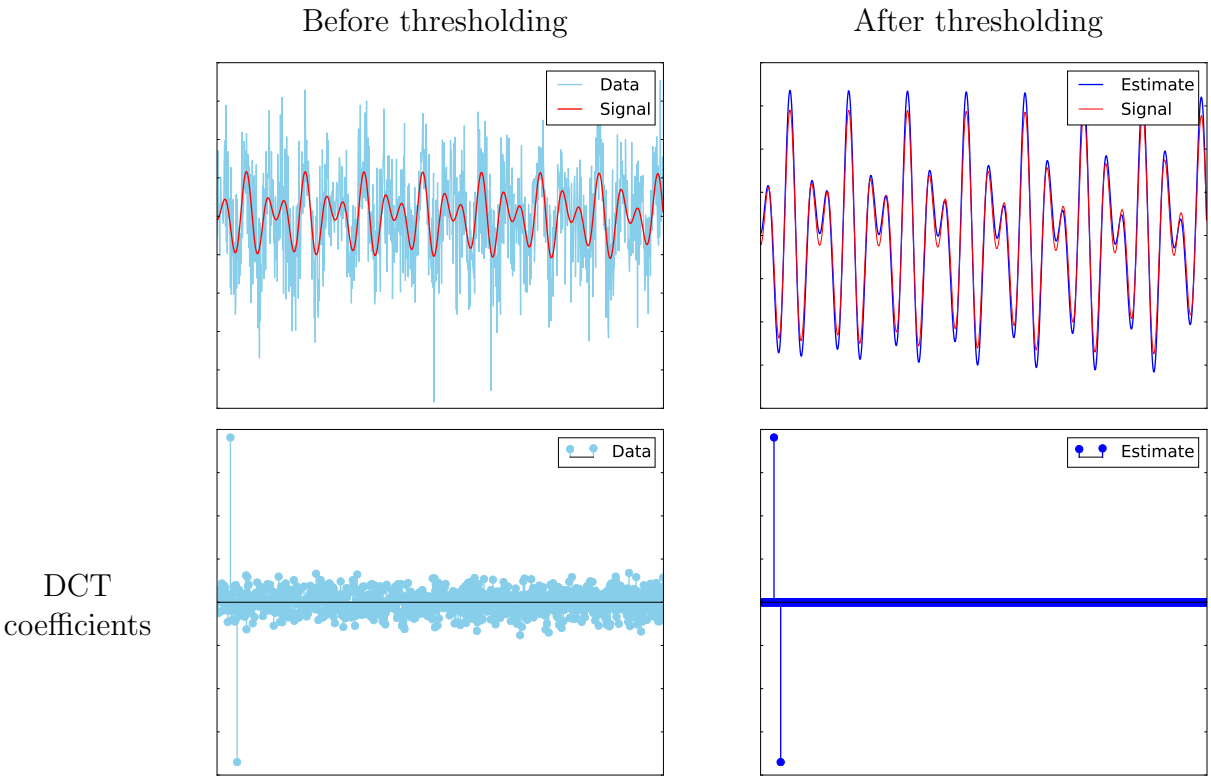


Figure 4: Denoising via hard thresholding in the DCT basis.

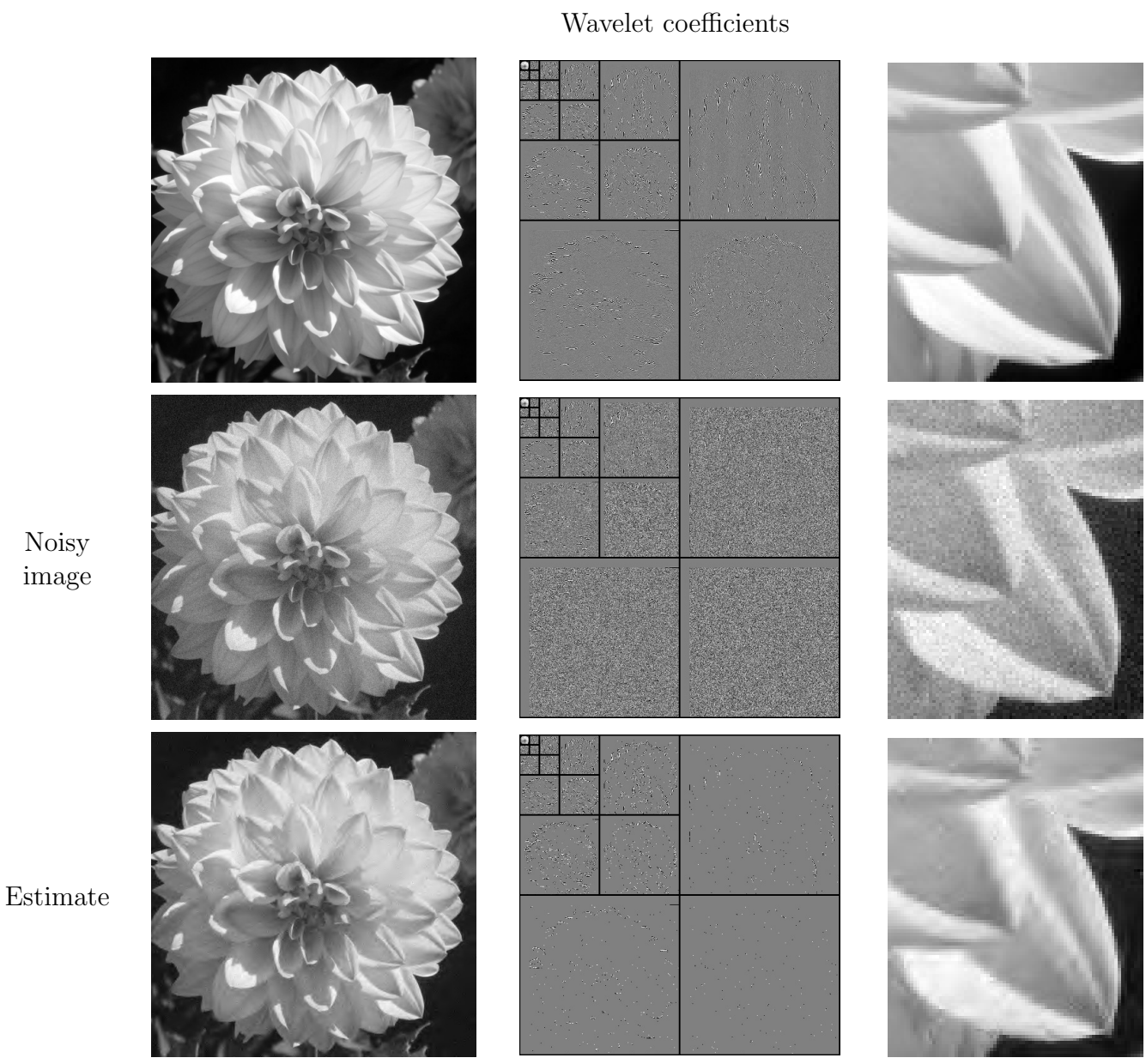


Figure 5: Denoising via hard thresholding in a biorthogonal wavelet basis.

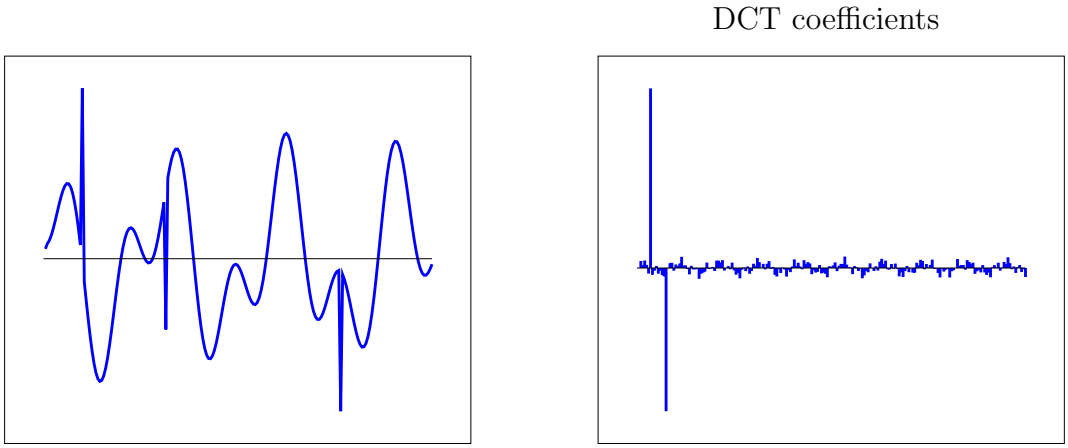


Figure 6: A signal consisting of spikes and sinusoids (left) and its DCT representation (right).

1.3 Sparsity in redundant representations

Figure 6 shows a signal that is not sparse either in a basis of spiky atoms or sinusoidal atoms. However, it is sparse in a dictionary that contains *both* sinusoids and spikes,

$$x = Dc = \begin{bmatrix} I & F \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = a + Fb, \tag{8}$$

where $I \in \mathbb{R}^{n \times n}$ is the identity matrix and $F \in \mathbb{R}^{n \times n}$ represents a basis of sinusoids (F stands for Fourier).

Clearly, the dictionary $D \in \mathbb{R}^{n \times 2n}$ is not invertible. There are many (in fact infinite!) ways of representing the signal in terms of the atoms of the dictionary. This makes it challenging to leverage the assumption that the signal has a sparse representation in the dictionary. Let us consider for a moment the problem of finding a sparse representation even if there is no noise. Ideally, we would like to solve the following optimization problem.

$$\min_{\tilde{c} \in \mathbb{R}^m} ||\tilde{c}||_0 \quad \text{such that } x = Dc, \tag{9}$$

where the ℓ_0 "norm" of a vector $x \in \mathbb{R}^n$ (which is not really a norm) is equal to the number of nonzero entries of x

$$||x||_0 := \text{card}\left(\{i \mid x_i \neq 0\}\right). \tag{10}$$

Problem 9 is intractable even for signals of very moderate size. However, it turns out that there is a tractable optimization problem that often produces sparse representations; it is obtained by replacing the ℓ_0 "norm" with the ℓ_1 norm

$$\min_{\tilde{c} \in \mathbb{R}^m} ||\tilde{c}||_1 \quad \text{such that } x = Dc. \tag{11}$$

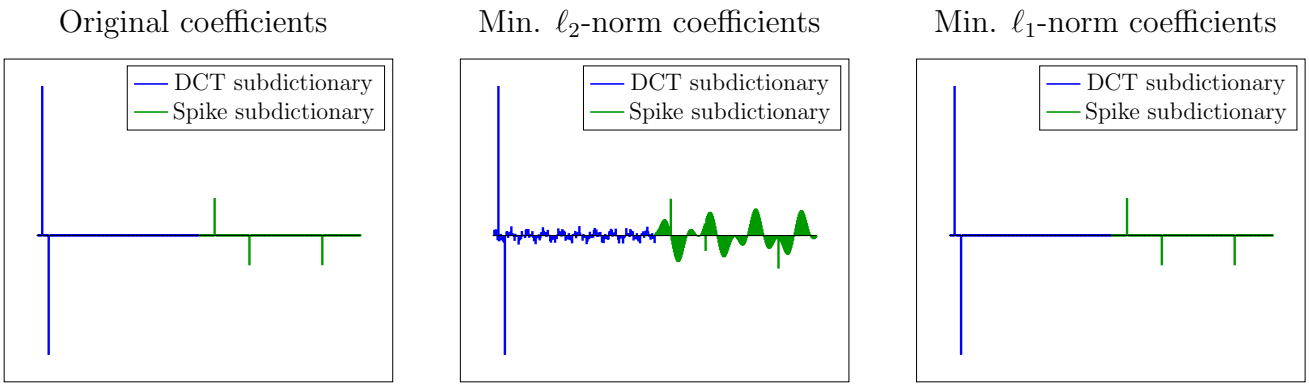


Figure 7: The original coefficients of the signal from Figure 6 in a dictionary containing spikes and sinusoids are shown on the left. Minimizing the ℓ_2 norm of the coefficients does not yield a sparse representation (center), but minimizing the ℓ_1 norm does.

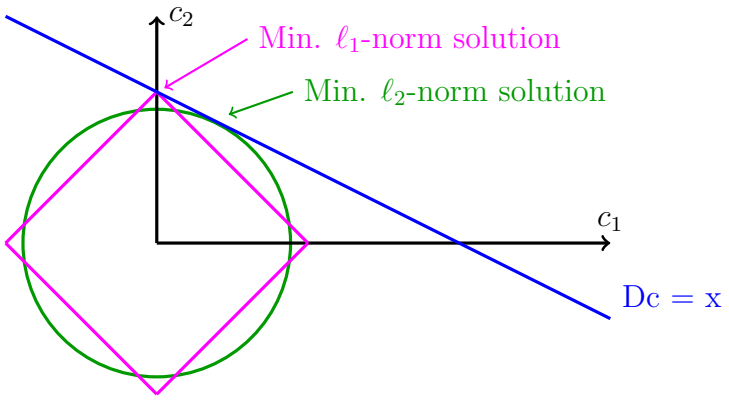


Figure 8: The minimum ℓ_1-norm solution is sparser than the minimum ℓ_2-norm solution because of the geometry of the ℓ_1-norm and ℓ_2-norm balls.

Figure 7 shows how ℓ_1-norm minimization allows to compute a sparse coefficient vector in the case of the signal in Figure 6. In contrast, minimizing the ℓ_2 norm of the coefficient vector produces a very dense solution.[2] This is due to the geometry of the ℓ_1 norm and the ℓ_2 norms. Figure 8 provides a picture for the case where the dictionary has two atoms. The ℓ_1-norm ball is more concentrated around the axes than the ℓ_2-norm ball. It is therefore more likely for the line representing the constraint $x = Dc$ to be tangent to the ball on an axis, where the solution has cardinality one instead of two. As a result, the minimum ℓ_1-norm solution is sparser than the minimum ℓ_2-norm solution.

In order to learn sparse representations when noise is present in the data, we can eliminate

[2]The numerical simulations in Figure 7 and several other figures in these notes were implemented using Stephen Boyd's and Michael Grant's CVX software which is available online at http://cvxr.com/cvx/

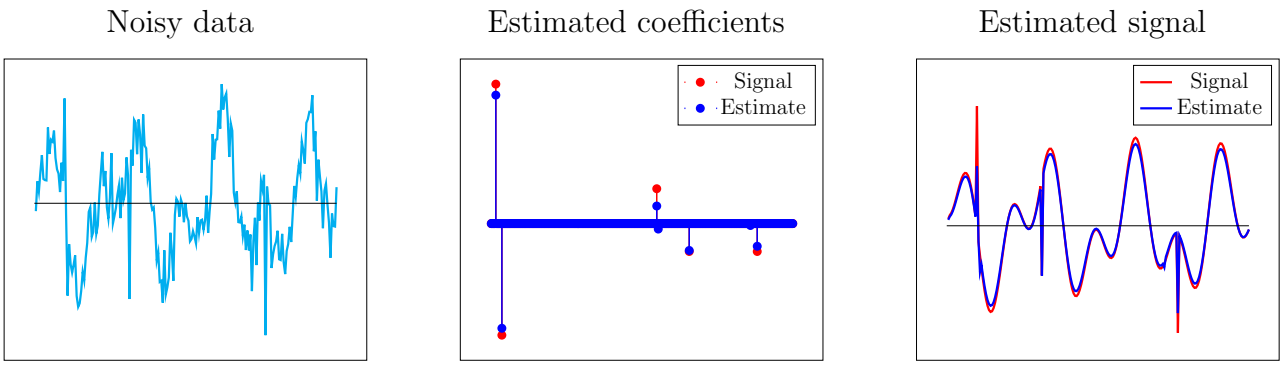


Figure 9: Denoising via ℓ_1-norm-regularized least squares.

the equality constraint in Problem 11 and add a data-fidelity term to the cost function. This is often known as **basis-pursuit denoising**. In more detail, we estimate the coefficients by solving

$$\hat{c} = \arg\min_{\tilde{c} \in \mathbb{R}^m} ||x - D\tilde{c}||_2^2 + \lambda ||\tilde{c}||_1 \tag{12}$$

where $\lambda > 0$ is a regularization parameter that determines the tradeoff between the term that promotes sparsity and the term that promotes data fidelity. Figure 9 shows the result of applying this denoising method to a signal that has a sparse representation in a dictionary of spikes and sinusoids.

1.4 Learning the dictionary

Dictionary-learning techniques allow to learn dictionaries directly from the data. This is very useful in situations where a dataset with a large number of signals cannot be compactly represented in any predefined dictionary. If the signals are grouped as columns of a matrix $X \in \mathbb{R}^{n \times k}$ (k is the number of signals in the dataset), the aim is to learn a dictionary $D \in \mathbb{R}^{n \times m}$ such that $X = DC$, where the matrix of coefficients $C \in \mathbb{R}^{m \times k}$ is very sparse. Following the heuristic that penalizing the ℓ_1 norm promotes sparse solutions, an option is to solve the following optimization program,

$$\min_{\widetilde{C} \in \mathbb{R}^{m \times k}} \left|\left|X - \widetilde{D}\widetilde{C}\right|\right|_{\mathrm{F}}^2 + \lambda \left|\left|\widetilde{C}\right|\right|_1 \quad \text{such that } \left|\left|\widetilde{D}_i\right|\right|_2 = 1, \quad 1 \leq i \leq m, \tag{13}$$

where the atoms of the dictionary are constrained to have unit norm. $||\cdot||_{\mathrm{F}}$ denotes the Frobenius norm, which is equal to the ℓ_2 norm of the entries of the matrix interpreted as a vector.

As we will discuss later in the course, this optimization problem is much more challenging to solve than (12), where the dictionary is fixed. Without getting into too much detail,

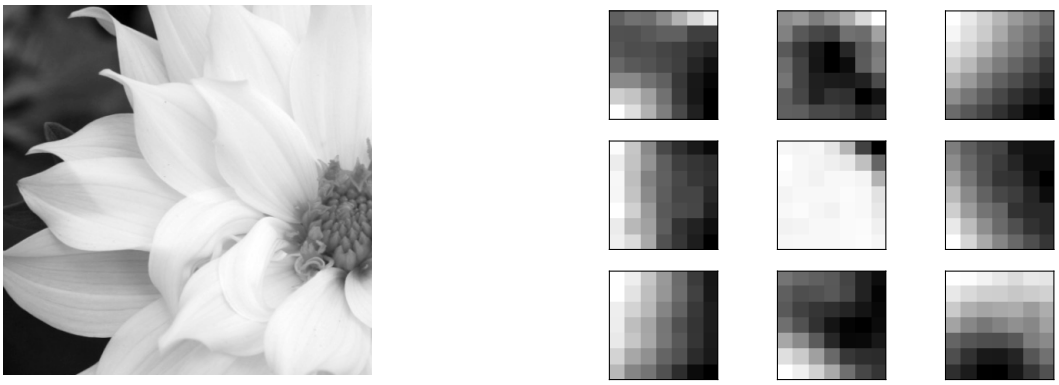


Figure 10: Atoms (right) learnt from patches extracted from a natural image (left).

the optimization problem is nonconvex, and hence may have many local minima. Figure 10 shows patches learnt from a natural image.[3] The corresponding dictionary can be used to denoise other images quite effectively, as shown in Figure 11.

1.5 Sparse regression

In statistics, the problem of **regression** is that of learning a function that allows to estimate a certain quantity of interest, the **response** or **dependent variable**, from other observed variables, known as **covariates** or **independent variables** . For example, we might be interested in estimating the price of a house from the number of square feet, the number of rooms and the location. In **linear regression**, we assume that the function is linear. In that case, the model is of the form

$$y_i \approx \sum_{j=1}^{p} \theta_j X_{ij}, \quad 1 \leq i \leq n, \tag{14}$$

where n is the number of data, $y \in \mathbb{R}^n$ is the response, $X_1, X_2, \ldots, X_p \in \mathbb{R}^n$ contain the covariates and $\theta_1, \theta_2, \ldots, \theta_p \in \mathbb{R}$ are the parameters of the linear model. In matrix form, $y \approx X\theta$. In order to calibrate the model, a common procedure is to fit the parameters so that the model approximates the response as closely as possible in ℓ_2 norm. This is achieved by solving the least-squares problem

$$\hat{\theta}_{\text{ls}} := \arg\min_{\tilde{\theta} \in \mathbb{R}^n} \left|\left| y - X\tilde{\theta} \right|\right|_2 . \tag{15}$$

[3]The numerical simulations in Figures 11 and other figures in these notes were implemented using scikit-learn, which is available online at http://scikit-learn.org.

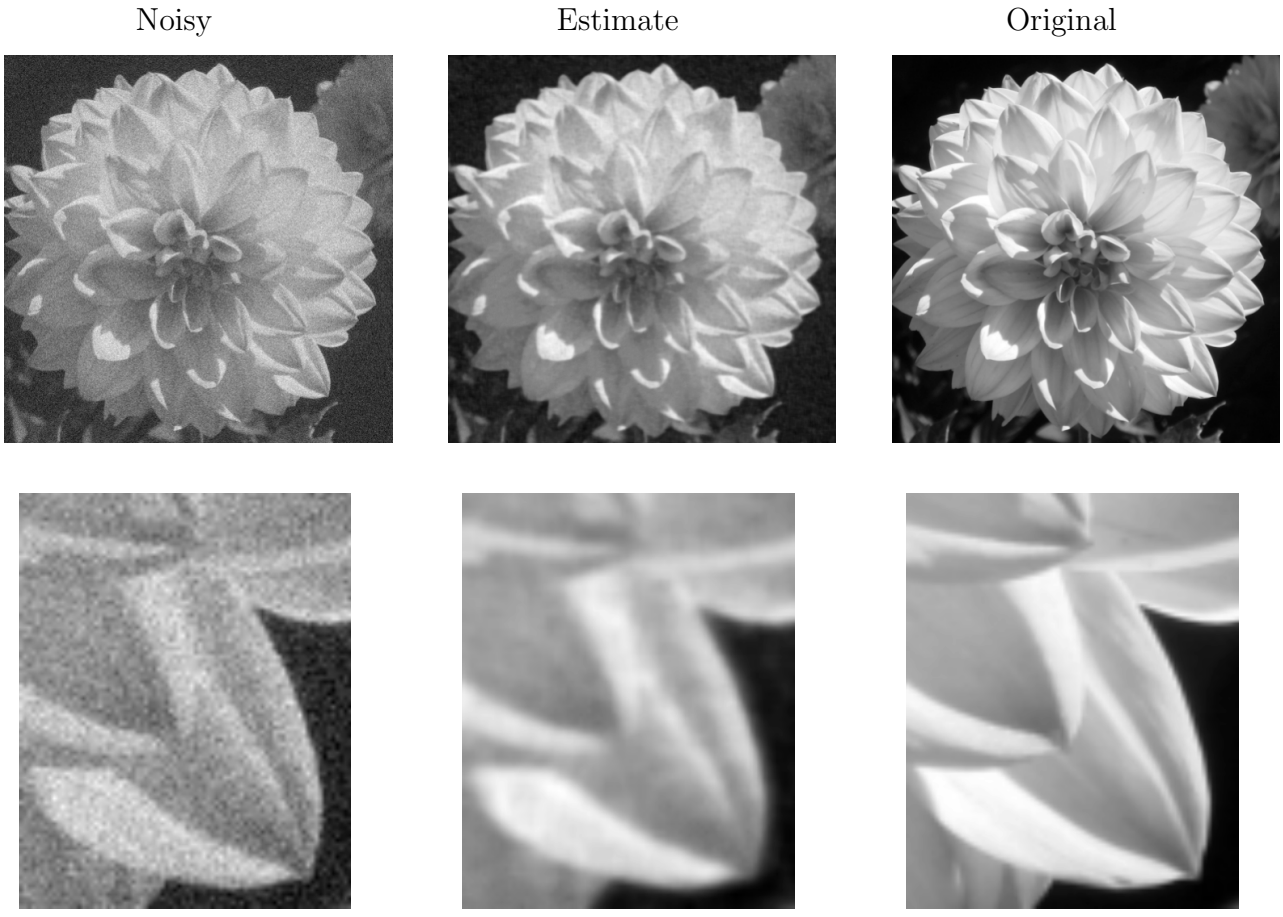


Figure 11: Denoising results using the dictionary learnt from the image shown in Figure 10.

In some applications, however, many of the covariates may actually be unrelated to the response. Imagine that we are trying to investigate the connection between the expression of several genes and a certain disease. Each covariate corresponds to a gene and the response quantifies a symptom of the disease. If most genes are unrelated to the disease, then we need to perform **model selection**, i.e. determine what genes we should incorporate in the regression function. Otherwise the model might try to explain the response using the irrelevant covariates. Although this could actually allow to obtain a better fit on the data that we are using to learn the model, it will hurt the ability of the model to **generalize** to new data. In machine learning and statistics this is known as **overfitting**. For this reason, if we have reason to suspect that only a few covariates are actually relevant, it makes sense to try to fit a sparse model to the data. As in the case of sparse representations in redundant dictionaries, penalizing the ℓ_1-norm often achieves this goal. In statistics, ℓ_1-norm regularized least squares is known as the **lasso**,

$$\hat{\theta}_{\text{lasso}} := \arg\min_{\tilde{\theta} \in \mathbb{R}^n} \left|\left| y - X\tilde{\theta} \right|\right|_2^2 + \lambda \left|\left| \tilde{\theta} \right|\right|_1, \tag{16}$$

$\lambda > 0$ is a regularization parameter that controls the level of regularization.

Let us illustrate all of this with a simple numerical simulation.

1. We generate a **training set** by computing the response as a linear combination of 3 covariates and add some noise to the data

$$y_{\text{train}} = \begin{bmatrix} X_1^{\text{train}} & X_2^{\text{train}} & X_3^{\text{train}} \end{bmatrix} \theta + z_{\text{train}}, \tag{17}$$

   where $n = 100$, i.e. the response, covariates and noise vectors all have dimension 100.

2. We fit the model using a covariate matrix that includes 47 extra covariates X_4, X_5, $\ldots$, X_{50} that are completely independent from the response,

$$X_{\text{train}} = \begin{bmatrix} X_1 & X_2 & X_3 & X_4 & \cdots & X_{50} \end{bmatrix}, \tag{18}$$

3. We test the model using a and a **test set** that consists of a response

$$y_{\text{test}} = \begin{bmatrix} X_1^{\text{test}} & X_2^{\text{test}} & X_3^{\text{test}} \end{bmatrix} \theta + z_{\text{test}}, \tag{19}$$

   and a matrix of covariates

$$X_{\text{test}} = \begin{bmatrix} X_1 & X_2 & X_3 & X_4 & \cdots & X_{50} \end{bmatrix}, \tag{20}$$

   where again $n = 100$, i.e. the response, covariates and noise vectors all have dimension 100. The coefficient vector θ is the same as in the training set.

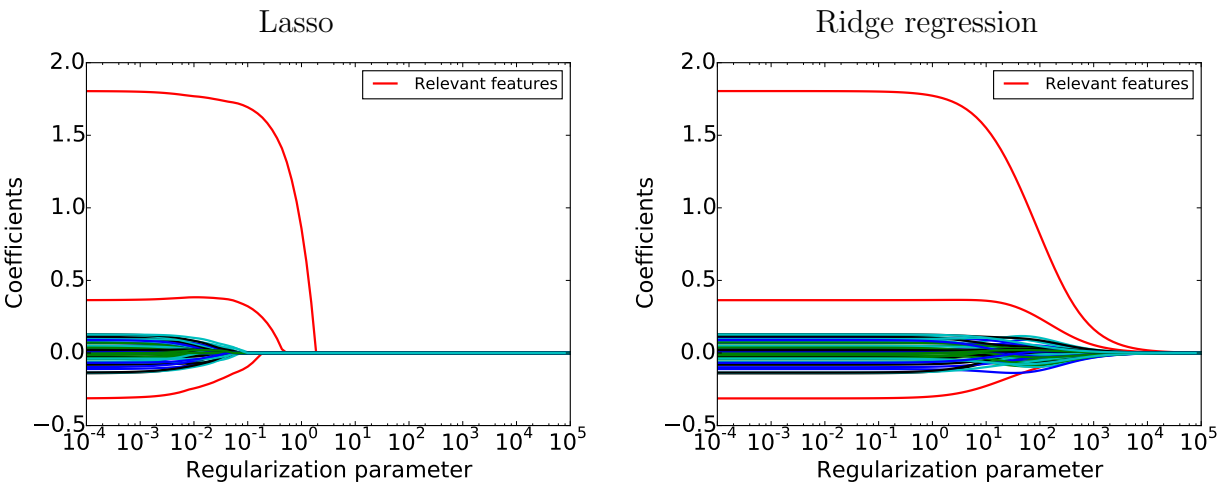


Figure 12: Value of the coefficients in the lasso (left) and ridge-regression models for different values of the regularization parameter λ.

4. We fit the model using least-squares regression, the lasso and also **ridge regression**, which penalizes the ℓ_2 norm of the parameter vector and is known as Tikhonov regularization in the applied-mathematics literature,

$$\hat{\theta}_{\text{ridge}} := \arg\min_{\tilde{\theta} \in \mathbb{R}^n} \left|\left| y - X\tilde{\theta} \right|\right|_2^2 + \lambda \left|\left| \tilde{\theta} \right|\right|_2^2. \tag{21}$$

Figure 12 shows the value of the fitted parameters for different values of the regularization parameter. When λ is very large, all of the coefficients are set to zero by the lasso and by ridge regression because the fitting error has no weight in the cost function. When λ is very small, both methods are equivalent to least-squares regression. In between these two regimes, the lasso model is sparse and for a certain range of values of λ it only includes the relevant features. In contrast, ridge regression allows to diminish the influence of the irrelevant covariates, but is not sparse. As discussed previously, this is due to the geometry of the ℓ_1-norm and ℓ_2-norm balls (see Figure 8).

Figure 13 plots the relative error achieved on the training and test sets for least-squares regression, the lasso and ridge regression. In the training set, least-squares regression achieves the smallest error by overfitting the data. This becomes obvious when we compare the error on the test set, where the fit of the sparse model obtained by the lasso is significantly better. Ridge regression also prevents overfitting to some extent, but not as effectively as the lasso.

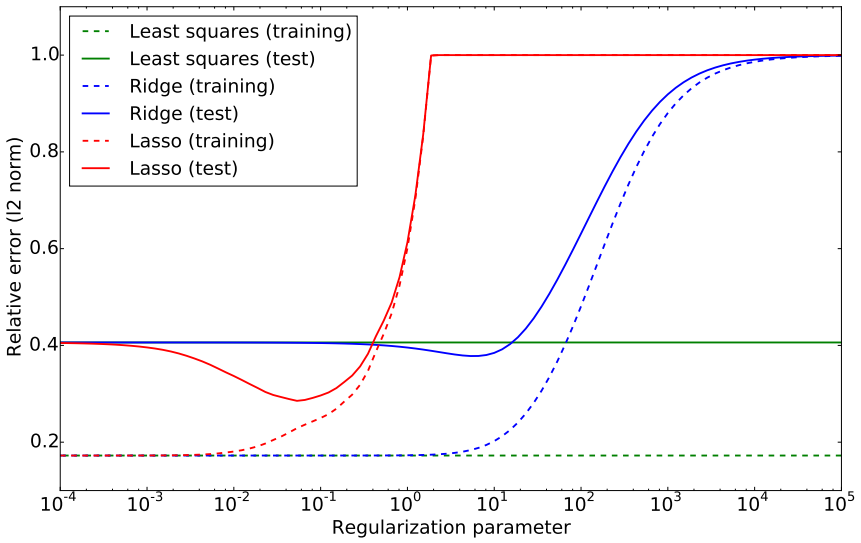


Figure 13: Relative ℓ_2 norm error in estimating the response achieved using least-squares regression, the lasso and ridge regression.

1.6 Inverse problems

This section describes two inverse problems that are ill posed unless we make some assumptions on the signal structure. In particular, leveraging sparsity-based models through the use of ℓ_1-norm regularization often allows to obtain estimates, even if the problem is underdetermined.

1.6.1 Super-resolution

Extracting fine-scale information from low-resolution data is a major challenge in many areas of the applied sciences. In microscopy, astronomy and any other application employing an optical device, spatial resolution is fundamentally limited by diffraction. An example with real data is shown in Figure 14. Figure 15 illustrates a model for the data-acquisition process in such cases: the object of interest is convolved with a point-spread function that blurs the fine-scale details, acting essentially as a low-pass filter. The problem of super-resolution is that of reconstructing the original signal from the blurred measurements.

Computing the convolution between two signals can be carried out by multiplying their spectra in the frequency domain. To simplify matters, let us assume that the signal of interest is a vector $x \in \mathbb{R}^n$ and that the point-spread function of the sensing mechanism is a perfect low-pass filter. In that case, the spectrum of the measurements, which we denote

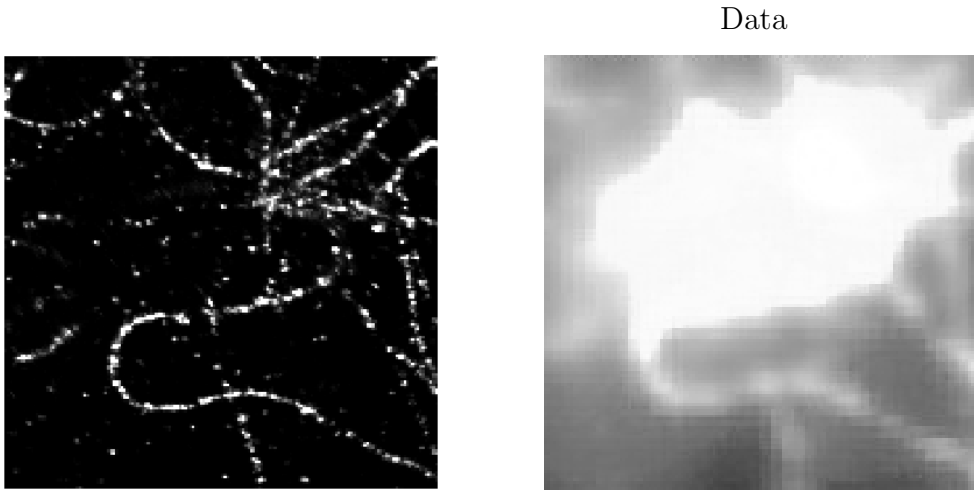


Figure 14: An example of deconvolution in fluorescence microscopy applied to data from the Moerner lab at Stanford by V. Morgenshtern. On the left, it is not possible to distinguish the individual fluorescent probes due to heavy aliasing. On the right, deconvolving the point sources in the individual frames and pooling the frames together reveals the fine-scale details of the object of interest.

by y, is equal to

$$y = \mathcal{F}_{\mathrm{L}}\, x, \tag{22}$$

where $x \in \mathbb{R}^m$ is the signal of interest and $\mathcal{F}_{\mathrm{L}}$ is a $n \times m$ submatrix of the discrete Fourier transform (DFT) matrix. Since some frequencies are suppressed, the number of measurements is greater than the dimension of the signal $m > n$, which implies that the system is underdetermined. If there is any solution to the system, there are infinite solutions, so we need to make further assumptions for the inverse problem not to be completely ill posed.

In many applications, a reasonable assumption is that the signal is well modeled as a superposition of point sources; examples include celestial bodies in astronomy, fluorescent probes in microscopy, or line spectra in signal processing. In such cases, we can enforce a sparsity prior in the same way by solving an ℓ_1-norm minimization problem, as in Section 1.3,

$$\min_{\tilde{x} \in \mathbb{R}^m} ||\tilde{x}||_1 \quad \text{such that } y = \mathcal{F}_{\mathrm{L}}\, \tilde{x}. \tag{23}$$

Figure 16 shows a simple example in which ℓ_1-norm minimization allows to recover a sparse signal exactly. In contrast, minimizing the ℓ_2 norm does not produces a sparse estimate, as illustrated in Figure 16. Later in the course, we will characterize under what conditions ℓ_1-norm minimization is guaranteed to achieve exact recovery.

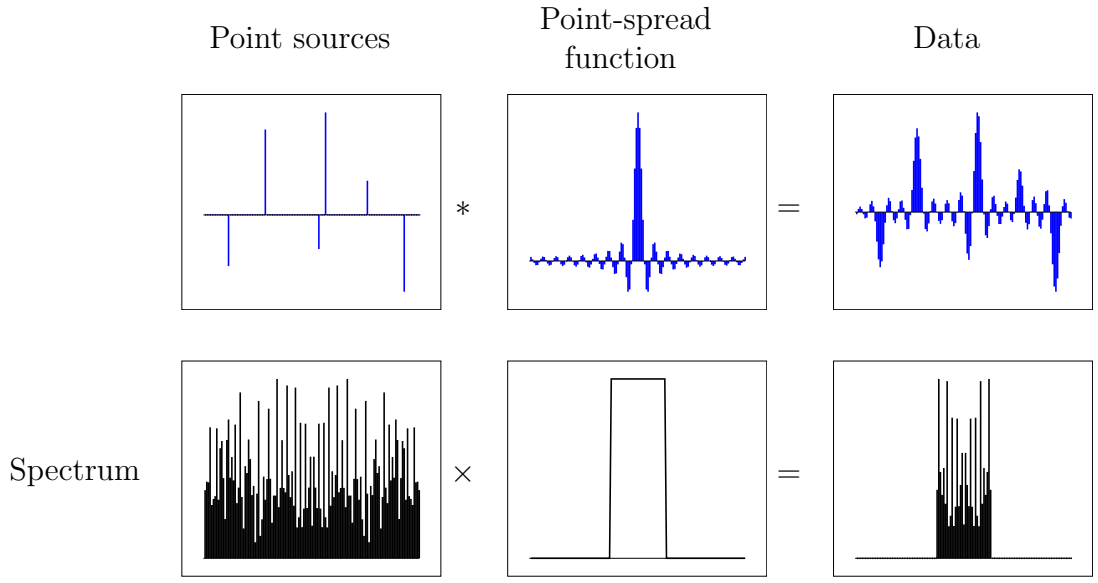


Figure 15: Convolution with a low-pass point-spread function (above) is equivalent to pointwise multiplication with a low-pass filter in the frequency domain (below).

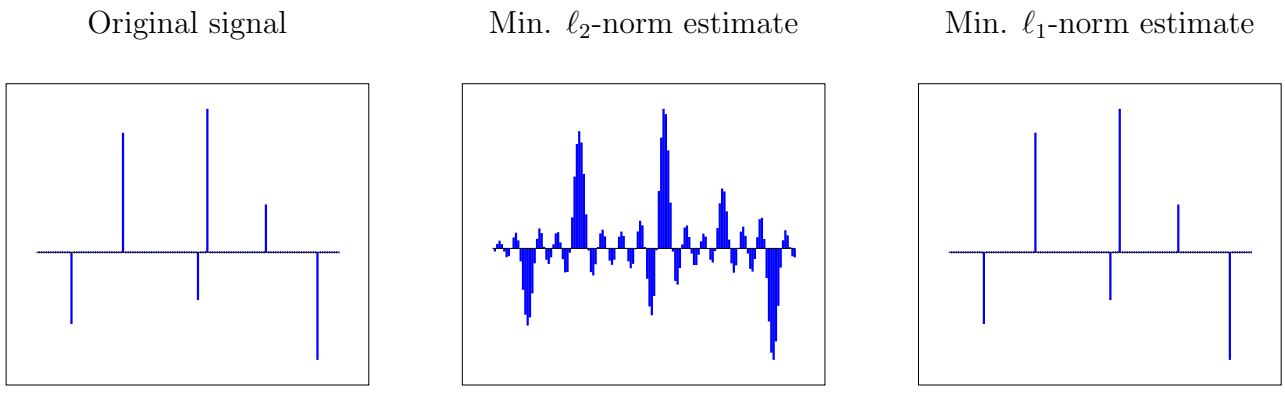


Figure 16: Minimizing the ℓ_1 norm of the estimate (right) allows to estimate the original signal (left) exactly, whereas minimizing the ℓ_2 norm does not produce a sparse estimate.

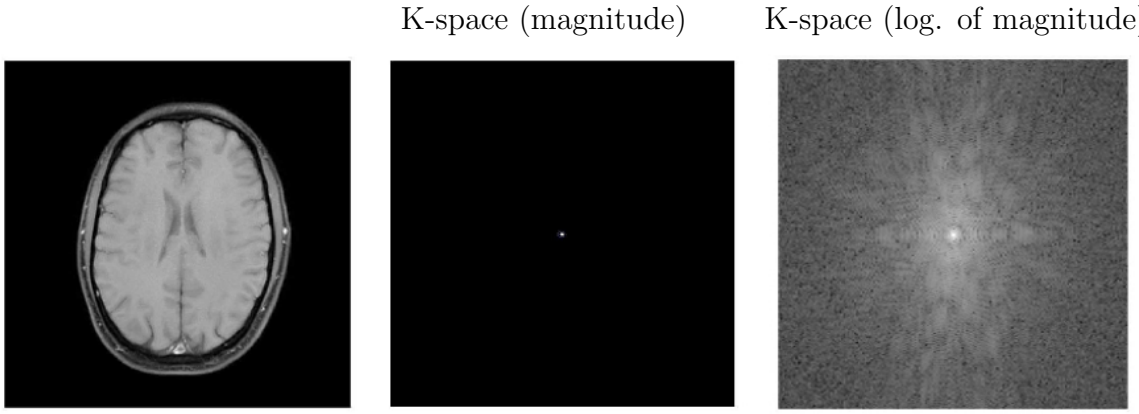


Figure 17: Image of a brain obtained by MRI, along with the magnitude of its 2D-Fourier or k-space representation and the logarithm of this magnitude.

1.6.2 Compressed sensing

Magnetic resonance imaging (MRI) is a popular medical imaging technique used in radiology. A simplified model of the data obtained via MRI is that they correspond to samples of the 2D or 3D Fourier transform of an image, known as **k space** in MRI jargon. An estimate of the image can be obtained by computing the inverse Fourier transform of the data, as shown in Figure 17.

An important challenge in MRI is to reduce measurement time. **Compressed sensing** achieves this by randomly undersampling the k-space representation of the image. Let us first consider a 1D version of the problem, where the signal is an m-dimensional vector x. The data y may be modeled as random samples of the DFT of the signal. More precisely,

$$y = \mathcal{F}_\Omega \, x, \tag{24}$$

where $x \in \mathbb{R}^m$ and the linear operator $\mathcal{F}_\Omega \in \mathbb{R}^{n \times m}$ corresponds to n random rows of the discrete Fourier transform (DFT) matrix. As in the case of super-resolution, $m > n$ so the system is underdetermined. Assuming that the signal is sparse, we can again solve ℓ_1-norm minimization problem to promote a sparse solution,

$$\min_{\tilde{x} \in \mathbb{R}^m} ||\tilde{x}||_1 \quad \text{such that } y = \mathcal{F}_\Omega \, \tilde{x}, \tag{25}$$

Figure 18 shows the results of applying compressed sensing to a sparse 1D signal.

In general, images such as the one in Figure 17 are not sparse. However they are often sparse in certain representations such as wavelet dictionaries, as discussed in Section 1.2. Minimizing the ℓ_1 norm of the coefficients in the sparsifying dictionary, allows to recover such images very accurately. An example is shown in Figure 19.

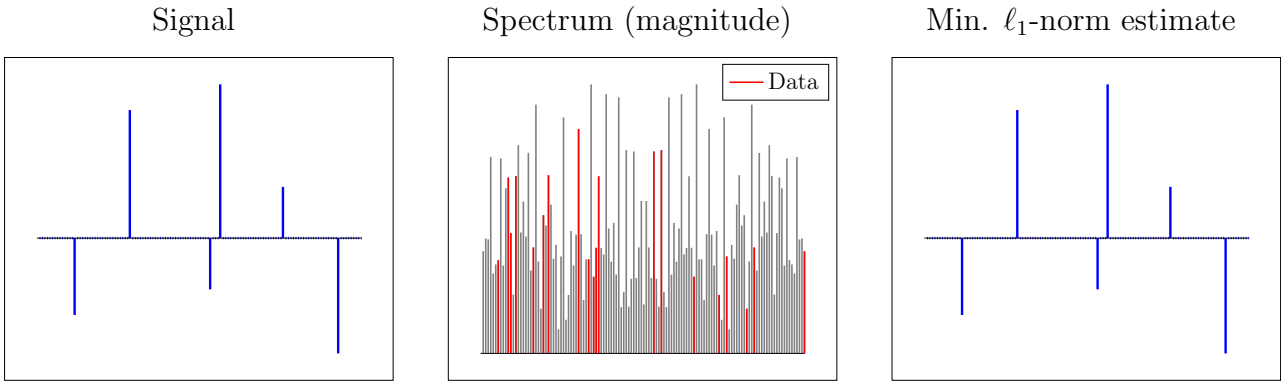


Figure 18: Minimizing the ℓ_1 norm of the estimate (right) allows to estimate the original signal (left) exactly from a small number of random samples of its spectrum (center).

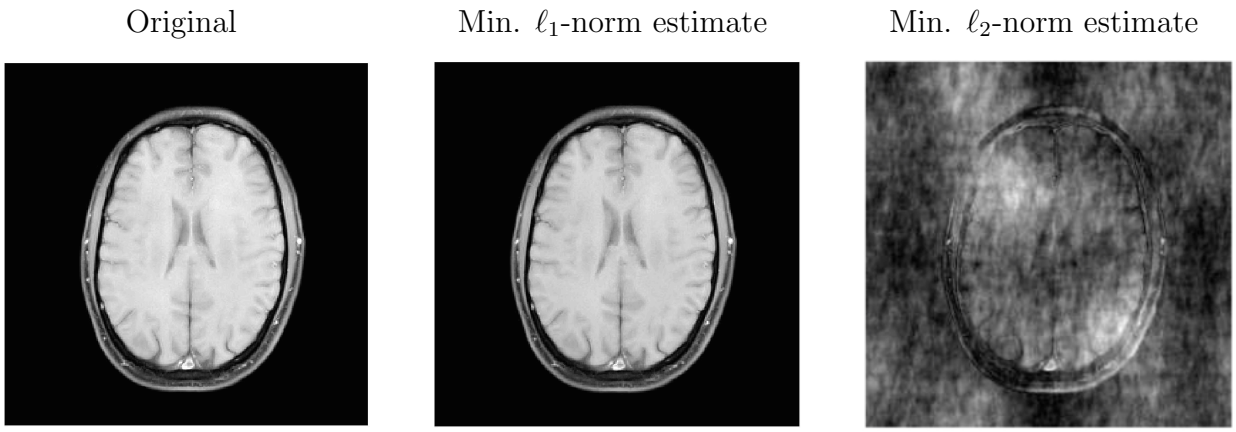


Figure 19: Compressed-sensing reconstruction (center) of a brain image (left) from random k-space samples. Minimizing the ℓ_2-norm recovers a heavily-aliased image (right).

Figure 20: A depiction of the Netflix challenge in matrix form. Each row corresponds to a user that ranks a subset of the movies, which correspond to the columns. The figure is due to Mahdi Soltanolkotabi.

2 Low-rank models

In this section we describe three applications of low-rank models: matrix completion for movie-rating prediction, low rank + sparse models for background subtraction and nonnegative matrix factorization for topic modeling.

2.1 Matrix completion

The Netflix Prize was a contest organized by Netflix from 2007 to 2009 in which teams of data scientists tried to develop algorithms to improve the prediction of movie ratings. The problem of predicting ratings can be recast as that of completing a matrix from some of its entries, as illustrated in Figure 20. It turns out that matrices of ratings are often well modeled as being approximately low rank. We demonstrate this through a simple example.

Bob, Molly, Mary and Larry rate the following six movies from 1 to 5,

$$A := \begin{array}{c} \begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} \end{array} \\ \begin{pmatrix} 1 & 1 & 5 & 4 \\ 2 & 1 & 4 & 5 \\ 4 & 5 & 2 & 1 \\ 5 & 4 & 2 & 1 \\ 4 & 5 & 1 & 2 \\ 1 & 2 & 5 & 5 \end{pmatrix} \end{array} \begin{array}{l} \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{array} \tag{26}$$

We subtract the average rating,

$$\mu := \frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}, \tag{27}$$

$$\tag{28}$$

from each entry in the matrix and then compute its singular value decomposition

$$A - \bar{A} = USV^T = U \begin{bmatrix} 7.79 & 0 & 0 & 0 \\ 0 & 1.62 & 0 & 0 \\ 0 & 0 & 1.55 & 0 \\ 0 & 0 & 0 & 0.62 \end{bmatrix} V^T, \tag{29}$$

where

$$\bar{A} := \begin{bmatrix} \mu & \mu & \cdots & \mu \\ \mu & \mu & \cdots & \mu \\ \cdots & \cdots & \cdots & \cdots \\ \mu & \mu & \cdots & \mu \end{bmatrix}. \tag{30}$$

The fact that the first singular value is significantly larger than the rest suggests that the matrix may be well approximated by a rank-1 matrix. This is the case (for ease of comparison the values of A are shown in brackets):

$$\bar{A} + \sigma_1 U_1 V_1^T = \begin{array}{c} \begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} \end{array} \\ \begin{pmatrix} 1.34\,(1) & 1.19\,(1) & 4.66\,(5) & 4.81\,(4) \\ 1.55\,(2) & 1.42\,(1) & 4.45\,(4) & 4.58\,(5) \\ 4.45\,(4) & 4.58\,(5) & 1.55\,(2) & 1.42\,(1) \\ 4.43\,(5) & 4.56\,(4) & 1.57\,(2) & 1.44\,(1) \\ 4.43\,(4) & 4.56\,(5) & 1.57\,(1) & 1.44\,(2) \\ 1.34\,(1) & 1.19\,(2) & 4.66\,(5) & 4.81\,(5) \end{pmatrix} \end{array} \begin{array}{l} \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{array} \tag{31}$$

This rank-1 model has an interesting interpretation. The first left singular vector is equal to

$$U_1 = \begin{array}{c} \begin{array}{cccccc} \text{D. Knight} & \text{Spiderman 3} & \text{Love Act.} & \text{B.J.'s Diary} & \text{P. Woman} & \text{Superman 2} \end{array} \\ \begin{pmatrix} -0.45 & -0.39 & 0.39 & 0.39 & 0.39 & -0.45 \end{pmatrix} \end{array}.$$

Entries with similar values represent movies that are rated similarly by viewers. The first right singular vector is equal to

$$V_1 = \begin{array}{c} \begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} \end{array} \\ \begin{pmatrix} 0.48 & 0.52 & -0.48 & -0.52 \end{pmatrix} \end{array}. \tag{32}$$

Here, entries with similar values correspond to users that have a similar taste (Bob and Molly vs Mary and Larry).

Now, let us consider the problem of completing the matrix if we only have access to a subset of its entries. This problem is known as **matrix completion**.

$$\begin{array}{c} \begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} \end{array} \\ \begin{pmatrix} 1 & ? & 5 & 4 \\ ? & 1 & 4 & 5 \\ 4 & 5 & 2 & ? \\ 5 & 4 & 2 & 1 \\ 4 & 5 & 1 & 2 \\ 1 & 2 & ? & 5 \end{pmatrix} \end{array} \begin{array}{l} \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{array} \tag{33}$$

Knowing that rating matrices are often approximately low-rank, we would like to compute a low-rank estimate from the available data. Achieving this through an optimization problem that penalizes the rank of the estimate is not computationally tractable. However penalizing the ℓ_1 norm of the singular values of the matrix does promote low-rank estimates. The ℓ_1 norm of the singular values is called the **nuclear norm** of the matrix, and is usually denoted by $||\cdot||_*$. In our example, we can leverage this insight in the following way.

1. We compute the average observed rating and subtract it from each entry in the matrix. We denote the vector of centered ratings by y.

2. We solve the optimization problem

$$\min_{\widetilde{X} \in \mathbb{R}^{m \times n}} \left|\left| \widetilde{X}_\Omega - y \right|\right|_2^2 + \lambda \left|\left| \widetilde{X} \right|\right|_* \tag{34}$$

   where $\lambda > 0$ is a regularization parameter. We denote the set of observed indices of the matrix as Ω. For any $m \times n$ matrix M, M_Ω is a vector containing the entries of M indexed by Ω.

3. We add the average observed rating to the solution of the optimization problem and round each entry to the nearest integer.

The result is

$$\begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} & \\ \left(\begin{array}{cccc} 1 & 2\ (1) & 5 & 4 \\ 2\ (2) & 1 & 4 & 5 \\ 4 & 5 & 2 & 2\ (1) \\ 5 & 4 & 2 & 1 \\ 4 & 5 & 1 & 2 \\ 1 & 2 & 5\ (5) & 5 \end{array}\right) & \begin{array}{l} \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{array} \end{array} \tag{35}$$

For comparison the original ratings are shown in brackets.

2.2 Low rank + sparse model

In computer vision, the problem of **background subtraction** is that of separating the background and foreground of a video sequence. Imagine that we take a video of a static background. We then stack the video frames in a matrix M, where each column corresponds to a vectorized frame. If the background is completely static, then all the frames are equal to a certain vector $f \in \mathbb{R}^m$ (m is the number of pixels in each frame) and the matrix is rank 1

$$M = \begin{bmatrix} f & f & \cdots & f \end{bmatrix} = f \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}.$$

If the background is not completely static, but instead experiences gradual changes, then the matrix containing the frames will be approximately low rank. Now, assume that there are sudden events in the foreground. If these events occupy a small part of the field of view and do not last very long, then the corresponding matrix can be modeled as sparse (most entries are equal to zero).

These observations motivate the following method for background subtraction. Stack the frames as columns of a matrix and separate the matrix into a low-rank and a sparse component. Applying the heuristic that the nuclear norm promotes low-rank structure and the ℓ_1 norm promotes sparsity, this suggests solving the optimization problem

$$\min_{\widetilde{L},\widetilde{S}\in\mathbb{R}^{m\times n}} \left|\left|\widetilde{L}\right|\right|_* + \lambda \left|\left|\widetilde{S}\right|\right|_1 \quad \text{such that } \widetilde{L} + \widetilde{S} = Y, \tag{36}$$

where $\lambda > 0$ is a regularization parameter and $||\cdot||_1$ denotes the ℓ_1 norm of a matrix interpreted as a vector. The results of applying this method to a real video sequence are shown in Figure 21.

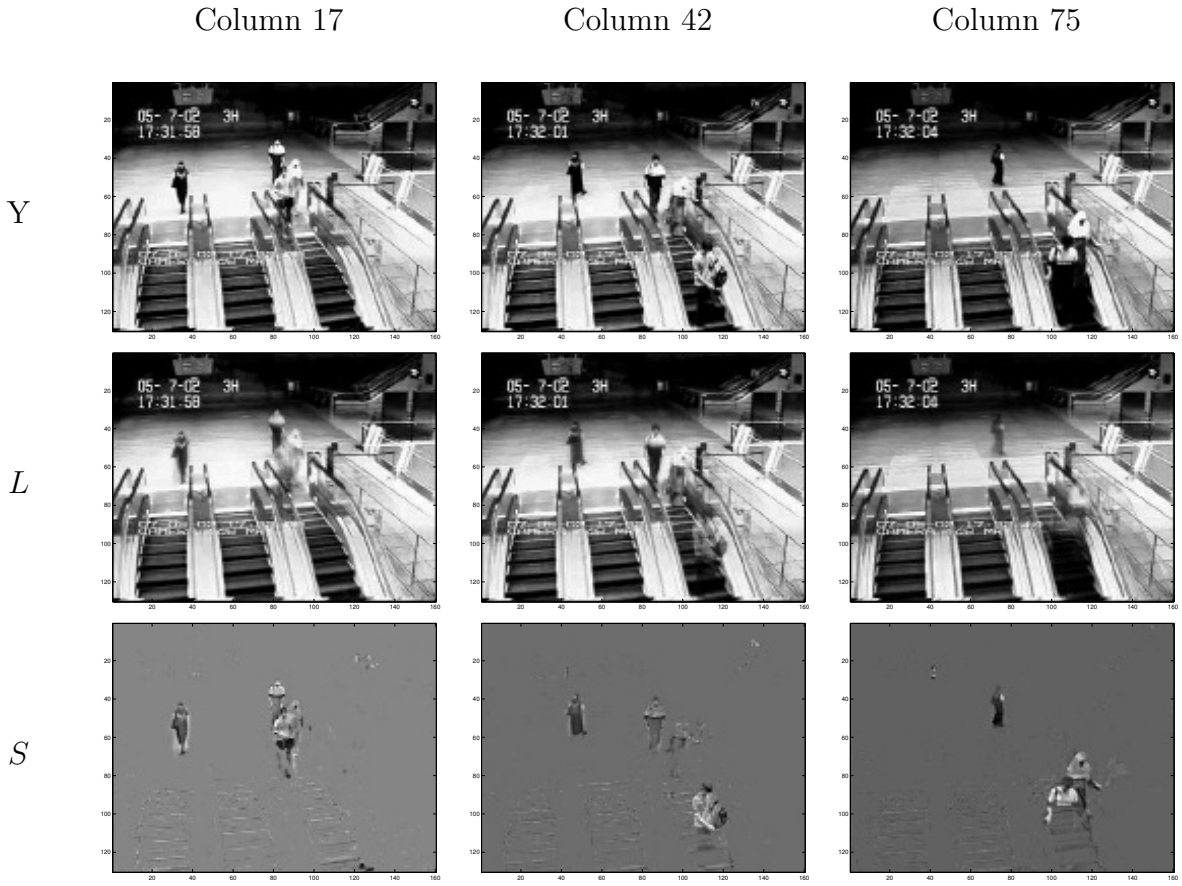


Figure 21: Background subtraction results from a video. This example is due to Stephen Becker. The code is available at http://cvxr.com/tfocs/demos/rpca.

2.3 Nonnegative matrix factorization

Topic modeling aims to learn the thematic structure of a text corpus automatically. Let us work on a simple example. We take six newspaper articles and compute the frequency of a list of words in each of them. Our final goal is to separate the words into different clusters that hopefully correspond to different topics. The following matrix contains the counts for each word and article. Each entry contains the number of times that the word corresponding to column j is mentioned in the article corresponding to row i.

$$A = \begin{array}{c} \begin{array}{cccccccccc} \text{singer} & \text{GDP} & \text{senate} & \text{election} & \text{vote} & \text{stock} & \text{bass} & \text{market} & \text{band} & \textit{Articles} \end{array} \\ \begin{pmatrix} 6 & 1 & 1 & 0 & 0 & 1 & 9 & 0 & 8 \\ 1 & 0 & 9 & 5 & 8 & 1 & 0 & 1 & 0 \\ 8 & 1 & 0 & 1 & 0 & 0 & 9 & 1 & 7 \\ 0 & 7 & 1 & 0 & 0 & 9 & 1 & 7 & 0 \\ 0 & 5 & 6 & 7 & 5 & 6 & 0 & 7 & 2 \\ 1 & 0 & 8 & 5 & 9 & 2 & 0 & 0 & 1 \end{pmatrix} \end{array} \begin{array}{c} \\ \text{a} \\ \text{b} \\ \text{c} \\ \text{d} \\ \text{e} \\ \text{f} \end{array}$$

Computing the singular-value decomposition of the matrix– after subtracting the mean of each entry as in (29)– we determine that the matrix is approximately low rank

$$A - \bar{A} = USV^T = U \begin{bmatrix} 19.32 & 0 & 0 & 0 & & \\ 0 & 14.46 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.99 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.77 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.67 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.93 \end{bmatrix} V^T. \tag{37}$$

Unfortunately the singular vectors do not have an intuitive interpretation as in Section 2.1. In particular, they do not allow to cluster the words

$$\begin{array}{rcccccccc} & & \text{a} & \text{b} & \text{c} & \text{d} & \text{e} & \text{f} \\ U_1 & = & (-0.51 & -0.40 & -0.54 & -0.11 & -0.38 & -0.38) \\ U_2 & = & (\ 0.19 & -0.45 & -0.19 & -0.69 & -0.2 & -0.46) \\ U_3 & = & (\ 0.14 & -0.27 & -0.09 & -0.58 & -0.69 & -0.29) \end{array} \tag{38}$$

or the articles

$$\begin{array}{rcccccccccc} & & \text{singer} & \text{GDP} & \text{senate} & \text{election} & \text{vote} & \text{stock} & \text{bass} & \text{market} & \text{band} \\ V_1 & = & (-0.38 & 0.05 & 0.4 & 0.27 & 0.4 & 0.17 & -0.52 & 0.14 & -0.38) \\ V_2 & = & (\ 0.16 & -0.46 & 0.33 & 0.15 & 0.38- & 0.49 & 0.1 & -0.47 & 0.12\ ) \\ V_3 & = & (-0.18 & -0.18 & -0.04 & -0.74 & -0.05 & 0.11 & -0.1 & -0.43 & -0.43) \end{array} \tag{39}$$

A problem here is that the singular vectors have negative entries that are difficult to interpret. In the case of rating prediction, negative ratings mean that a person does not like a movie. In contrast articles either are about a topic or they are not. Following this intuition, we can try to obtain a low-rank model with nonnegative entries, i.e. compute two matrices $W \in \mathbb{R}^{m \times r}$ and $H \in \mathbb{R}^{r \times n}$ such that

$$M \approx WH, \quad W_{i,j} \geq 0, \qquad 1 \leq i \leq m,\ 1 \leq j \leq r, \tag{40}$$

$$H_{i,j} \geq 0, \qquad 1 \leq i \leq r,\ 1 \leq i \leq n, \tag{41}$$

where the rank of the model is equal to $r < m, n$. This is known as **nonnegative matrix factorization**. Solving the problem for our example with $r = 3$ yields some interesting results. The entries of H allow to cluster the words into three topics,

$$\begin{array}{rcccccccccc} & & \text{singer} & \text{GDP} & \text{senate} & \text{election} & \text{vote} & \text{stock} & \text{bass} & \text{market} & \text{band} & \\ H_1 & = & (\ 0.34 & 0 & 3.73 & 2.54 & 3.67 & 0.52 & 0 & 0.35 & 0.35\ ) & \\ H_2 & = & (\ 0 & 2.21 & 0.21 & 0.45 & 0 & 2.64 & 0.21 & 2.43 & 0.22\ ) & \\ H_3 & = & (\ 3.22 & 0.37 & 0.19 & 0.2 & 0 & 0.12 & 4.13 & 0.13 & 3.43\ ) & \end{array} \tag{42}$$

The first topic correspond to the entries that are not zero (or very small) in H_1: senate, election and vote. The second corresponds to H_2: GDP, stock and market. The third corresponds to H_3: singer, bass and band.

The entries of W allow to assign topics to articles. b, e and f are about politics (topic 1), d and e about economics (topic 3) and a and c about music (topic 3)

$$\begin{array}{rcccccccc} & & \text{a} & \text{b} & \text{c} & \text{d} & \text{e} & \text{f} & \\ W_1 & = & (0.03 & 2.23 & 0 & 0 & 1.59 & 2.24) & \\ W_2 & = & (\ 0.1 & 0 & 0.08 & 3.13 & 2.32 & 0\ ) & \\ W_3 & = & (2.13 & 0 & 2.22 & 0 & 0 & 0.03) & \end{array} \tag{43}$$

Finally, we check that the factorization provides a good fit to the data. The product WH is equal to

$$\begin{array}{ccccccccc} \text{singer} & \text{GDP} & \text{senate} & \text{election} & \text{vote} & \text{stock} & \text{bass} & \text{market} & \text{band} \\ \begin{pmatrix} 6.89\ (6) \\ 0.75\ (1) \\ 7.14\ (8) \\ 0\ (0) \\ 0.53\ (0) \\ 0.86\ (1) \end{pmatrix} & \begin{pmatrix} 1.01\ (1) \\ 0\ (0) \\ 0.99\ (1) \\ 7\ (6.91) \\ 5.12\ (5) \\ 0.01\ (0) \end{pmatrix} & \begin{pmatrix} 0.53\ (1) \\ 8.32\ (9) \\ 0.44\ (0) \\ 0.67\ (1) \\ 6.45\ (6) \\ 8.36\ (8) \end{pmatrix} & \begin{pmatrix} 0.54\ (0) \\ 5.66\ (5) \\ 0.47\ (1) \\ 1.41\ (0) \\ 5.09\ (7) \\ 5.69\ (5) \end{pmatrix} & \begin{pmatrix} 0.10\ (0) \\ 8.18\ (8) \\ 0\ (0) \\ 0\ (0) \\ 5.85\ (5) \\ 8.22\ (9) \end{pmatrix} & \begin{pmatrix} 0.53\ (1) \\ 1.15\ (1) \\ 0.47\ (0) \\ 8.28\ (9) \\ 6.97\ (6) \\ 1.16\ (2) \end{pmatrix} & \begin{pmatrix} 8.83\ (9) \\ 0\ (0) \\ 9.16\ (9) \\ 0.65\ (1) \\ 0.48\ (0) \\ 0.14\ (0) \end{pmatrix} & \begin{pmatrix} 0.53\ (0) \\ 0.78\ (1) \\ 0.48\ (1) \\ 7.60\ (7) \\ 6.19\ (7) \\ 0.79\ (0) \end{pmatrix} & \begin{pmatrix} 7.36\ (8) \\ 0.78\ (0) \\ 7.62\ (7) \\ 0.69\ (0) \\ 1.07\ (2) \\ 0.9\ (1) \end{pmatrix} & \begin{matrix} \textit{Art.} \\ \text{a} \\ \text{b} \\ \text{c} \\ \text{d} \\ \text{e} \\ \text{f} \end{matrix} \end{array}$$

For ease of comparison the values of A are shown in brackets.

Convex optimization

Notation

Matrices are written in uppercase: A, vectors are written in lowercase: a. A_{ij} denotes the element of A in position (i, j), A_i denotes the ith column of A (it's a vector!). Beware that x_i may denote the ith entry of a vector x or a the ith vector in a list depending on the context. $\mathcal{I}$ denotes a subvector of x that contains the entries listed in the set $\mathcal{I}$. For example, $x_{1:n}$ contains the first n entries of x.

1 Convexity

1.1 Convex sets

A set is convex if it contains all segments connecting points that belong to it.

Definition 1.1 (Convex set)**.** *A convex set $\mathcal{S}$ is any set such that for any $x, y \in \mathcal{S}$ and $\theta \in (0, 1)$*

$$\theta x + (1 - \theta)\, y \in \mathcal{S}. \tag{1}$$

Figure 1 shows a simple example of a convex and a nonconvex set.

The following lemma establishes that the intersection of convex sets is convex.

Lemma 1.2 (Intersection of convex sets)**.** *Let $\mathcal{S}_1, \ldots, \mathcal{S}_m$ be convex subsets of $\mathbb{R}^n$, $\cap_{i=1}^m \mathcal{S}_i$ is convex.*

Proof. Any $x, y \in \cap_{i=1}^m \mathcal{S}_i$ also belong to S_1. By convexity of $\mathcal{S}_1$ $\theta x + (1 - \theta)\, y$ belongs to S_1 for any $\theta \in (0, 1)$ and therefore also to $\cap_{i=1}^m \mathcal{S}_i$. □

The following theorem shows that projection onto non-empty closed convex sets is unique. The proof is in Section B.1 of the appendix.

Theorem 1.3 (Projection onto convex set)**.** *Let $\mathcal{S} \subseteq \mathbb{R}^n$ be a non-empty closed convex set. The projection of any vector $x \in \mathbb{R}^n$ onto $\mathcal{S}$*

$$\mathcal{P}_{\mathcal{S}}(x) := \arg\min_{s \in \mathcal{S}} ||x - s||_2 \tag{2}$$

exists and is unique.

Figure 1: An example of a nonconvex set (left) and a convex set (right).

A convex combination of n points is any linear combination of the points with nonnegative coefficients that add up to one. In the case of two points, this is just the segment between the points.

Definition 1.4 (Convex combination)**.** *Given n vectors $x_1, x_2, \ldots, x_n \in \mathbb{R}^n$,*

$$x := \sum_{i=1}^{n} \theta_i x_i \tag{3}$$

is a convex combination of $x_1, x_2, \ldots, x_n$ as along as the real numbers $\theta_1, \theta_2, \ldots, \theta_n$ are nonnegative and add up to one,

$$\theta_i \geq 0, \quad 1 \leq i \leq n, \tag{4}$$

$$\sum_{i=1}^{n} \theta_i = 1. \tag{5}$$

The convex hull of a set $\mathcal{S}$ contains all convex combination of points in $\mathcal{S}$. Intuitively, it is the smallest convex set that contains $\mathcal{S}$.

Definition 1.5 (Convex hull)**.** *The convex hull of a set $\mathcal{S}$ is the set of all convex combinations of points in $\mathcal{S}$.*

A justification of why we penalize the ℓ_1-norm to promote sparse structure is that the ℓ_1-norm ball is the convex hull of the intersection between the ℓ_0 "norm" ball and the ℓ_∞-norm ball. The lemma is illustrated in 2D in Figure 2 and proved in Section 1.6 of the appendix.

Lemma 1.6 (ℓ_1-norm ball)**.** *The ℓ_1-norm ball is the convex hull of the intersection between the ℓ_0 "norm" ball and the ℓ_∞-norm ball.*

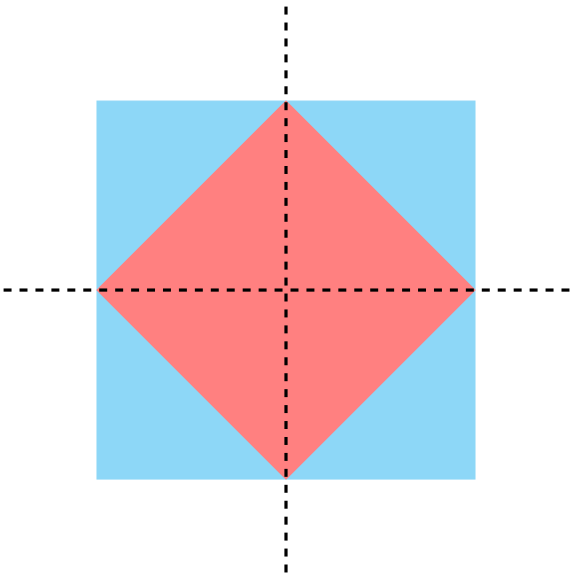


Figure 2: Illustration of Lemma (1.6) The ℓ_0 "norm" ball is shown in black, the ℓ_∞-norm ball in blue and the ℓ_1-norm ball in a reddish color.

1.2 Convex functions

We now define convexity for functions.

Definition 1.7 (Convex function)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if for any $x, y \in \mathbb{R}^n$ and any $\theta \in (0, 1)$,*

$$\theta f(x) + (1-\theta) f(y) \geq f(\theta x + (1-\theta) y). \tag{6}$$

The function is strictly convex if the inequality is always strict, i.e. if $x \neq y$ implies that

$$\theta f(x) + (1-\theta) f(y) > f(\theta x + (1-\theta) y). \tag{7}$$

A concave function is a function f such that $-f$ is convex.

Remark 1.8 (Extended-value functions)**.** *We can also consider an arbitrary function f that is only defined in a subset of $\mathbb{R}^n$. In that case f is convex if and only if its extension $\tilde{f}$ is convex, where*

$$\tilde{f}(x) := \begin{cases} f(x) & \text{if } x \in \operatorname{dom}(f), \\ \infty & \text{if } x \notin \operatorname{dom}(f). \end{cases} \tag{8}$$

Equivalently, f is convex if and only if its domain $\operatorname{dom}(f)$ is convex and any two points in $\operatorname{dom}(f)$ satisfy (6).

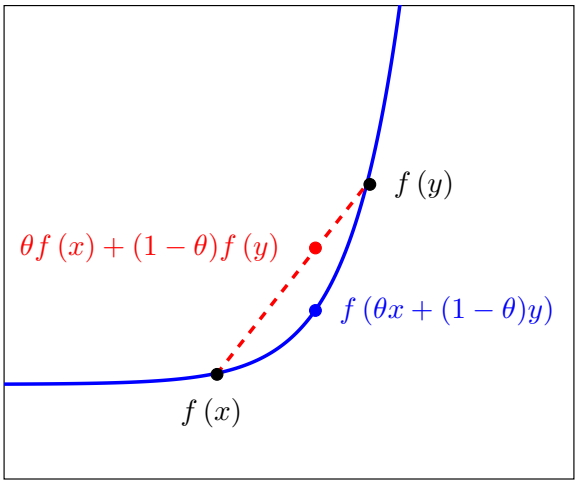


Figure 3: Illustration of condition (6) in Definition 1.7. The curve corresponding to the function must lie below any chord joining two of its points.

Condition (6) is illustrated in Figure 3. The curve corresponding to the function must lie below any chord joining two of its points. It is therefore not surprising that we can determine whether a function is convex by restricting our attention to its behavior along lines in $\mathbb{R}^n$. This is established by the following lemma, which is proved formally in Section B.2 of the appendix.

Lemma 1.9 (Equivalent definition of convex functions)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if for any two points $x, y \in \mathbb{R}^n$ the univariate function $g_{x,y} : [0,1] \to \mathbb{R}$ defined by*

$$g_{x,y}(\alpha) := f(\alpha x + (1-\alpha) y) \tag{9}$$

is convex. Similarly, f is strictly convex if and only if $g_{x,y}$ is strictly convex for any a, b.

Section A in the appendix provides a definition of the norm of a vector and lists the most common ones. It turns out that all norms are convex.

Lemma 1.10 (Norms are convex)**.** *Any valid norm $||\cdot||$ is a convex function.*

Proof. By the triangle inequality inequality and homogeneity of the norm, for any $x, y \in \mathbb{R}^n$ and any $\theta \in (0,1)$

$$||\theta x + (1-\theta) y|| \leq ||\theta x|| + ||(1-\theta) y|| = \theta ||x|| + (1-\theta) ||y|| . \tag{10}$$

□

The ℓ_0 "norm" is not really norm, as explained in Section A, and is not convex either.

Lemma 1.11 (ℓ_0 "norm")**.** *The ℓ_0 "norm" is not convex.*

Proof. We provide a simple counterexample. Let $x := \binom{1}{0}$ and $y := \binom{0}{1}$, then for any $\theta \in (0,1)$

$$||\theta x + (1-\theta)\, y||_0 = 2 > 1 = \theta\, ||x||_0 + (1-\theta)\, ||y||_0 \,. \tag{11}$$

□

We end the section by establishing a property of convex functions that is crucial in optimization.

Theorem 1.12 (Local minima are global)**.** *Any local minimum of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ is also a global minimum.*

We defer the proof of the theorem to Section B.4 of the appendix.

1.3 Sublevel sets and epigraph

In this section we define two sets associated to a function that are very useful when reasoning geometrically about convex functions.

Definition 1.13 (Sublevel set)**.** *The γ-sublevel set of a function $f : \mathbb{R}^n \to \mathbb{R}$, where $\gamma \in \mathbb{R}$, is the set of points in $\mathbb{R}^n$ at which the function is smaller or equal to γ,*

$$C_\gamma := \{x \mid f(x) \leq \gamma\} \,. \tag{12}$$

Lemma 1.14 (Sublevel sets of convex functions)**.** *The sublevel sets of a convex function are convex.*

Proof. If $x, y \in \mathbb{R}^n$ belong to the γ-sublevel set of a convex function f then for any $\theta \in (0,1)$

$$f(\theta x + (1-\theta)\, y) \leq \theta f(x) + (1-\theta)\, f(y) \quad \text{by convexity of } f \tag{13}$$

$$\leq \gamma \tag{14}$$

because both x and y belong to the γ-sublevel set. We conclude that any convex combination of x and y also belongs to the γ-sublevel set. □

Recall that the graph of a function $f : \mathbb{R}^n \to \mathbb{R}$ is the curve in $\mathbb{R}^{n+1}$

$$\text{graph}(f) := \{x \mid f(x_{1:n}) = x_{n+1}\} \,, \tag{15}$$

where $x_{1:n} \in \mathbb{R}^n$ contains the first n entries of x. The epigraph of a function is the set in $\mathbb{R}^{n+1}$ that lies above the graph of the function. An example is shown in Figure 4.

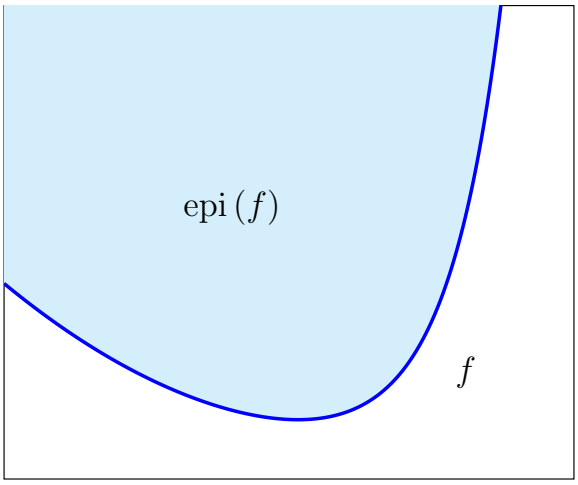


Figure 4: Epigraph of a function.

Definition 1.15 (Epigraph)**.** *The epigraph of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as*

$$\operatorname{epi}(f) := \{x \mid f(x_{1:n}) \leq x_{n+1}\}. \tag{16}$$

Epigraphs allow to reason geometrically about convex functions. The following basic result is proved in Section B.5 of the appendix.

Lemma 1.16 (Epigraphs of convex functions are convex)**.** *A function is convex if and only if its epigraph is convex.*

1.4 Operations that preserve convexity

It may be challenging to determine whether a function of interest is convex or not by using the definition directly. Often, an easier alternative is to express the function in terms of simpler functions that are known to be convex. In this section we list some operations that preserve convexity.

Lemma 1.17 (Composition of convex and affine function)**.** *If $f : \mathbb{R}^n \to \mathbb{R}$ is convex, then for any $A \in \mathbb{R}^{n \times m}$ and any $b \in \mathbb{R}^n$, the function*

$$h(x) := f(Ax + b) \tag{17}$$

is convex.

Proof. By convexity of f, for any $x, y \in \mathbb{R}^m$ and any $\theta \in (0,1)$

$$h\left(\theta x + (1-\theta)\, y\right) = f\left(\theta\left(Ax+b\right) + (1-\theta)\left(Ay+b\right)\right) \tag{18}$$
$$\leq \theta f\left(Ax+b\right) + (1-\theta)\, f\left(Ay+b\right) \tag{19}$$
$$= \theta\, h\left(x\right) + (1-\theta)\, h\left(y\right). \tag{20}$$

□

Corollary 1.18 (Least squares)**.** *For any $A \in \mathbb{R}^{n\times m}$ and any $y \in \mathbb{R}^n$ the least-squares cost function*

$$||Ax - y||_2 \tag{21}$$

is convex.

Lemma 1.19 (Nonnegative weighted sums)**.** *The weighted sum of m convex functions $f_1, \ldots, f_m$*

$$f := \sum_{i=1}^{m} \alpha_i\, f_i \tag{22}$$

is convex as long as the weights $\alpha_1, \ldots, \alpha \in \mathbb{R}$ are nonnegative.

Proof. By convexity of $f_1, \ldots, f_m$, for any $x, y \in \mathbb{R}^m$ and any $\theta \in (0,1)$

$$f\left(\theta x + (1-\theta)\, y\right) = \sum_{i=1}^{m} \alpha_i\, f_i\left(\theta x + (1-\theta)\, y\right) \tag{23}$$
$$\leq \sum_{i=1}^{m} \alpha_i \left(\theta f_i\left(x\right) + (1-\theta)\, f_i\left(y\right)\right) \tag{24}$$
$$= \theta\, f\left(x\right) + (1-\theta)\, f\left(y\right). \tag{25}$$

□

Corollary 1.20 (Regularized least squares)**.** *Regularized least-squares cost functions of the form*

$$||Ax - y||_2^2 + ||x||, \tag{26}$$

where $||\cdot||$ is an arbitrary norm, are convex.

Proposition 1.21 (Pointwise maximum/supremum of convex functions)**.** *The pointwise maximum of m convex functions $f_1, \ldots, f_m$ is convex*

$$f_{\max}\left(x\right) := \max_{1\leq i\leq m} f_i\left(x\right). \tag{27}$$

The pointwise supremum of a family of convex functions indexed by a set $\mathcal{I}$

$$f_{\sup}(x) := \sup_{i \in \mathcal{I}} f_i(x). \tag{28}$$

is convex.

Proof. We prove that the supremum is unique, as it implies the result for the maximum. For any $0 \leq \theta \leq 1$ and any $x, y \in \mathbb{R}$,

$$\begin{aligned} f_{\sup}(\theta x + (1-\theta) y) &= \sup_{i \in \mathcal{I}} f_i(\theta x + (1-\theta) y) && (29)\\ &\leq \sup_{i \in \mathcal{I}} \theta f_i(x) + (1-\theta) f_i(y) \quad \text{by convexity of the } f_i && (30)\\ &\leq \theta \sup_{i \in \mathcal{I}} f_i(x) + (1-\theta) \sup_{j \in \mathcal{I}} f_j(y) && (31)\\ &= \theta f_{\sup}(x) + (1-\theta) f_{\sup}(y). && (32) \end{aligned}$$

□

2 Differentiable functions

In this section we characterize the convexity of differentiable functions in terms of the behavior of their first and second order Taylor expansions, or equivalently in terms of their gradient and Hessian.

2.1 First-order conditions

Consider the first-order Taylor expansion of $f : \mathbb{R}^n \to \mathbb{R}$ at x,

$$f_x^1(y) := f(x) + \nabla f(x)(y - x). \tag{33}$$

Note that this first-order approximation is a linear function. The following proposition, proved in Section B.6 of the appendix, establishes that a function f is convex if and only if f_x^1 is a lower bound for f for any $x \in \mathbb{R}^n$. Figure 5 illustrates the condition with an example.

Proposition 2.1 (First-order condition)**.** *A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if for every $x, y \in \mathbb{R}^n$*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x). \tag{34}$$

It is strictly convex if and only if

$$f(y) > f(x) + \nabla f(x)^T (y - x). \tag{35}$$

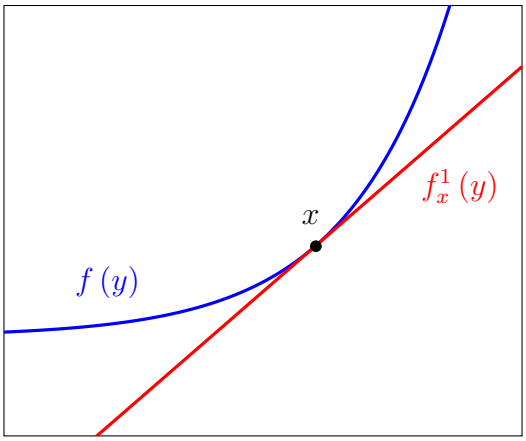


Figure 5: An example of the first-order condition for convexity. The first-order approximation at any point is a lower bound of the function.

An immediate corollary is that for a convex function, any point at which the gradient is zero is a global minimum. If the function is strictly convex, the minimum is unique.

Corollary 2.2. *If a differentiable function f is convex and $\nabla f(x) = 0$, then for any $y \in \mathbb{R}$*

$$f(y) \geq f(x). \tag{36}$$

If f is strictly convex then for any $y \in \mathbb{R}$

$$f(y) > f(x). \tag{37}$$

For any differentiable function f and any $x \in \mathbb{R}^n$ let us define the hyperplane $\mathcal{H}_{f,x} \subset \mathbb{R}^{n+1}$ that corresponds to the first-order approximation of f at x,

$$\mathcal{H}_{f,x} := \left\{ y \mid y_{n+1} = f_x^1(y_{1:n}) \right\}. \tag{38}$$

Geometrically, Proposition 2.1 establishes that $\mathcal{H}_{f,x}$ lies above the epigraph of f. In addition, the hyperplane and $\text{epi}(f)$ intersect at x. In convex analysis jargon, $\mathcal{H}_{f,x}$ is a supporting hyperplane of $\text{epi}(f)$ at x.

Definition 2.3 (Supporting hyperplane)**.** *A hyperplane $\mathcal{H}$ is a supporting hyperplane of a set $\mathcal{S}$ at x if*

- *$\mathcal{H}$ and $\mathcal{S}$ intersect at x,*
- *$\mathcal{S}$ is contained in one of the half-spaces bounded by $\mathcal{H}$.*

The optimality condition has a very intuitive geometric interpretation in terms of the supporting hyperplane $\mathcal{H}_{f,x}$. $\nabla f = 0$ implies that $\mathcal{H}_{f,x}$ is horizontal if the vertical dimension corresponds to the $n+1$th coordinate. Since the epigraph lies above hyperplane, the point at which they intersect must be a minimum of the function.

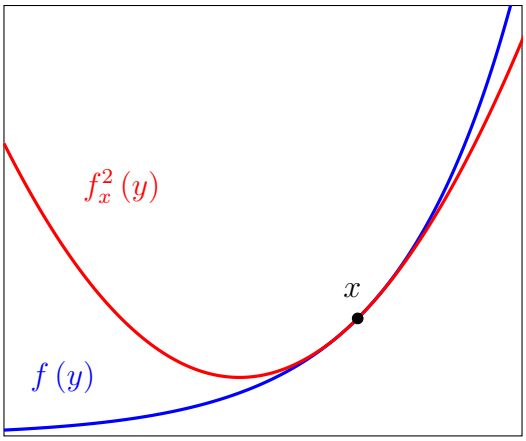


Figure 6: An example of the second-order condition for convexity. The second-order approximation at any point is convex.

2.2 Second-order conditions

For univariate functions that are twice differentiable, convexity is dictated by the curvature of the function. As you might recall from basic calculus, curvature is the rate of change of the slope of the function and is consequently given by its second derivative. The following lemma, proved in Section B.8 of the appendix, establishes that univariate functions are convex if and only if their curvature is always nonnegative.

Lemma 2.4. *A twice-differentiable function $g : \mathbb{R} \to \mathbb{R}$ is convex if and only if $g''(\alpha) \geq 0$ for all $\alpha \in \mathbb{R}$.*

By Lemma 1.9, we can establish convexity in $\mathbb{R}^n$ by considering the restriction of the function along an arbitrary line. By multivariable calculus, the second directional derivative of f at a point x in the direction of a unit vector is equal to $u^T \nabla^2 f(x) u$. As a result, if the Hessian is positive semidefinite, the curvatures is nonnegative in every direction and the function is convex.

Corollary 2.5. *A twice-differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if for every $x \in \mathbb{R}^n$, the Hessian matrix $\nabla^2 f(x)$ is positive semidefinite.*

Proof. By Lemma 1.9 we just need to show that the univariate function $g_{a,b}$ defined by (9) is convex for all $a, b \in \mathbb{R}^n$. By Lemma 2.4 this holds if and only if the second derivative of $g_{a,b}$ is nonnegative. Applying some basic multivariate calculus, we have that for any $\alpha \in (0, 1)$

$$g''_{a,b}(\alpha) := (a-b)^T \nabla^2 f(\alpha a + (1-\alpha) b)(a-b). \tag{39}$$

This quantity is nonnegative for all $a, b \in \mathbb{R}^n$ if and only if $\nabla^2 f(x)$ is positive semidefinite for any $x \in \mathbb{R}^n$. □

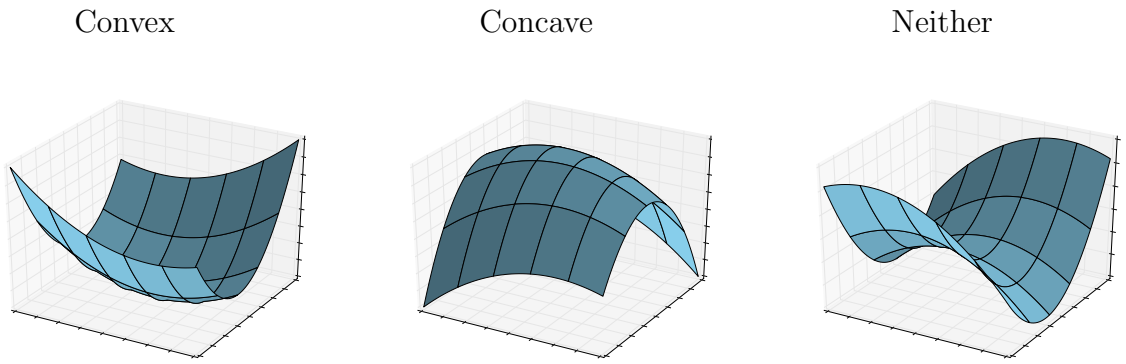


Figure 7: Quadratic forms for which the Hessian is positive definite (left), negative definite (center) and neither positive nor negative definite (right).

Remark 2.6 (Strict convexity)**.** *If the Hessian is positive definite, then the function is strictly convex (the proof is essentially the same). However, there are functions that are strictly convex for which the Hessian may equal zero at some points. An example is the univariate function* $f(x) = x^4$*, for which* $f''(0) = 0$*.*

We can interpret Corollary 2.5 in terms of the second-order Taylor expansion of $f : \mathbb{R}^n \to \mathbb{R}$ at x.

Definition 2.7 (Second-order approximation)**.** *The second-order or quadratic approximation of* f *at* x *is*

$$f_x^2(y) := f(x) + \nabla f(x)(y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x). \tag{40}$$

f_x^2 is a quadratic form that shares the same value at x. By the corollary, f is convex if and only if this quadratic approximation is always convex. This is illustrated in Figure 6. Figure 7 shows some examples of quadratic forms in two dimensions.

3 Nondifferentiable functions

If a function is not differentiable we cannot use its gradient to check whether it is convex or not. However, we can extend the first-order characterization derived in Section 2.1 by checking whether the function has a supporting hyperplane at every point. If a supporting hyperplane exists at x, the gradient of the hyperplane is called a subgradient of f at x.

Definition 3.1 (Subgradient)**.** *The subgradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ at $x \in \mathbb{R}^n$ is a vector $q \in \mathbb{R}^n$ such that*

$$f(y) \geq f(x) + q^T (y - x), \quad \text{for all } y \in \mathbb{R}^n. \tag{41}$$

The set of all subgradients is called the subdifferential of the function at x.

The following theorem, proved in Section B.9 of the appendix, establishes that if a subgradient exists at every point, the function is convex.

Theorem 3.2. *If a function $f : \mathbb{R}^n \to \mathbb{R}$ has a non-empty subdifferential at any $x \in \mathbb{R}^n$ then f is convex.*

The subdifferential allows to obtain an optimality condition for nondifferentiable convex functions.

Proposition 3.3 (Optimality condition)**.** *A convex function attains its minimum value at a vector x if the zero vector is a subgradient of f at x.*

Proof. By the definition of subgradient, if $q := 0$ is a subgradient at x for any $y \in \mathbb{R}^n$

$$f(y) \geq f(x) + q^T (y - x) = f(x). \tag{42}$$

□

If a function is differentiable at a point, then the gradient is the only subgradient at that point.

Proposition 3.4 (Subdifferential of differentiable functions)**.** *If a convex function $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable at $x \in \mathbb{R}^n$, then its subdifferential at x only contains $\nabla f(x)$.*

Proof. By Proposition 2.1 ∇f is a subgradient at x. Now, let q be an arbitrary subgradient at x. By the definition of subgradient,

$$\begin{aligned} f(x + \alpha e_i) &\geq f(x) + q^T \alpha e_i && (43)\\ &= f(x) + q_i \alpha, && (44)\\ f(x) &\geq f(x - \alpha e_i) + q^T \alpha e_i && (45)\\ &= f(x - \alpha e_i) + q_i \alpha. && (46) \end{aligned}$$

Combining both inequalities

$$\frac{f(x) - f(x - \alpha e_i)}{\alpha} \leq q_i \leq \frac{f(x + \alpha e_i) - f(x)}{\alpha}. \tag{47}$$

If we let $\alpha \to 0$, this implies $q_i = \frac{\partial f(x)}{\partial x_i}$. Consequently, $q = \nabla f$. □

An important nondifferentiable convex function in optimization-based data analysis is the ℓ_1 norm. The following proposition characterizes its subdifferential.

Proposition 3.5 (Subdifferential of ℓ_1 norm)**.** *The subdifferential of the ℓ_1 norm at $x \in \mathbb{R}^n$ is the set of vectors $q \in \mathbb{R}^n$ that satisfy*

$$q_i = \text{sign}\,(x_i) \quad \text{if } x_i \neq 0, \tag{48}$$

$$|q_i| \leq 1 \quad \text{if } x_i = 0. \tag{49}$$

Proof. The proof relies on the following simple lemma, proved in Section B.10.

Lemma 3.6. *q is a subgradient of $||\cdot||_1 : \mathbb{R}^n \to \mathbb{R}$ at x if and only if q_i is a subgradient of $|\cdot| : \mathbb{R} \to \mathbb{R}$ at x_i for all $1 \leq i \leq n$.*

If $x_i \neq 0$ the absolute-value function is differentiable at x_i, so by Proposition (3.4), q_i is equal to the derivative $q_i = \text{sign}\,(x_i)$.

If $x_i = 0$, q_i is a subgradient of the absolute-value function if and only if $|\alpha| \geq q_i\,\alpha$ for any $\alpha \in \mathbb{R}$, which holds if and only if $|q_i| \leq 1$. □

4 Optimization problems

4.1 Definition

We start by defining a canonical optimization problem. The vast majority of optimization problems (certainly all of the ones that we will study in the course) can be cast in this form.

Definition 4.1 (Optimization problem)**.**

$$\text{minimize} \quad f_0\,(x) \tag{50}$$

$$\text{subject to} \quad f_i\,(x) \leq 0, \quad 1 \leq i \leq m, \tag{51}$$

$$h_i\,(x) = 0, \quad 1 \leq i \leq p, \tag{52}$$

where $f_0, f_1, \ldots, f_m, h_1, \ldots, h_p : \mathbb{R}^n \to \mathbb{R}$.

The problem consists of a cost function f_0, inequality constraints and equality constraints. Any vector that satisfies all the constraints in the problem is said to be feasible. A solution to the problem is any vector x^* such that for all feasible vectors x

$$f_0\,(x) \geq f_0\,(x^*)\,. \tag{53}$$

If a solution exists $f\,(x^*)$ is the optimal value or optimum of the optimization problem.

An optimization problem is convex if it satisfies the following conditions:

- The cost function f_0 is convex.
- The functions that determine the inequality constraints $f_1, \ldots, f_m$ are convex.
- The functions that determine the equality constraints $h_1, \ldots, h_p$ are affine, i.e. $h_i(x) = a_i^T x + b_i$ for some $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$.

Note that under these assumptions the feasibility set is convex. Indeed, it corresponds to the intersection of several convex sets: the 0-sublevel sets of $f_1, \ldots, f_m$, which are convex by Lemma 1.14, and the hyperplanes $h_i(x) = a_i^T x + b_i$. The intersection is convex by Lemma 1.2.

If both the cost function and the constraint functions are all affine, the problem is a linear program (LP).

Definition 4.2 (Linear program)**.**

$$\begin{aligned} \text{minimize} \quad & a^T x && (54)\\ \text{subject to} \quad & c_i^T x \leq d_i, \quad 1 \leq i \leq m, && (55)\\ & Ax = b. && (56) \end{aligned}$$

It turns out that ℓ_1-norm minimization can be cast as an LP. The theorem is proved in Section B.11 of the appendix.

Theorem 4.3 (ℓ_1-norm minimization as an LP)**.** *The optimization problem*

$$\begin{aligned} \text{minimize} \quad & ||x||_1 && (57)\\ \text{subject to} \quad & Ax = b && (58) \end{aligned}$$

can be recast as the linear program

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} t_i && (59)\\ \text{subject to} \quad & t_i \geq x_i, && (60)\\ & t_i \geq -x_i, && (61)\\ & Ax = b. && (62) \end{aligned}$$

If the cost function is a positive semidefinite quadratic form and the constraints are affine the problem is a quadratic program (QP).

Definition 4.4 (Quadratic program)**.**

$$\begin{aligned} \text{minimize} \quad & x^T Q x + a^T x && (63)\\ \text{subject to} \quad & c_i^T x \leq d_i, \quad 1 \leq i \leq m, && (64)\\ & Ax = b, && (65) \end{aligned}$$

where $Q \in \mathbb{R}^{n \times n}$ is positive semidefinite.

A corollary of Theorem 4.3 is that ℓ_1-norm regularized least squares can be cast as a QP.

Corollary 4.5 (ℓ_1-norm regularized least squares as a QP)**.** *The optimization problem*

$$\text{minimize} \quad ||Ax - y||_2^2 + \lambda \, ||x||_1 \tag{66}$$

can be recast as the quadratic program

$$\begin{aligned} \text{minimize} \quad & x^T A^T A x - 2 y^T x + \lambda \sum_{i=1}^{n} t_i && (67) \\ \text{subject to} \quad & t_i \geq x_i, && (68) \\ & t_i \geq -x_i. && (69) \end{aligned}$$

We will discuss other types of convex optimization problems, such as semidefinite programs later on in the course.

4.2 Duality

The Lagrangian of the optimization problem in Definition 4.1 is defined as the cost function augmented by a weighted linear combination of the constraint functions,

$$L\left(x, \lambda, \nu\right) := f_0\left(x\right) + \sum_{i=1}^{m} \lambda_i \, f_i\left(x\right) + \sum_{j=1}^{p} \nu_j \, h_j\left(x\right), \tag{70}$$

where the vectors $\lambda \in \mathbb{R}^m, \nu \in \mathbb{R}^p$ are called Lagrange multipliers or dual variables. In contrast, x is the primal variable.

Note that as long as $\lambda_i \geq 0$ for $1 \leq i \leq m$, the Lagrangian is a lower bound for the value of the cost function at any feasible point. Indeed, if x is feasible and $\lambda_i \geq 0$ for $1 \leq i \leq m$ then

$$\begin{aligned} \lambda_i \, f_i\left(x\right) &\leq 0, && (71) \\ \nu_j \, h_j\left(x\right) &= 0, && (72) \end{aligned}$$

for all $1 \leq i \leq m, 1 \leq j \leq p$. This immediately implies,

$$L\left(x, \lambda, \nu\right) \leq f_0\left(x\right). \tag{73}$$

The Lagrange dual function is the infimum of the Lagrangian over the primal variable x

$$l\left(\lambda, \nu\right) := \inf_{x \in \mathbb{R}^n} f_0\left(x\right) + \sum_{i=1}^{m} \lambda_i \, f_i\left(x\right) + \sum_{j=1}^{p} \nu_j \, h_j\left(x\right). \tag{74}$$

Proposition 4.6 (Lagrange dual function as a lower bound of the primal optimum)**.** *Let p^* denote an optimal value of the optimization problem in Definition 4.1,*

$$l(\lambda, \nu) \leq p^*, \tag{75}$$

as long as $\lambda_i \geq 0$ for $1 \leq i \leq n$.

Proof. The result follows directly from (73),

$$\begin{aligned} p^* &= f_0(x^*) && (76)\\ &\geq L(x^*, \lambda, \nu) && (77)\\ &\geq l(\lambda, \nu). && (78) \end{aligned}$$

□

Optimizing this lower bound on the primal optimum over the Lagrange multipliers yields the dual problem of the original optimization problem, which is called the primal problem in this context.

Definition 4.7 (Dual problem)**.** *The dual problem of the optimization problem from Definition 4.1 is*

$$\begin{aligned} &\text{maximize} && l(\lambda, \nu) && (79)\\ &\text{subject to} && \lambda_i \geq 0, \quad 1 \leq i \leq m. && (80) \end{aligned}$$

Note that the cost function is a pointwise supremum of linear (and hence convex) functions, so by Proposition 1.21 the dual problem is a convex optimization problem even if the primal is nonconvex! The following result, which is an immediate corollary to Proposition 4.6, states that the optimum of the dual problem is a lower bound for the primal optimum. This is known as weak duality.

Corollary 4.8 (Weak duality)**.** *Let p^* denote an optimum of the optimization problem in Definition 4.1 and d^* an optimum of the corresponding dual problem,*

$$d^* \leq p^*. \tag{81}$$

In the case of convex functions, the optima of the primal and dual problems are often equal, i.e.

$$d^* = p^*. \tag{82}$$

This is known as strong duality. A simple condition that guarantees strong duality for convex optimization problems is Slater's condition.

Definition 4.9 (Slater's condition)**.** *A vector $x \in \mathbb{R}^n$ satisfies Slater's condition for a convex optimization problem if*

$$f_i(x) < 0, \quad 1 \leq i \leq m, \tag{83}$$

$$Ax = b. \tag{84}$$

A proof of strong duality under Slater's condition can be found in Section 5.3.2 of [1].

References

A very readable and exhaustive reference is Boyd and Vandenberghe's seminal book on convex optimization [1], which unfortunately does not cover subgradients.[1] Nesterov's book [2] and Rockafellar's book [3] do cover subgradients.

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[2] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*.

[3] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*. Springer Science & Business Media, 2009.

A Norms

The norm of a vector is a generalization of the concept of *length*.

Definition A.1 (Norm)**.** *Let $\mathcal{V}$ be a vector space, a norm is a function $||\cdot||$ from $\mathcal{V}$ to $\mathbb{R}$ that satisfies the following conditions.*

- *It is homogeneous. For all $\alpha \in \mathbb{R}$ and $x \in \mathcal{V}$*

$$||\alpha\, x|| = |\alpha|\, ||x||\,. \tag{85}$$

- *It satisfies the triangle inequality*

$$||x + y|| \leq ||x|| + ||y||\,. \tag{86}$$

  In particular, it is nonnegative (set $y = -x$).

- $||x|| = 0$ *implies that x is the zero vector* 0.

[1] However, see http://see.stanford.edu/materials/lsocoee364b/01-subgradients_notes.pdf

The ℓ_2 norm is induced by the inner product $\langle x, y\rangle = x^T y$

$$||x||_2 := \sqrt{x^T x}. \tag{87}$$

Definition A.2 (ℓ_2 norm)**.** *The ℓ_2 norm of a vector $x \in \mathbb{R}^n$ is defined as*

$$||x||_2 := \sqrt{\sum_{i=1}^{n} x_i^2}. \tag{88}$$

Definition A.3 (ℓ_1 norm)**.** *The ℓ_1 norm of a vector $x \in \mathbb{R}^n$ is defined as*

$$||x||_1 := \sum_{i=1}^{n} |x_i| . \tag{89}$$

Definition A.4 (ℓ_∞ norm)**.** *The ℓ_∞ norm of a vector $x \in \mathbb{R}^n$ is defined as*

$$||x||_\infty := \max_{1 \le i \le n} |x_i| . \tag{90}$$

Remark A.5. *The ℓ_0 "norm" is not a norm, as it is not homogeneous. For example, if x is not the zero vector,*

$$||2x||_0 = ||x||_0 \neq 2\,||x||_0 . \tag{91}$$

B Proofs

B.1 Proof of Theorem 1.3

Existence

Since $\mathcal{S}$ is non-empty we can choose an arbitrary point $\tilde{s} \in \mathcal{S}$. Minimizing $||x - s||_2$ over $\mathcal{S}$ is equivalent to minimizing $||x - s||_2$ over $\mathcal{S} \cap \{y \mid ||x - y||_2 \le ||x - \tilde{s}||_2\}$. Indeed, the solution cannot be a point that is farther away from x than $\tilde{s}$. By Weierstrass's extreme-value theorem, the optimization problem

$$\begin{aligned} &\text{minimize} && ||x - s||_2^2 && (92)\\ &\text{subject to} && s \in \mathcal{S} \cap \{y \mid ||x - y||_2 \le ||x - \tilde{s}||_2\} && (93) \end{aligned}$$

has a solution because $||x - s||_2^2$ is a continuous function and the feasibility set is bounded and closed, and hence compact. Note that this also holds if $\mathcal{S}$ is not convex.

Uniqueness

Assume that there are two distinct projections $s_1 \neq s_2$. Consider the point

$$s := \frac{s_1 + s_2}{2}, \tag{94}$$

which belongs to $\mathcal{S}$ because $\mathcal{S}$ is convex. The difference between x and s and the difference between s_1 and s are orthogonal vectors,

$$\begin{aligned}
\langle x - s, s_1 - s \rangle &= \left\langle x - \frac{s_1 + s_2}{2}, s_1 - \frac{s_1 + s_2}{2} \right\rangle && (95)\\
&= \left\langle \frac{x - s_1}{2} + \frac{x - s_2}{2}, \frac{x - s_1}{2} - \frac{x - s_2}{2} \right\rangle && (96)\\
&= \frac{1}{4}\left(||x - s_1||^2 + ||x - s_2||^2 \right) && (97)\\
&= 0, && (98)
\end{aligned}$$

because $||x - s_1|| = ||x - s_2||$ by assumption. By Pythagoras's theorem this implies

$$\begin{aligned}
||x - s_1||_2^2 &= ||x - s||_2^2 + ||s_1 - s||_2^2 && (99)\\
&= ||x - s||_2^2 + \left|\left| \frac{s_1 - s_2}{2} \right|\right|_2^2 && (100)\\
&> ||x - s||_2^2 && (101)
\end{aligned}$$

because $s_1 \neq s_2$ by assumption. We have reached a contradiction, so the projection is unique.

B.2 Proof of Lemma 1.9

The proof for strict convexity is exactly the same, replacing the inequalities by strict inequalities.

f being convex implies that $g_{x,y}$ is convex for any $x, y \in \mathbb{R}^n$

For any $\alpha, \beta, \theta \in (0, 1)$

$$\begin{aligned}
g_{x,y}\left(\theta\alpha + (1-\theta)\beta\right) &= f\left(\left(\theta\alpha + (1-\theta)\beta\right)x + \left(1 - \theta\alpha - (1-\theta)\beta\right)y\right) && (102)\\
&= f\left(\theta\left(\alpha x + (1-\alpha)y\right) + (1-\theta)\left(\beta x + (1-\beta)y\right)\right) && (103)\\
&\leq \theta f\left(\alpha x + (1-\alpha)y\right) + (1-\theta) f\left(\beta x + (1-\beta)y\right) \quad \text{by convexity of } f\\
&= \theta g_{x,y}(\alpha) + (1-\theta) g_{x,y}(\beta). && (104)
\end{aligned}$$

$g_{x,y}$ being convex for any $x, y \in \mathbb{R}^n$ implies that f is convex

For any $\alpha, \beta, \theta \in (0,1)$

$$
\begin{aligned}
f\left(\theta x + (1-\theta)\, y\right) &= g_{x,y}\left(\theta\right) && (105)\\
&\leq \theta g_{x,y}\left(1\right) + (1-\theta)\, g_{x,y}\left(0\right) \quad \text{by convexity of } g_{x,y} && (106)\\
&= \theta f\left(x\right) + (1-\theta)\, f\left(y\right). && (107)
\end{aligned}
$$

B.3 Proof of Lemma 1.6

We prove that the ℓ_1-norm ball $\mathcal{B}_{\ell_1}$ is equal to the convex hull of the intersection between the ℓ_0 "norm" ball $\mathcal{B}_{\ell_0}$ and the ℓ_∞-norm ball $\mathcal{B}_{\ell_\infty}$ by showing that the sets contain each other.

$\underline{\mathcal{B}_{\ell_1} \subseteq \mathcal{C}\left(\mathcal{B}_{\ell_0} \cap \mathcal{B}_{\ell_\infty}\right)}$

Let x be an n-dimensional vector in $\mathcal{B}_{\ell_1}$. If we set $\theta_i := |x(i)|$, where $x\left(i\right)$ is the ith entry of x by $x\left(i\right)$, and $\theta_0 = 1 - \sum_{i=1}^{n} \theta_i$ we have

$$
\begin{aligned}
\sum_{i=1}^{n+1} \theta_i &= 1, && (108)\\
\theta_i &\geq 0 \quad \text{for } 1 \leq i \leq n \text{ by definition}, && (109)\\
\theta_0 &= 1 - \sum_{i=1}^{n+1} \theta_i && (110)\\
&= 1 - ||x||_1 && (111)\\
&\geq 0 \quad \text{because } x \in \mathcal{B}_{\ell_1}. && (112)
\end{aligned}
$$

We can express x as a convex combination of the standard basis vectors $e_1, e_2, \ldots, e_n$, which belong to $\mathcal{B}_{\ell_0} \cap \mathcal{B}_{\ell_\infty}$ since they have a single nonzero entry equal to one, and the zero vector e_0, which also belongs to $\mathcal{B}_{\ell_0} \cap \mathcal{B}_{\ell_\infty}$,

$$
x = \sum_{i=1}^{n} \theta_i e_i + \theta_0 e_0. \tag{113}
$$

$\underline{\mathcal{C}\left(\mathcal{B}_{\ell_0} \cap \mathcal{B}_{\ell_\infty}\right) \subseteq \mathcal{B}_{\ell_1}}$

Let x be an n-dimensional vector in $\mathcal{C}\left(\mathcal{B}_{\ell_0} \cap \mathcal{B}_{\ell_\infty}\right)$. By the definition of convex hull, we can write

$$
x = \sum_{i=1}^{m} \theta_i y_i, \tag{114}
$$

where $m > 0$, $y_1, \ldots, y_m \in \mathbb{R}^n$ have a single entry bounded by one, $\theta_i \geq 0$ for all $1 \leq i \leq m$ and $\sum_{i=1}^m \theta_i = 1$. This immediately implies $x \in \mathcal{B}_{\ell_1}$, since

$$
\begin{aligned}
||x||_1 &\leq \sum_{i=1}^m \theta_i \, ||y_i||_1 \quad \text{by the Triangle inequality} && (115)\\
&\leq \sum_{i=1}^n \theta_i \, ||y_i||_\infty \quad \text{because each } y_i \text{ only has one nonzero entry} && (116)\\
&\leq \sum_{i=1}^n \theta_i && (117)\\
&\leq 1. && (118)
\end{aligned}
$$

B.4 Proof of Theorem 1.12

We prove the result by contradiction. Let x_{loc} be a local minimum and x_{glob} a global minimum such that $f(x_{\text{glob}}) < f(x_{\text{loc}})$. Since x_{loc} is a local minimum, there exists $\gamma > 0$ for which $f(x_{\text{loc}}) \leq f(x)$ for all $x \in \mathbb{R}^n$ such that $||x - x_{\text{loc}}||_2 \leq \gamma$. If we choose $\theta \in (0, 1)$ small enough, $x_\theta := \theta x_{\text{loc}} + (1 - \theta) x_{\text{glob}}$ satisfies $||x - x_{\text{loc}}||_2 \leq \gamma$ and therefore

$$
\begin{aligned}
f(x_{\text{loc}}) &\leq f(x_\theta) && (119)\\
&\leq \theta f(x_{\text{loc}}) + (1 - \theta) f(x_{\text{glob}}) \quad \text{by convexity of } f && (120)\\
&< f(x_{\text{loc}}) \quad \text{because } f(x_{\text{glob}}) < f(x_{\text{loc}}). && (121)
\end{aligned}
$$

B.5 Proof of Lemma 1.16

f being convex implies that $\text{epi}(f)$ is convex

Let $x, y \in \mathbb{R}^{n+1} \in \text{epi}(f)$, then for any $\theta \in (0, 1)$

$$
\begin{aligned}
f(\theta x_{1:n} + (1 - \theta) y_{1:n}) &\leq \theta f(x_{1:n}) + (1 - \theta) f(y_{1:n}) \quad \text{by convexity of } f && (122)\\
&\leq \theta x_{n+1} + (1 - \theta) y_{n+1} && (123)
\end{aligned}
$$

because $x, y \in \mathbb{R}^{n+1} \in \text{epi}(f)$ so $f(x_{1:n}) \leq x_{n+1}$ and $f(y_{1:n}) \leq y_{n+1}$. This implies that $\theta x + (1 - \theta) y \in \text{epi}(f)$.

$\text{epi}(f)$ being convex implies that f is convex

For any $x, y \in \mathbb{R}^n$, let us define $\tilde{x}, \tilde{y} \in \mathbb{R}^{n+1}$ such that

$$
\begin{aligned}
\tilde{x}_{1:n} := x, \quad \tilde{x}_{n+1} := f(x), && (124)\\
\tilde{y}_{1:n} := y, \quad \tilde{y}_{n+1} := f(y). && (125)
\end{aligned}
$$

By definition of $\operatorname{epi}(f)$, $\tilde{x}, \tilde{y} \in \operatorname{epi}(f)$. For any $\theta \in (0,1)$, $\theta\tilde{x} + (1-\theta)\,\tilde{y}$ belongs to $\operatorname{epi}(f)$ because it is convex. As a result,

$$\begin{aligned} f(\theta x + (1-\theta)\,y) &= f(\theta\tilde{x}_{1:n} + (1-\theta)\,\tilde{y}_{1:n}) && (126)\\ &\leq \theta\tilde{x}_{n+1} + (1-\theta)\,\tilde{y}_{n+1} && (127)\\ &= \theta f(x) + (1-\theta)\,f(y)\,. && (128) \end{aligned}$$

B.6 Proof of Proposition 2.1

The proof for strict convexity is almost exactly the same; we omit the details.

The following lemma, proved in Section B.7 below establishes that the result holds for univariate functions

Lemma B.1. *A univariate differentiable function $g : \mathbb{R} \to \mathbb{R}$ is convex if and only if for all $\alpha, \beta \in \mathbb{R}$*

$$g(\beta) \geq g'(\alpha)\,(\beta - \alpha) \tag{129}$$

and strictly convex if and only if for all $\alpha, \beta \in \mathbb{R}$

$$g(\beta) > g'(\alpha)\,(\beta - \alpha)\,. \tag{130}$$

To complete the proof we extend the result to the multivariable case using Lemma 1.9.

If $f(y) \geq f(x) + \nabla f(x)^T (y - x)$ for any $x, y \in \mathbb{R}^n$ then f is convex

By Lemma 1.9 we just need to show that the univariate function $g_{a,b}$ defined by (9) is convex for all $a, b \in \mathbb{R}^n$. Applying some basic multivariate calculus yields

$$g'_{a,b}(\alpha) = \nabla f(\alpha\,a + (1-\alpha)\,b)^T (a - b)\,. \tag{131}$$

Let $\alpha, \beta \in \mathbb{R}$. Setting $x := \alpha\,a + (1-\alpha)\,b$ and $y := \beta\,a + (1-\beta)\,b$ we have

$$\begin{aligned} g_{a,b}(\beta) &= f(y) && (132)\\ &\geq f(x) + \nabla f(x)^T (y - x) && (133)\\ &= f(\alpha\,a + (1-\alpha)\,b) + \nabla f(\alpha\,a + (1-\alpha)\,b)^T (a - b)\,(\beta - \alpha) && (134)\\ &= g_{a,b}(\alpha) + g'_{a,b}(\alpha)\,(\beta - \alpha) \quad \text{by (131)}, && (135) \end{aligned}$$

which establishes that $g_{a,b}$ is convex by Lemma B.1 above.

If f is convex then $f(y) \geq f(x) + \nabla f(x)^T (y - x)$ for any $x, y \in \mathbb{R}^n$

By Lemma 1.9, $g_{x,y}$ is convex for any $x, y \in \mathbb{R}^n$.

$$\begin{aligned} f(y) &= g_{x,y}(1) && (136)\\ &\geq g_{x,y}(0) + g'_{x,y}(0) \quad \text{by convexity of } g_{x,y} \text{ and Lemma B.1} && (137)\\ &= f(x) + \nabla f(x)^T (y - x) \quad \text{by (131).} && (138) \end{aligned}$$

B.7 Proof of Lemma B.1

g being convex implies $g(\beta) \geq g'(\alpha)(\beta - \alpha)$ for all $\alpha, \beta \in \mathbb{R}$

If g is convex then for any $\alpha, \beta \in \mathbb{R}$ and any $0 \leq \theta \leq 1$

$$\theta(g(\beta) - g(\alpha)) + g(\alpha) \geq g(\alpha + \theta(\beta - \alpha)). \tag{139}$$

Rearranging the terms we have

$$g(\beta) \geq \frac{g(\alpha + \theta(\beta - \alpha)) - g(\alpha)}{\theta} + g(\alpha). \tag{140}$$

Setting $h = \theta(\beta - \alpha)$, this implies

$$g(\beta) \geq \frac{g(\alpha + h) - g(\alpha)}{h}(\beta - \alpha) + g(\alpha). \tag{141}$$

Taking the limit when $h \to 0$ yields

$$g(\beta) \geq g'(\alpha)(\beta - \alpha). \tag{142}$$

If $g(\beta) \geq g'(\alpha)(\beta - \alpha)$ for all $\alpha, \beta \in \mathbb{R}$ then g is convex

Let $z = \theta\alpha + (1 - \theta)\beta$, then by if $g(\beta) \geq g'(\alpha)(\beta - \alpha)$

$$\begin{aligned} g(\alpha) &\geq g'(z)(\alpha - z) + g(z) && (143)\\ &= g'(z)(1 - \theta)(\alpha - \beta) + g(z) && (144)\\ g(\beta) &\geq g'(z)(\beta - z) + g(z) && (145)\\ &= g'(z)\theta(\beta - \alpha) + g(z) && (146) \end{aligned}$$

Multiplying (144) by θ, then (146) by $1 - \theta$ and summing the inequalities, we obtain

$$\theta g(\alpha) + (1 - \theta) g(\beta) \geq g(\theta\alpha + (1 - \theta)\beta). \tag{147}$$

B.8 Proof of Lemma 2.4

The second derivative of g is nonnegative anywhere if and only if the first derivative is nondecreasing, because g'' is the derivative of g'.

If g is convex g' is nondecreasing

By Lemma B.1, if the function is convex then for any $\alpha, \beta \in \mathbb{R}$ such that $\beta > \alpha$

$$g(\alpha) \geq g'(\beta)(\alpha - \beta) + g(\beta), \tag{148}$$
$$g(\beta) \geq g'(\alpha)(\beta - \alpha) + g(\alpha). \tag{149}$$

Rearranging, we obtain

$$g'(\beta)(\beta - \alpha) \geq g(\beta) - g(\alpha) \geq g'(\alpha)(\beta - \alpha). \tag{150}$$

Since $\beta - \alpha > 0$, we have $g'(\beta) \geq g'(\alpha)$.

If g' is nondecreasing, g is convex

For arbitrary $\alpha, \beta, \theta \in \mathbb{R}$, such that $\beta > \alpha$ and $0 < \theta < 1$, let $\eta = \theta\beta + (1-\theta)\alpha$. Since $\beta > \eta > \alpha$, by the mean-value theorem there exist $\gamma_1 \in [\alpha, \eta]$ and $\gamma_2 \in [\eta, \beta]$ such that

$$g'(\gamma_1) = \frac{g(\eta) - g(\alpha)}{\eta - \alpha}, \tag{151}$$
$$g'(\gamma_2) = \frac{g(\beta) - g(\eta)}{\beta - \eta}. \tag{152}$$

Since $\gamma_1 < \gamma_2$, if g' is nondecreasing

$$\frac{g(\beta) - g(\eta)}{\beta - \eta} \geq \frac{g(\eta) - g(\alpha)}{\eta - \alpha}, \tag{153}$$

which implies

$$\frac{\eta - \alpha}{\beta - \alpha} g(\beta) + \frac{\beta - \eta}{\beta - \alpha} g(\alpha) \geq g(\eta). \tag{154}$$

Recall that $\eta = \theta\beta + (1-\theta)\alpha$, so that $\theta = (\eta - \alpha)/(\beta - \alpha)$ and $\theta = (\eta - \alpha)/(\beta - \alpha)$ and $1 - \theta = (\beta - \eta)/(\beta - \alpha)$. (154) is consequently equivalent to

$$\theta g(\beta) + (1-\theta) g(\alpha) \geq g(\theta\beta + (1-\theta)\alpha). \tag{155}$$

B.9 Proof of Theorem 3.2

For arbitrary $x, y \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$ there exists a subgradient q of f at $\alpha x + (1-\alpha)\, y$. This implies

$$\begin{aligned} f(y) &\geq f(\alpha x + (1-\alpha)\, y) + q^T (y - \alpha x - (1-\alpha)\, y) && (156)\\ &= f(\alpha x + (1-\alpha)\, y) + \alpha\, q^T (y - x), && (157)\\ f(x) &\geq f(\alpha x + (1-\alpha)\, y) + q^T (x - \alpha x - (1-\alpha)\, y) && (158)\\ &= f(\alpha x + (1-\alpha)\, y) + (1-\alpha)\, q^T (y - x). && (159) \end{aligned}$$

Multiplying equation (157) by $1-\alpha$ and equation (159) by α and adding them together yields

$$\alpha f(x) + (1-\alpha)\, f(y) \geq f(\alpha x + (1-\alpha)\, y). \tag{160}$$

B.10 Proof of Lemma 3.6

If q is a subgradient for $||\cdot||_1$ at x then q_i is a subgradient for $|\cdot|$ at $|x_i|$ for $1 \leq i \leq n$

$$\begin{aligned} |y_i| &= ||x + (y_i - x_i)\, e_i||_1 - ||x||_1 && (161)\\ &\geq q^T (y_i - x_i)\, e_i && (162)\\ &= q_i (y_i - x_i). && (163) \end{aligned}$$

If q_i is a subgradient for $|\cdot|$ at $|x_i|$ for $1 \leq i \leq n$ then q is a subgradient for $||\cdot||_1$ at x

$$\begin{aligned} ||y||_1 &= \sum_{i=1}^{n} |y_i| && (164)\\ &\geq \sum_{i=1}^{n} |x_i| + q_i (y_i - x_i) && (165)\\ &= ||x||_1 + q^T (y - x). && (166) \end{aligned}$$

B.11 Proof of Theorem 4.3

To show that the linear problem and the ℓ_1-norm minimization problem are equivalent, we show that they have the same set of solutions.

Let us denote an arbitrary solution of the LP by $\left(x^{\mathrm{lp}}, t^{\mathrm{lp}}\right)$. For any solution x^{ℓ_1} of the ℓ_1-norm minimization problem, we define $t_i^{\ell_1} := \left|x_i^{\ell_1}\right|$. $\left(x^{\ell_1}, t^{\ell_1}\right)$ is feasible for the LP so

$$\left|\left|x^{\ell_1}\right|\right|_1 = \sum_{i=1}^{n} t_i^{\ell_1} \tag{167}$$

$$\geq \sum_{i=1}^{n} t_i^{\mathrm{lp}} \quad \text{by optimality of } t^{\mathrm{lp}} \tag{168}$$

$$\geq \left|\left|x^{\mathrm{lp}}\right|\right|_1 \quad \text{by constraints (60) and (61).} \tag{169}$$

This implies that any solution of the LP is also a solution of the ℓ_1-norm minimization problem.

To prove the converse, we fix a solution x^{ℓ_1} of the ℓ_1-norm minimization problem. Setting $t_i^{\ell_1} := \left|x_i^{\ell_1}\right|$, we show that $\left(x^{\ell_1}, t^{\ell_1}\right)$ is a solution of the LP. Indeed,

$$\sum_{i=1}^{n} t_i^{\ell_1} = \left|\left|x^{\ell_1}\right|\right|_1 \tag{170}$$

$$\leq \left|\left|x^{\mathrm{lp}}\right|\right|_1 \quad \text{by optimality of } x^{\ell_1} \tag{171}$$

$$\leq \sum_{i=1}^{n} t_i^{\mathrm{lp}} \quad \text{by constraints (60) and (61).} \tag{172}$$

Optimization methods

1 Introduction

In these notes we provide an overview of a selection of optimization methods. We focus on methods which rely on first-order information, i.e. gradients and subgradients, to make local progress towards a solution. In practice, these algorithms tend to converge to medium-precision solutions very rapidly and scale reasonably well with the problem dimension. As a result, they are widely used in machine-learning and signal-processing applications.

2 Differentiable functions

We begin by presenting gradient descent, a very simple algorithm that provably converges to the minimum of a convex differentiable function.

2.1 Gradient descent

Consider the optimization problem

$$\text{minimize} \quad f(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable and convex. Gradient descent exploits first-order local information encoded in the gradient to iteratively approach the point at which f achieves its minimum value. By multivariable calculus, at any point $x \in \mathbb{R}^n$ $-\nabla f(x)$ is the direction in which f decreases the most, see Figure 1. If we have no additional information about f, it makes sense to move in this direction at each iteration.

Algorithm 2.1 (Gradient descent, aka steepest descent)**.** *We set the initial point $x^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Then we apply*

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f\left(x^{(k)}\right), \tag{2}$$

$\alpha_k > 0$ is a nonnegative real number which we call the step size.

Gradient descent can be run for a certain number of iterations, which might depend on computational constraints, or until a stopping criterion is met. An example of a stopping rule is checking whether the relative progress $\left|\left|x^{(k+1)} - x^{(k)}\right|\right|_2 / \left|\left|x^{(k)}\right|\right|_2$ is below a certain

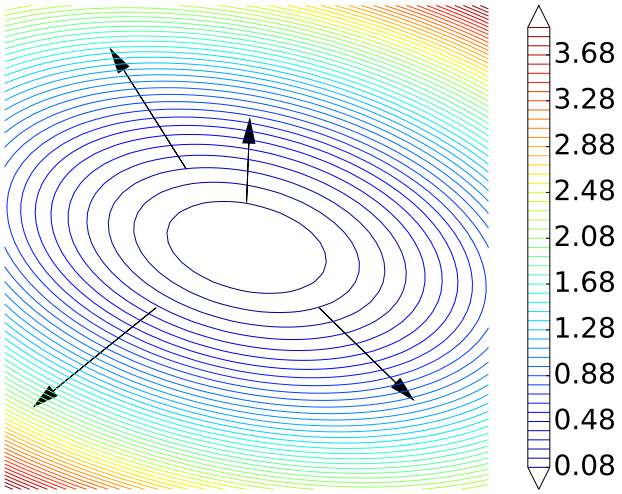


Figure 1: Contour lines of a function $f : \mathbb{R}^2 \to \mathbb{R}$. The gradients at different points are represented by black arrows, which are orthogonal to the contour lines.

value. Figure 2 shows two examples in which gradient descent is applied in one and two dimensions. In both cases the method converges to the minimum.

In the examples of Figure 2 the step size is constant. In practice, determining a constant step that is adequate for a particular function can be challenging. Figure 3 shows two examples to illustrate this. In the first the step size is too small and as a result convergence is extremely slow. In the second the step size is too large which causes the algorithm to repeatedly overshoot the minimum and eventually diverge.

Ideally, we would like to adapt the step size automatically as the iterations progress. A possibility is to search for the minimum of the function along the direction of the gradient,

$$\alpha_k := \arg\min_{\alpha} h(\alpha) \tag{3}$$

$$= \arg\min_{\alpha \in \mathbb{R}} f\left(x^{(k)} - \alpha_k \nabla f\left(x^{(k)}\right)\right). \tag{4}$$

This is called a line search. Recall that the restriction of an n-dimensional convex function to a line in its domain is also convex. As a result the line search problem is a one-dimensional convex problem. However, it may still be costly to solve. The backtracking line search is an alternative heuristic that produces very similar results in practice at less cost. The idea is to ensure that we make some progress in each iteration, without worrying about actually

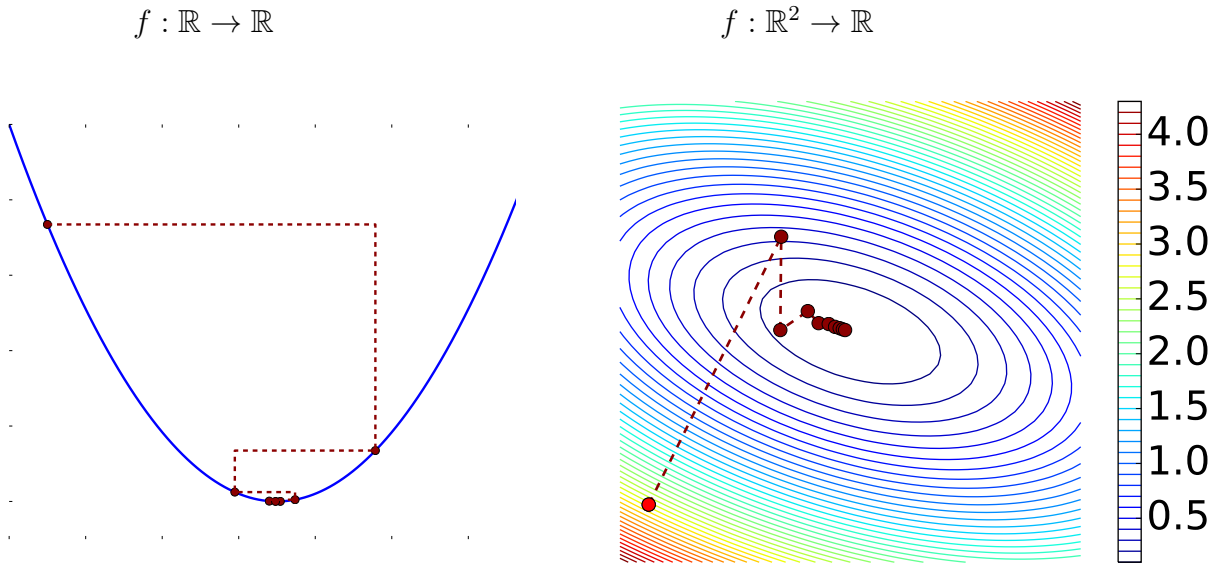


Figure 2: Iterations of gradient descent applied to a univariate (left) and a bivariate (right) function. The algorithm converges to the minimum in both cases.

minimizing the univariate function. By the first-order characterization of convexity we have

$$\begin{aligned} h(\alpha) &= f(x - \alpha \nabla f(x)) && (5)\\ &\geq f(x) - \nabla f(x)^T (x - (x - \alpha \nabla f(x))) && (6)\\ &= f(x) - \alpha ||\nabla f(x)||_2^2 && (7)\\ &= h(0) - \alpha ||\nabla f(x)||_2^2 . && (8) \end{aligned}$$

The backtracking line search starts at a large value of α and decreases it until the function is below $f(x) - \frac{1}{2} ||\nabla f(x)||_2^2$, a condition known as Armijo rule. Note that the Armijo rule will be satisfied eventually. The reason is that the line $h(0) - \alpha ||\nabla f(x)||_2^2$ is the only supporting line of h at zero because h is differentiable and convex (so the only subgradient at a point is the gradient). Consequently $h(\alpha)$ must be below the line $h(0) - \frac{\alpha}{2} ||f(x)||_2^2$ as $\alpha \to 0$, because otherwise this other line would also support h at zero.

Algorithm 2.2 (Backtracking line search with Armijo rule)**.** *Given $\alpha^0 \geq 0$ and $\beta, \eta \in (0,1)$, set $\alpha_k := \alpha^0 \beta^i$ for the smallest integer i such that*

$$f\left(x^{(k+1)}\right) \leq f\left(x^{(k)}\right) - \frac{1}{2} \alpha_k \left|\left|\nabla f\left(x^{(k)}\right)\right|\right|_2^2 . \quad (9)$$

Figure 4 shows the result of applying gradient descent with a backtracking line search to the same example as in Figure 3. In this case, the line search manages to adjust the step size so that the method converges.

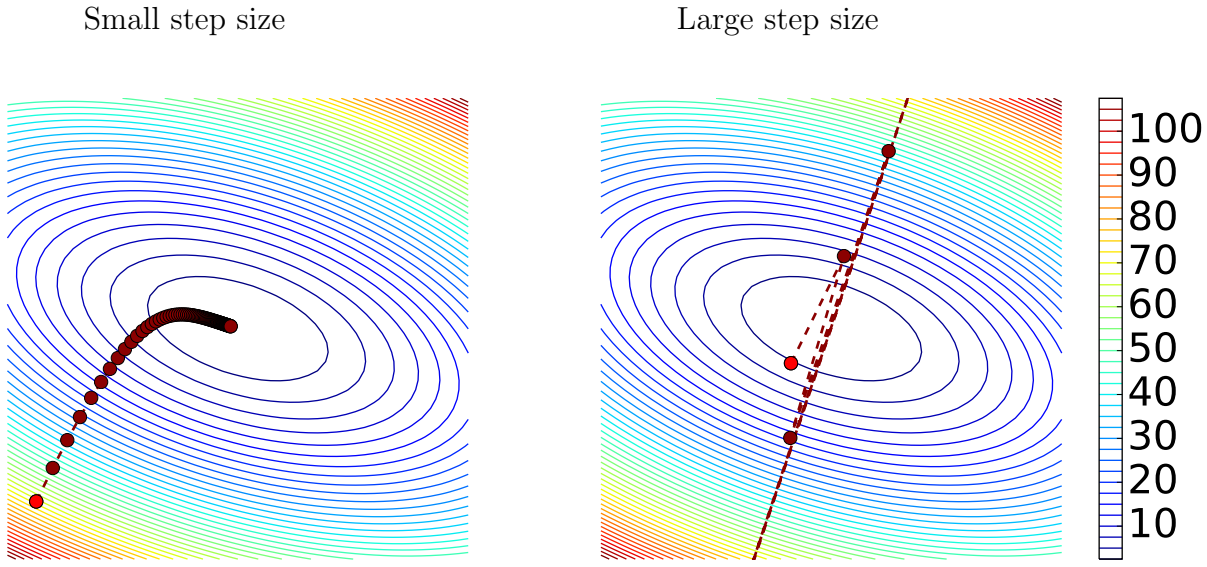


Figure 3: Iterations of gradient descent when the step size is small (left) and large (right). In the first case the convergence is very small, whereas in the second the algorithm diverges away from the minimum. The initial point is bright red.

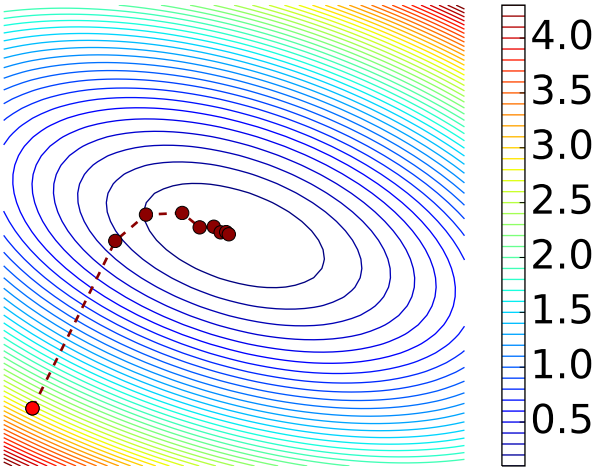


Figure 4: Gradient descent using a backtracking line search based on the Armijo rule. The function is the same as in Figure 3.

2.2 Convergence analysis

In this section we will analyze the convergence of gradient descent for a certain class of functions. We begin by introducing a notion of continuity for functions from $\mathbb{R}^n$ to $\mathbb{R}^m$.

Definition 2.3 (Lipschitz continuity)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz continuous with Lipschitz constant L if for any $x, y \in \mathbb{R}^n$*

$$||f(y) - f(x)||_2 \leq L ||y - x||_2 . \tag{10}$$

We will focus on functions that have Lipschitz-continuous gradients. The following proposition, proved in Section A.1 of the appendix, shows that we are essentially considering functions that are upper bounded by a quadratic function.

Proposition 2.4 (Quadratic upper bound)**.** *If the gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous with Lipschitz constant L,*

$$||\nabla f(y) - \nabla f(x)||_2 \leq L ||y - x||_2 \tag{11}$$

then for any $x, y \in \mathbb{R}^n$

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2} ||y - x||_2^2 . \tag{12}$$

The quadratic upper bound immediately implies a bound on the value of the cost function after k iterations of gradient descent that will be very useful.

Corollary 2.5. *Let $x^{(i)}$ be the ith iteration of gradient descent and $\alpha_i \geq 0$ the ith step size, if ∇f is L-Lipschitz continuous,*

$$f\left(x^{(k+1)}\right) \leq f\left(x^{(k)}\right) - \alpha_k \left(1 - \frac{\alpha_k L}{2}\right) \left|\left|\nabla f\left(x^{(k)}\right)\right|\right|_2^2 . \tag{13}$$

Proof. Applying the quadratic upper bound we obtain

$$f\left(x^{(k+1)}\right) \leq f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)^T \left(x^{(k+1)} - x^{(k)}\right) + \frac{L}{2} \left|\left|x^{(k+1)} - x^{(k)}\right|\right|_2^2 . \tag{14}$$

The result follows because $x^{(k+1)} - x^{(k)} = -\alpha_k \nabla f\left(x^{(k)}\right)$. □

We can now establish that if the step size is small enough, the value of the cost function at each iteration will decrease (unless we are at the minimum where the gradient is zero).

Corollary 2.6 (Gradient descent is indeed a descent method)**.** *If $\alpha_k \leq \frac{1}{L}$*

$$f\left(x^{(k+1)}\right) \leq f\left(x^{(k)}\right) - \frac{\alpha_k}{2} \left|\left|\nabla f\left(x^{(k)}\right)\right|\right|_2^2 . \tag{15}$$

Note that up to now we are *not* assuming that the function we are minimizing is convex. Gradient descent will make local progress even for nonconvex functions if the step size is sufficiently small. We now establish global convergence for gradient descent applied to convex functions with Lipschitz-continuous gradients.

Theorem 2.7. *We assume that f is convex, ∇f is L-Lipschitz continuous and there exists a point x^* at which f achieves a finite minimum. If we set the step size of gradient descent to $\alpha_k = \alpha \leq 1/L$ for every iteration,*

$$f\left(x^{(k)}\right) - f\left(x^*\right) \leq \frac{\left|\left|x^{(0)} - x^*\right|\right|_2^2}{2\,\alpha\,k} \tag{16}$$

Proof. By the first-order characterization of convexity

$$f\left(x^{(i-1)}\right) + \nabla f\left(x^{(i-1)}\right)^T \left(x^* - x^{(i-1)}\right) \leq f\left(x^*\right), \tag{17}$$

which together with Corollary 2.6 yields

$$\begin{aligned} f\left(x^{(i)}\right) - f\left(x^*\right) &\leq \nabla f\left(x^{(i-1)}\right)^T \left(x^{(i-1)} - x^*\right) - \frac{\alpha}{2}\left|\left|\nabla f\left(x^{(i-1)}\right)\right|\right|_2^2 && (18)\\ &= \frac{1}{2\,\alpha}\left(\left|\left|x^{(i-1)} - x^*\right|\right|_2^2 - \left|\left|x^{(i-1)} - x^* - \alpha\nabla f\left(x^{(i-1)}\right)\right|\right|_2^2\right) && (19)\\ &= \frac{1}{2\,\alpha}\left(\left|\left|x^{(i-1)} - x^*\right|\right|_2^2 - \left|\left|x^{(i)} - x^*\right|\right|_2\right) && (20) \end{aligned}$$

Using the fact that by Corollary 2.6 the value of f never increases, we have

$$\begin{aligned} f\left(x^{(k)}\right) - f\left(x^*\right) &\leq \frac{1}{k}\sum_{i=1}^{k} f\left(x^{(i)}\right) - f\left(x^*\right) && (21)\\ &= \frac{1}{2\,\alpha\,k}\left(\left|\left|x^{(0)} - x^*\right|\right|_2^2 - \left|\left|x^{(k)} - x^*\right|\right|_2^2\right) && (22)\\ &\leq \frac{\left|\left|x^{(0)} - x^*\right|\right|_2^2}{2\,\alpha\,k}. && (23) \end{aligned}$$

□

The theorem assumes that we know the Lipschitz constant of the gradient beforehand. However, the following lemma establishes that a backtracking line search with the Armijo rule is capable of adjusting the step size adequately.

Lemma 2.8 (Backtracking line search)**.** *If the gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous with Lipschitz constant L the step size obtained by applying a backtracking line search using the Armijo rule with $\eta = 0.5$ satisfies*

$$\alpha_k \geq \alpha_{\min} := \min\left\{\alpha^0, \frac{\beta}{L}\right\}. \tag{24}$$

Proof. By Corollary 2.5 the Armijo rule with $\eta = 0.5$ is satisfied if $\alpha_k \leq 1/L$. Since there must exist an integer i for which $\beta/L \leq \alpha^0 \beta^i \leq 1/L$ this establishes the result. □

We can now adapt the proof of Theorem 2.7 to establish convergence when we apply a backtracking line search.

Theorem 2.9 (Convergence with backtracking line search)**.** *If f is convex and ∇f is L-Lipschitz continuous. Gradient descent with a backtracking line search produces a sequence of points that satisfy*

$$f\left(x^{(k)}\right) - f\left(x^*\right) \leq \frac{\left|\left|x^{(0)} - x^*\right|\right|_2^2}{2\,\alpha_{\min}\,k}, \tag{25}$$

where $\alpha_{\min} := \min\left\{\alpha^0, \frac{\beta}{L}\right\}$.

Proof. Following the reasoning in the proof of Theorem 2.7 up until equation (20) we have

$$f\left(x^{(i)}\right) - f\left(x^*\right) \leq \frac{1}{2\,\alpha_i}\left(\left|\left|x^{(i-1)} - x^*\right|\right|_2^2 - \left|\left|x^{(i)} - x^*\right|\right|_2\right). \tag{26}$$

By Lemma 2.8 $\alpha_i \geq \alpha_{\min}$, so we just mimic the steps at the end of the proof of Theorem 2.7 to obtain

$$\begin{aligned} f\left(x^{(k)}\right) - f\left(x^*\right) &\leq \frac{1}{k}\sum_{i=1}^{k} f\left(x^{(i)}\right) - f\left(x^*\right) && (27)\\ &= \frac{1}{2\,\alpha_{\min}\,k}\left(\left|\left|x^{(0)} - x^*\right|\right|_2^2 - \left|\left|x^{(k)} - x^*\right|\right|_2^2\right) && (28)\\ &\leq \frac{\left|\left|x^{(0)} - x^*\right|\right|_2^2}{2\,\alpha_{\min}\,k}. && (29) \end{aligned}$$

□

The results that we have proved imply that we need $\mathcal{O}\left(1/\epsilon\right)$ to compute a point at which the cost function has a value that is ϵ close to the minimum. However, in practice gradient descent and related methods often converge much faster. If we restrict our class of functions of interest further this can often be made theoretically rigorous. To illustrate this we introduce strong convexity.

Definition 2.10 (Strong convexity)**.** *A function $f : \mathbb{R}^n$ is S-strongly convex if for any $x, y \in \mathbb{R}^n$*

$$f\left(y\right) \geq f\left(x\right) + \nabla f\left(x\right)^T\left(y - x\right) + S\left|\left|y - x\right|\right|^2. \tag{30}$$

Strong convexity means that the function is lower bounded by a quadratic with a fixed curvature at any point. Under this condition, gradient descent converges much faster. The following result establishes that we only need $\mathcal{O}\left(\log \frac{1}{\epsilon}\right)$ iterations to get an ϵ-optimal solution. For the proof see [6].

Theorem 2.11. *If f is S-strongly convex and ∇f is L-Lipschitz continuous*

$$f\left(x^{(k)}\right) - f\left(x^*\right) \leq \frac{c^k L \left|\left|x^{(k)} - x_0\right|\right|_2^2}{2}, \qquad c := \frac{\frac{L}{S} - 1}{\frac{L}{S} + 1}. \tag{31}$$

2.3 Accelerated gradient descent

The following theorem by Nesterov shows that no algorithm that uses first-order information can converge faster than $\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\right)$ for the class of functions with Lipschitz-continuous gradients. The proof is constructive, see Section 2.1.2 of [4] for the details.

Theorem 2.12 (Lower bound on rate of convergence)**.** *There exist convex functions with L-Lipschitz-continuous gradients such that for any algorithm that selects $x^{(k)}$ from*

$$x^{(0)} + \text{span}\left\{\nabla f\left(x^{(0)}\right), \nabla f\left(x^{(1)}\right), \ldots, \nabla f\left(x^{(k-1)}\right)\right\} \tag{32}$$

we have

$$f\left(x^{(k)}\right) - f\left(x^*\right) \geq \frac{3L \left|\left|x^{(0)} - x^*\right|\right|_2^2}{32\ (k+1)^2} \tag{33}$$

This rate is in fact optimal. The convergence of $\mathcal{O}\left(1/\sqrt{\epsilon}\right)$ can be achieved if we modify gradient descent by adding a momentum term:

$$y^{(k+1)} = x^{(k)} - \alpha_k \nabla f\left(x^{(k)}\right), \tag{34}$$

$$x^{(k+1)} = \beta_k\, y^{(k+1)} + \gamma_k\, y^{(k)}, \tag{35}$$

where β_k and γ_k may depend on k. This version of gradient descent is usually known as accelerated gradient descent or Nesterov's method. In Section 3.2 we will illustrate the application of this idea to the proximal gradient method. Intuitively, a momentum term prevents descent methods from overreacting to changes in the local slope of the function. We refer the interested reader to [2, 4] for more details.

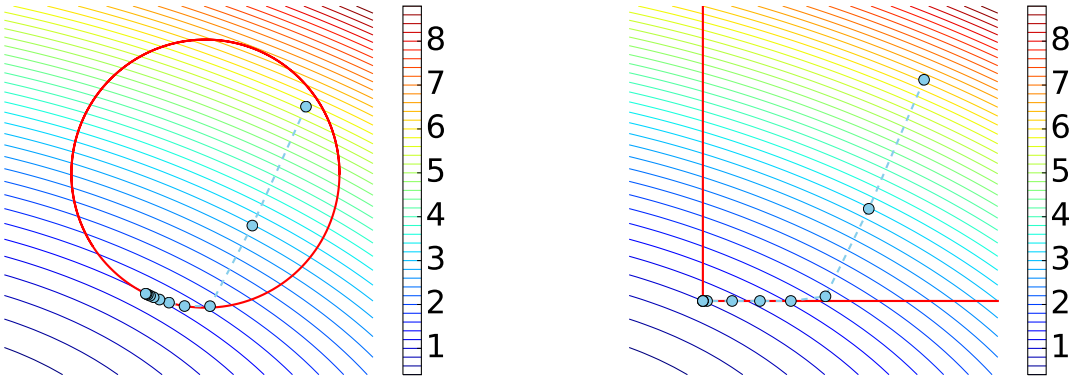


Figure 5: Iterations of projected gradient descent applied to a convex function with feasibility sets equal to the unit ℓ_2 norm (left) and the positive quadrant(right).

2.4 Projected gradient descent

In this section we explain how to adapt gradient descent to minimize a function within a convex feasibility set, i.e. to solve

$$\begin{aligned} &\text{minimize} \quad f(x) && (36)\\ &\text{subject to} \quad x \in \mathcal{S}, && (37) \end{aligned}$$

where f is differentiable and $\mathcal{S}$ is convex. The method is very simple. At each iteration we take a gradient-descent step and project on the feasibility set.

Algorithm 2.13 (Projected gradient descent)**.** *We set the initial point $x^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Then we compute*

$$x^{(k+1)} = \mathcal{P}_{\mathcal{S}} \left(x^{(k)} - \alpha_k \nabla f \left(x^{(k)} \right) \right), \tag{38}$$

until a convergence criterion is satisfied.

Figure 5 shows the results of applying projected gradient descent to minimize a convex function with two different feasibility sets in $\mathbb{R}^2$: the unit ℓ_2 norm and the positive quadrant.

3 Nondifferentiable functions

In this section we describe several methods to minimize convex nondifferentiable functions.

3.1 Subgradient method

Consider the optimization problem

$$\text{minimize} \quad f(x) \tag{39}$$

where f is convex but nondifferentiable. This implies that we cannot compute a gradient and advance in the steepest descent direction as in gradient descent. However, we can generalize the idea by using subgradients, which exist because f is convex. This will be useful as long as it is efficient to compute the subgradient of the function.

Algorithm 3.1 (Subgradient method)**.** *We set the initial point $x^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Then we compute*

$$x^{(k+1)} = x^{(k)} - \alpha_k \, q^{(k)}, \tag{40}$$

where $q^{(k)}$ is a subgradient of f at $x^{(k)}$, until a convergence criterion is satisfied.

Interestingly, the subgradient method is not a descent method. The value of the cost function can actually increase as the iterations progress. However, the method can be shown to converge at a rate of order $\mathcal{O}(1/\epsilon^2)$ as long as the step size decreases along iterations, see [6].

We now illustrate the performance of the subgradient method applied to least-squares regression with ℓ_1-norm regularization. The cost function in the optimization problem,

$$\text{minimize} \quad \frac{1}{2} \left|\left|Ax - y\right|\right|_2^2 + \lambda \left|\left|x\right|\right|_1, \tag{41}$$

is convex but not differentiable. To compute a subgradient of the function at $x^{(k)}$ we use the fact that $\text{sign}(x)$ is a subgradient of the ℓ_1 norm at x, as we established in the previous lecture,

$$q^{(k)} = A^T \left(Ax^{(k)} - y\right) + \lambda \, \text{sign}\left(x^{(k)}\right). \tag{42}$$

The subgradient-method iteration for this problem is consequently of the form

$$x^{(k+1)} = x^{(k)} - \alpha_k \left(A^T \left(Ax^{(k)} - y\right) + \lambda \, \text{sign}\left(x^{(k)}\right)\right). \tag{43}$$

Figure 6 shows the result of applying this algorithm to an example in which $A \in \mathbb{R}^{2000 \times 1000}$, $y = Ax_0 + z$ where x_0 is 100-sparse and z is iid Gaussian. The example illustrates that decreasing the step size at each iteration achieves faster convergence.

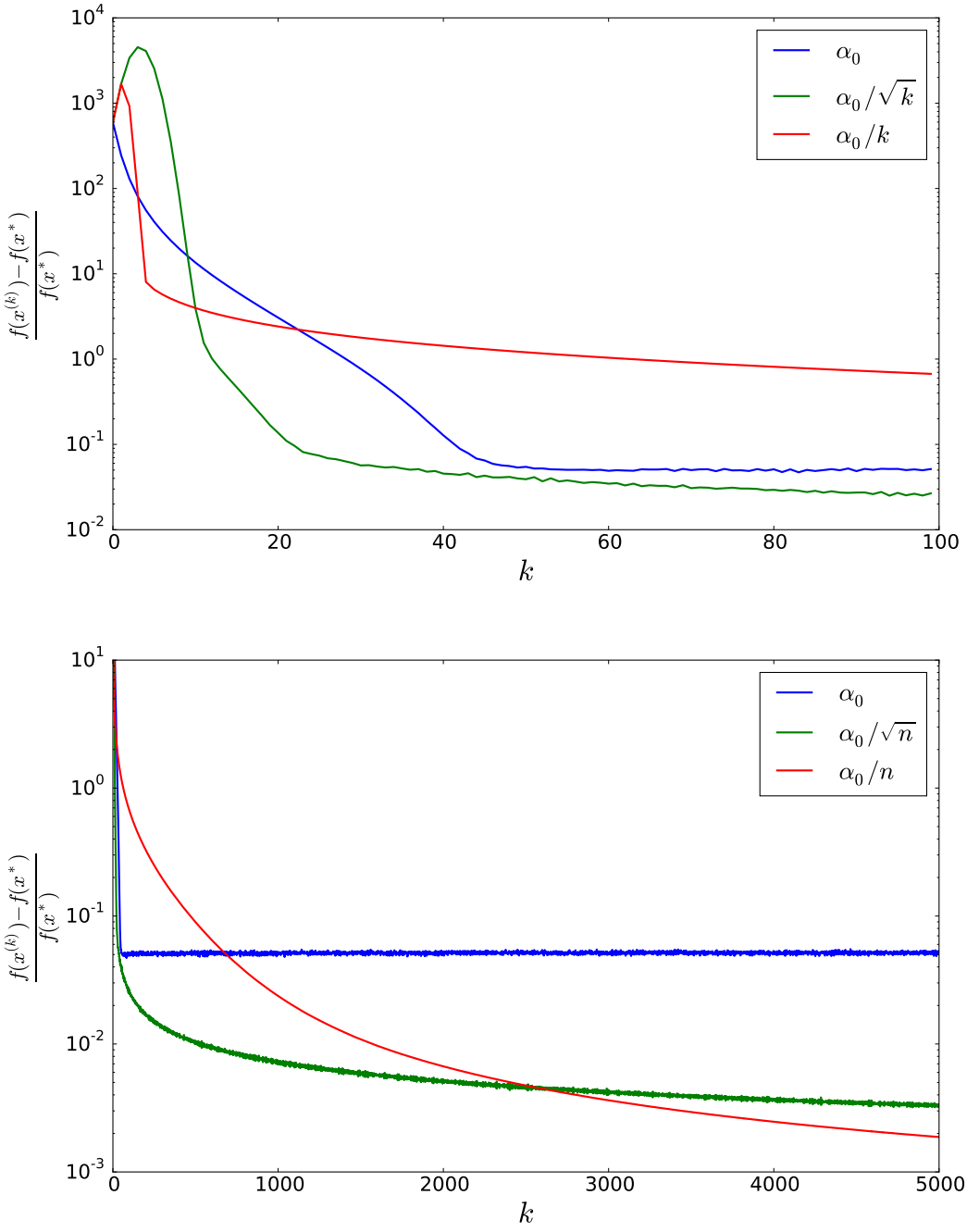


Figure 6: Subgradient method applied to least-squares regression with ℓ_1-norm regularization for different choices of step size (α_0 is a constant).

3.2 Proximal gradient method

As we saw in the previous section, convergence of subgradient method is slow, both in terms of theoretical guarantees and in the example of Figure 6. In this section we introduce an alternative method that can be applied to a class of functions which is very useful for optimization-based data analysis.

Definition 3.2 (Composite function)**.** *A composite function is a function that can be written as the sum*

$$f(x) + g(x) \tag{44}$$

where f convex and differentiable and g is convex but not differentiable.

Clearly, the least-squares regression cost function with ℓ_1-norm regularization is of this form.

In order to motivate proximal methods, let us begin by interpreting the gradient-descent iteration as the solution to a *local* linearization of the function.

Lemma 3.3. *The minimum of the function*

$$h(x) := f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)^T \left(x - x^{(k)}\right) + \frac{1}{2\,\alpha} \left|\left|x - x^{(k)}\right|\right|_2^2 \tag{45}$$

is $x^{(k)} - \alpha \nabla f\left(x^{(k)}\right)$.

Proof.

$$\begin{aligned} x^{(k+1)} &:= x^{(k)} - \alpha_k \nabla f\left(x^{(k)}\right) && (46)\\ &= \arg\min_x \left|\left|x - \left(x^{(k)} - \alpha_k \nabla f\left(x^{(k)}\right)\right)\right|\right|_2^2 && (47)\\ &= \arg\min_x f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)^T \left(x - x^{(k)}\right) + \frac{1}{2\,\alpha_k} \left|\left|x - x^{(k)}\right|\right|_2^2. && (48) \end{aligned}$$

□

A natural generalization of gradient descent is to minimize the sum of g and the local first-order approximation of f.

$$\begin{aligned} x^{(k+1)} &= \arg\min_x f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)^T \left(x - x^{(k)}\right) + \frac{1}{2\,\alpha_k} \left|\left|x - x^{(k)}\right|\right|_2^2 + g(x) && (49)\\ &= \arg\min_x \frac{1}{2} \left|\left|x - \left(x^{(k)} - \alpha_k \nabla f\left(x^{(k)}\right)\right)\right|\right|_2^2 + \alpha_k\, g(x) && (50)\\ &= \mathrm{prox}_{\alpha_k\, g}\left(x^{(k)} - \alpha_k \nabla f\left(x^{(k)}\right)\right). && (51) \end{aligned}$$

We have written the iteration in terms of the proximal operator of the function g.

Definition 3.4 (Proximal operator)**.** *The proximal operator of a function* $g : \mathbb{R}^n \to \mathbb{R}$ *is*

$$\text{prox}_g(y) := \arg\min_x g(x) + \frac{1}{2}||x - y||_2^2. \tag{52}$$

Solving the modified local first-order approximation of the composite function iteratively yields the proximal-gradient method, which will be useful if the proximal operator of g can be computed efficiently.

Algorithm 3.5 (Proximal-gradient method)**.** *We set the initial point* $x^{(0)}$ *to an arbitrary value in* $\mathbb{R}^n$*. Then we compute*

$$x^{(k+1)} = \text{prox}_{\alpha_k g}\left(x^{(k)} - \alpha_k \nabla f\left(x^{(k)}\right)\right), \tag{53}$$

until a convergence criterion is satisfied.

This algorithm may be interpreted as a fixed-point method. Indeed, vector is a fixed point of the proximal-gradient iteration if and only if it is a minimum of the composite function. This suggests that it is a good idea to apply the iteration repeatedly but that does not prove convergence (for this we would need to prove that the operator is contractive, see [6]).

Proposition 3.6 (Fixed point of proximal operator)**.** *A vector* $\hat{x}$ *is a solution to*

$$\text{minimize} \quad f(x) + g(x), \tag{54}$$

if and only if it is a fixed point of the proximal-gradient iteration

$$\hat{x} = \text{prox}_{\alpha g}(\hat{x} - \alpha \nabla f(\hat{x})) \tag{55}$$

for any $\alpha > 0$*.*

Proof. $\hat{x}$ is a solution to the optimization problem if and only if there exists a subgradient q of g at $\hat{x}$ such that $\nabla f(\hat{x}) + q = 0$. $\hat{x}$ is the solution to

$$\text{minimize} \quad \alpha g(x) + \frac{1}{2}||\hat{x} - \alpha \nabla f(\hat{x}) - x||_2^2, \tag{56}$$

which is the case if and only if there exists a subgradient q of g at $\hat{x}$ such that $\alpha \nabla f(\hat{x}) + \alpha q = 0$. As long as $\alpha > 0$ the two conditions are equivalent. □

In the case of the indicator function of a set,

$$\mathcal{I}_{\mathcal{S}}(x) := \begin{cases} 0 & \text{if } x \in \mathcal{S}, \\ \infty & \text{if } x \notin \mathcal{S}, \end{cases} \tag{57}$$

the proximal operator is just the projection onto the set.

Lemma 3.7 (Proximal operator of indicator function)**.** *The proximal operator of the indicator function of a convex set $\mathcal{S} \subseteq \mathbb{R}^n$ is projection onto $\mathcal{S}$.*

Proof. The lemma follows directly from the definitions of proximal operator, projection and indicator function. □

An immediate consequence of this is that projected-gradient descent can be interpreted as a special case of the proximal-gradient method.

Proximal methods are very useful for fitting sparse models because the proximal operator of the ℓ_1 norm is very tractable.

Proposition 3.8 (Proximal operator of ℓ_1 norm)**.** *The proximal operator of the ℓ_1 norm weighted by a constant $\lambda > 0$ is the soft-thresholding operator*

$$\text{prox}_{\lambda \, ||\cdot||_1}(y) = \mathcal{S}_\lambda(y) \tag{58}$$

where

$$\mathcal{S}_\lambda(y)_i := \begin{cases} y_i - \text{sign}(y_i)\,\lambda & \text{if } |y_i| \geq \lambda, \\ 0 & \text{otherwise.} \end{cases} \tag{59}$$

The proposition, proved in Section A.3 of the appendix, allows us to derive the following algorithm for least-squares regression with ℓ_1-norm regularization.

Algorithm 3.9 (Iterative Shrinkage-Thresholding Algorithm (ISTA))**.** *We set the initial point $x^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Then we compute*

$$x^{(k+1)} = \mathcal{S}_{\alpha_k \lambda}\left(x^{(k)} - \alpha_k A^T\left(Ax^{(k)} - y\right)\right), \tag{60}$$

until a convergence criterion is satisfied.

ISTA can be accelerated using a momentum term as in Nesterov's accelerated gradient method. This yields a fast version of the algorithm called FISTA.

Algorithm 3.10 (Fast Iterative Shrinkage-Thresholding Algorithm (FISTA))**.** *We set the initial point $x^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Then we compute*

$$z^{(0)} = x^{(0)} \tag{61}$$

$$x^{(k+1)} = \mathcal{S}_{\alpha_k \lambda}\left(z^{(k)} - \alpha_k A^T\left(Az^{(k)} - y\right)\right), \tag{62}$$

$$z^{(k+1)} = x^{(k+1)} + \frac{k}{k+3}\left(x^{(k+1)} - x^{(k)}\right), \tag{63}$$

until a convergence criterion is satisfied.

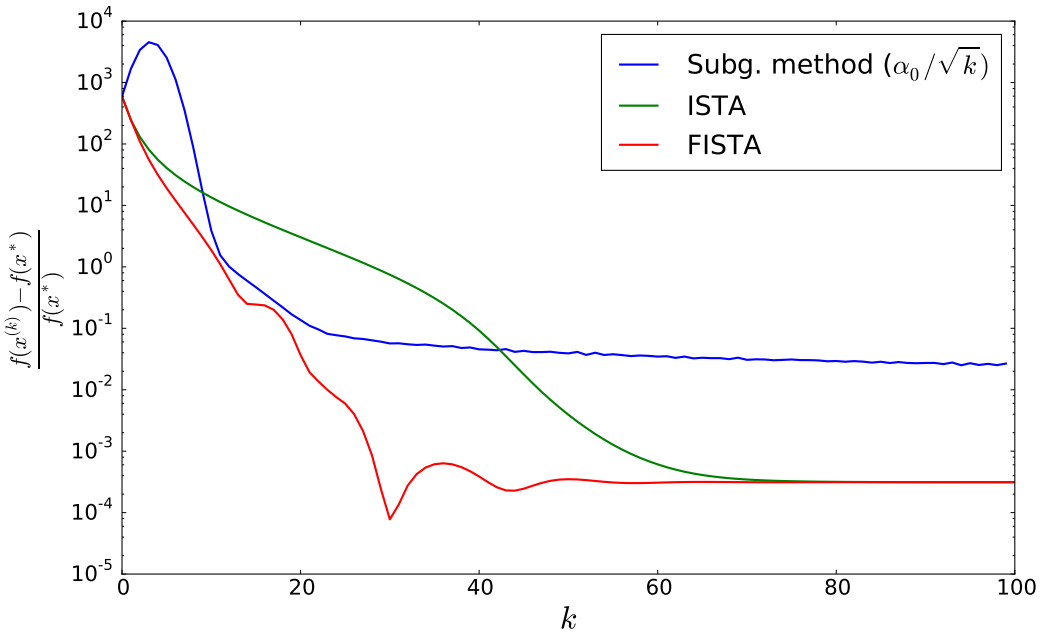


Figure 7: ISTA and FISTA applied to least-squares regression with ℓ_1-norm regularization.

ISTA and FISTA were proposed by Beck and Teboulle in [1]. ISTA is a descent method. It has the same convergence rate as gradient descent $\mathcal{O}\left(1/\epsilon\right)$ both with a constant step size and with a backtracking line search, under the condition that ∇f be L-Lipschitz continuous. FISTA in contrast is not a descent method, but it can be shown to converge in $\mathcal{O}\left(1/\sqrt{\epsilon}\right)$ to an ϵ-optimal solution.

To illustrate the performance of ISTA and FISTA, we apply them to the same example used in Figure 6. Even without applying a backtracking line search both methods converge to a solution of middle precision (around 10^{-3} or 10^{-4}) much more rapidly than the subgradient method. The results are shown in Figure 7.

3.3 Coordinate descent

Coordinate descent is an optimization method that decouples n-dimensional problems of the form

$$\text{minimize} \quad h\left(x\right) \tag{64}$$

by solving a sequence of 1D problems. The algorithm is very simple; we just fix all the entries of the variable vector except one and optimize over it. This procedure is called coordinate

descent because it is equivalent to iteratively minimizing the function in the direction of the axes.

Algorithm 3.11 (Coordinate descent)**.** *We set the initial point $x^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. At each iteration we choose an arbitrary position $1 \leq i \leq n$ and set*

$$x_i^{(k+1)} = \arg\min_{x_i \in \mathbb{R}} h\left(x_1^{(k)}, \ldots, x_i, \ldots, x_n^{(k)}\right), \tag{65}$$

until a convergence criterion is satisfied. The order of the entries that we optimize over can be fixed or random.

The method is guaranteed to converge for composite functions where f is differentiable and convex and g has an additive decomposition,

$$h(x) := f(x) + g(x) = f(x) + \sum_{i=1}^{n} g_i(x_i),$$

where the functions $g_1, g_2, \ldots, g_n : \mathbb{R} \to \mathbb{R}$ are convex and nondifferentiable [5].

In order to apply coordinate descent it is necessary for the one-dimensional optimization problems to be easy to solve. This is the case for least-squares regression with ℓ_1-norm regularization as demonstrated by the following proposition, which is proved in Section A.4.

Proposition 3.12 (Coordinate-descent subproblems for least-squares regression with ℓ_1-norm regularization)**.** *Let*

$$h(x) = \frac{1}{2} ||Ax - y||_2^2 + \lambda ||x||_1. \tag{66}$$

The solution to the subproblem $\min_{x_i} h(x_1, \ldots, x_i, \ldots, x_n)$ is

$$\hat{x}_i = \frac{\mathcal{S}_\lambda(\gamma_i)}{||A_i||_2^2} \tag{67}$$

where A_i is the ith column of A and

$$\gamma_i := \sum_{l=1}^{m} A_{li} \left(y_l - \sum_{j \neq i} A_{lj} x_j\right). \tag{68}$$

For more information on coordinate descent and its application to sparse regression, see Chapter 5 of [3].

References

For further reader on the convergence of first-order methods we recommend Nesterov's book [4] and Bubeck's notes [2]. Chapter 5 of [3] in Hastie, Tibshirani and Wainwright is a great description of proximal-gradient and coordinate-descent methods and their application to sparse regression.

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[2] S. Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

[3] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC Press, 2015.

[4] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*.

[5] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

[6] L. Vandenberghe. Notes on optimization methods for large-scale systems.

A Proofs

A.1 Proof of Proposition 2.4

Consider the function

$$g(x) := \frac{L}{2} x^T x - f(x). \tag{69}$$

We first establish that g is convex using the following lemma, proved in Section A.2 below.

Lemma A.1 (Monotonicity of gradient)**.** *A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if*

$$(\nabla f(y) - \nabla f(x))^T (y - x) \geq 0. \tag{70}$$

By the Cauchy-Schwarz inequality, Lipschitz continuity of the gradient of f implies

$$(\nabla f(y) - \nabla f(x))^T (y - x) \leq L ||y - x||_2^2, \tag{71}$$

for any $x, y \in \mathbb{R}^n$. This directly implies

$$\begin{aligned}
\left(\nabla g\left(y\right) - \nabla g\left(x\right)\right)^T \left(y - x\right) &= \left(Ly - Lx + \nabla f\left(x\right) - \nabla f\left(y\right)\right)^T \left(y - x\right) && (72)\\
&= L\left|\left|y - x\right|\right|_2^2 - \left(\nabla f\left(y\right) - \nabla f\left(x\right)\right)^T \left(y - x\right) && (73)\\
&\geq 0 && (74)
\end{aligned}$$

and hence that g is convex. By the first-order condition for convexity,

$$\begin{aligned}
\frac{L}{2} y^T y - f\left(y\right) &= g\left(y\right) && (75)\\
&\geq g\left(x\right) + \nabla g\left(x\right)^T \left(y - x\right) && (76)\\
&= \frac{L}{2} x^T x - f\left(x\right) + \left(Lx - \nabla f\left(x\right)\right)^T \left(y - x\right). && (77)
\end{aligned}$$

Rearranging the inequality we conclude that

$$f\left(y\right) \leq f\left(x\right) + \nabla f\left(x\right)^T \left(y - x\right) + \frac{L}{2}\left|\left|y - x\right|\right|_2^2. \quad (78)$$

A.2 Proof of Lemma A.1

Convexity implies $\left(\nabla f\left(y\right) - \nabla f\left(x\right)\right)^T \left(y - x\right) \geq 0$ for all $x, y \in R^n$

If f is convex, by the first-order condition for convexity

$$\begin{aligned}
f\left(y\right) &\geq f\left(x\right) + \nabla f\left(x\right)^T \left(y - x\right), && (79)\\
f\left(x\right) &\geq f\left(y\right) + \nabla f\left(y\right)^T \left(x - y\right), && (80)\\
& && (81)
\end{aligned}$$

Adding the two inequalities directly implies the result.

$\left(\nabla f\left(y\right) - \nabla f\left(x\right)\right)^T \left(y - x\right) \geq 0$ for all $x, y \in R^n$ implies convexity

Recall the univariate function $g_{a,b} : [0, 1] \to \mathbb{R}$ defined by

$$g_{a,b}\left(\alpha\right) := f\left(\alpha a + \left(1 - \alpha\right) b\right), \quad (82)$$

for any $a, b \in \mathbb{R}^n$. By multivariate calculus, $g'_{a,b}\left(\alpha\right) = \nabla f\left(\alpha\, a + \left(1 - \alpha\right) b\right)^T \left(a - b\right)$. For any $\alpha \in (0, 1)$ we have

$$\begin{aligned}
g'_{a,b}\left(\alpha\right) - g'_{a,b}\left(0\right) &= \left(\nabla f\left(\alpha\, a + \left(1 - \alpha\right) b\right) - \nabla f\left(b\right)\right)^T \left(a - b\right) && (83)\\
&= \frac{1}{\alpha}\left(\nabla f\left(\alpha\, a + \left(1 - \alpha\right) b\right) - \nabla f\left(b\right)\right)^T \left(\alpha\, a + \left(1 - \alpha\right) b - b\right) && (84)\\
&\geq 0 \quad \text{because } \left(\nabla f\left(y\right) - \nabla f\left(x\right)\right)^T \left(y - x\right) \geq 0 \text{ for any } x, y. && (85)
\end{aligned}$$

This allows us to prove that the first-order condition for convexity holds. For any x, y

$$f(x) = g_{x,y}(1) \tag{86}$$

$$= g_{x,y}(0) + \int_0^1 g'_{x,y}(\alpha)\ \mathrm{d}\,\alpha \tag{87}$$

$$\geq g_{x,y}(0) + g'_{x,y}(0) \tag{88}$$

$$= f(y) + \nabla f(y)(x - y). \tag{89}$$

A.3 Proof of Proposition 3.8

Writing the function as a sum,

$$\lambda ||x||_1 + \frac{1}{2}||y - x||_2^2 = \sum_{i=1}^{n} |x_i| + \frac{1}{2}(y_i - x_i)^2 \tag{90}$$

reveals that it decomposes into independent nonnegative terms. The univariate function

$$h(\beta) := \lambda |\beta| + \frac{1}{2}(y_i - \beta)^2 \tag{91}$$

is strictly convex and consequently has a unique global minimum. It is also differentiable everywhere except at zero. If $\beta \geq 0$ the derivative is $\lambda + \beta - y_i$, so if $y_i \geq \lambda$, the minimum is achieved at $y_i - \lambda$. If $y_i < \lambda$ the function is increasing for $\beta \geq 0$, so the minimizer must be smaller or equal to zero. The derivative for $\beta < 0$ is $-\lambda + \beta - y_i$ so the minimum is achieved at $y_i + \lambda$ if $y_i \leq \lambda$. Otherwise the function is decreasing for all $\beta < 0$. As a result, if $-\lambda < y_i < \lambda$ the minimum must be at zero.

A.4 Proof of Proposition 3.12

Note that

$$\min_{x_i} h(x) = \min_{x_i} \sum_{l=1}^{n} \frac{1}{2}\left(\sum_{j=1}^{m} A_{lj}x_j - y_l\right)^2 + \lambda \sum_{j=l}^{n} |x_l| \tag{92}$$

$$= \min_{x_i} \sum_{l=1}^{n} \frac{1}{2}A_{li}^2 x_i^2 + \left(\sum_{j\neq i}^{m} A_{lj}x_j - y_l\right) A_{li}x_i + \lambda\ |x_l| \tag{93}$$

$$= \min_{x_i} \frac{1}{2}||A_i||_2^2\, x_i^2 - \gamma_i x_i + \lambda\ |x_i|. \tag{94}$$

The univariate function

$$g(\beta) := \frac{1}{2}||A_i||_2^2\, \beta^2 - \gamma_i \beta + \lambda\ |\beta| \tag{95}$$

is strictly convex and consequently has a unique global minimum. It is also differentiable everywhere except at zero. If $\beta \geq 0$ the derivative is $||A_i||_2^2 \beta - \gamma_i + \lambda$, so if $\gamma_i \geq \lambda$, the minimum is achieved at $(\gamma_i - \lambda) / ||A_i||_2^2$. If $\gamma_i < \lambda$ the function is increasing for $\beta \geq 0$, so the minimizer must be smaller or equal to zero. The derivative for $\beta < 0$ is $||A_i||_2^2 \beta - \gamma_i - \lambda$ so the minimum is achieved at $(\gamma_i + \lambda) / ||A_i||_2^2$ if $\gamma_i \leq \lambda$. Otherwise the function is decreasing for all $\beta < 0$. As a result, if $-\lambda < \gamma_i < \lambda$ the minimum must be at zero.

Sparse linear models and denoising

1 Introduction

1.1 Definition and motivation

Finding representations of signals that allow to process them more effectively is a central problem in signal processing and data analysis. We will consider linear models, where the signal x is a represented as a sum of weighted atoms $\phi_1, \phi_2, \ldots, \phi_m$

$$x = \sum_{i=1}^{m} c_i \, \phi_i. \tag{1}$$

The set of atoms $\{\phi_1, \phi_2, \ldots, \phi_m\}$ is often called a dictionary. The m-dimensional vector of coefficients c is the representation of the signal x in the dictionary.

A sparse linear model consists of the sum of a small number of atoms selected from a certain dictionary

$$x = \sum_{i \in \mathcal{I}} c_i \, \phi_i \qquad |\mathcal{I}| \ll m, \tag{2}$$

here $\mathcal{I}$ is a set of indices of atoms in the dictionary. Its cardinality is significantly smaller than the number of atoms in the dictionary.

Signals in specific applications tend to have similar characteristics. In this lecture we will study how to use sparse linear models to exploit this common structure and enhance data analysis in applications such as compression and denoising. These models may also be applied to tackle inverse problems, as we will see in the next couple of lectures.

Finding dictionaries that are able to represent classes of signals, such as images or speech, parsimoniously has been an extremely active area of research in the last 30 years. It is very related to the problem of computing useful features in machine learning. There are two main approaches to building sparsifying dictionaries. The first is to use domain knowledge and intuition. This is the approach on which we will focus in this lecture. The second approach is to learn the transformation directly from a database of signals in the class of interest. We will study such methods later on in the course.

1.2 Bases and overcomplete dictionaries

If the atoms of the dictionary $\{\phi_1, \ldots, \phi_n\}$ form an orthonormal basis of $\mathbb{R}^n$ (or C^n) fitting the coefficients of a linear model is extremely simple. If (1) and the atoms are orthonormal then

$$c_i = \langle \phi_i, x \rangle \,. \tag{3}$$

The coefficients are obtained by computing inner products with the atoms. The representation is simply

$$x = \sum_{i=1}^{m} \langle \phi_i, x \rangle \, \phi_i. \tag{4}$$

If we construct a matrix using the atoms as columns

$$U := \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_n \end{bmatrix} \tag{5}$$

then $c = U^T x$ (or $c = U^*$ in the complex case).

If the atoms $\{\phi_1, \ldots, \phi_n\}$ form a basis, then the matrix

$$B := \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_n \end{bmatrix}, \tag{6}$$

has an inverse matrix B^{-1}. The rows of B^{-1}, which we denote by $\theta_1, \theta_2, \ldots, \theta_n$, can be interpreted as dual atoms. We have

$$x = BB^{-1}x = \sum_{i=1}^{m} \langle \theta_i, x \rangle \, \phi_i \tag{7}$$

so

$$c_i = \langle \theta_i, x \rangle \,. \tag{8}$$

If the atoms $\{\phi_1, \ldots, \phi_m\}$ are linearly independent and $m > n$, then the dictionary

$$D := \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_m \end{bmatrix} \tag{9}$$

is overcomplete and no longer has an inverse. There are two alternative ways in which we can use a given overcomplete dictionary to define a sparse linear model:

1. **Synthesis:** The synthesis sparse model assumes that there is a sparse m-dimensional vector of coefficients c such that

   $$x = Dc. \tag{10}$$

   As we will see, finding such a c from a given x is not necessarily an easy task.

2. **Analysis:** In the analysis sparse model assumes that

$$D^T x \tag{11}$$

is sparse, i.e. that the signal has nonzero correlation with a small number of atoms in the dictionary.

If the dictionary is an orthonormal basis, both models are equivalent.

2 Linear transforms

In this section we describe some of the most important transforms in signal processing.

2.1 Frequency representation

The frequency decomposition or spectrum of a function is obtained by representing the function as a superposition of sinusoids. To make this more formal, let us consider an infinite dictionary of sinusoids

$$\phi_k := \left\{ e^{2\pi k t} = \cos(2\pi k t) + i \sin(2\pi k t),\ k \in \mathbb{Z} \right\}. \tag{12}$$

Recall that $\mathbb{L}_2([0,1])$, the space of square-integrable functions defined on the unit interval, is a Hilbert space when endowed with the inner product

$$\langle f, g \rangle = \int_0^1 \overline{f(t)} g(t)\, \mathrm{d}\, t. \tag{13}$$

It is not difficult to check that the dictionary of sinusoidal is orthonormal under this inner product: the atoms are mutually orthogonal and have unit norm.

The Fourier series coefficients of a signal $f \in \mathbb{L}_2$ are obtained by taking its inner product with atoms from the dictionary,

$$c_k := \langle \phi_k, f \rangle \tag{14}$$

$$= \int_0^1 f(t)\, e^{-i2\pi k t}\, \mathrm{d}\, t. \tag{15}$$

The Fourier series of order n of f is defined as

$$S_n(t) := \sum_{k=-n}^{n} c_k e^{i2\pi k t} = \sum_{k=-n}^{n} \langle \phi_k, f \rangle \phi_k. \tag{16}$$

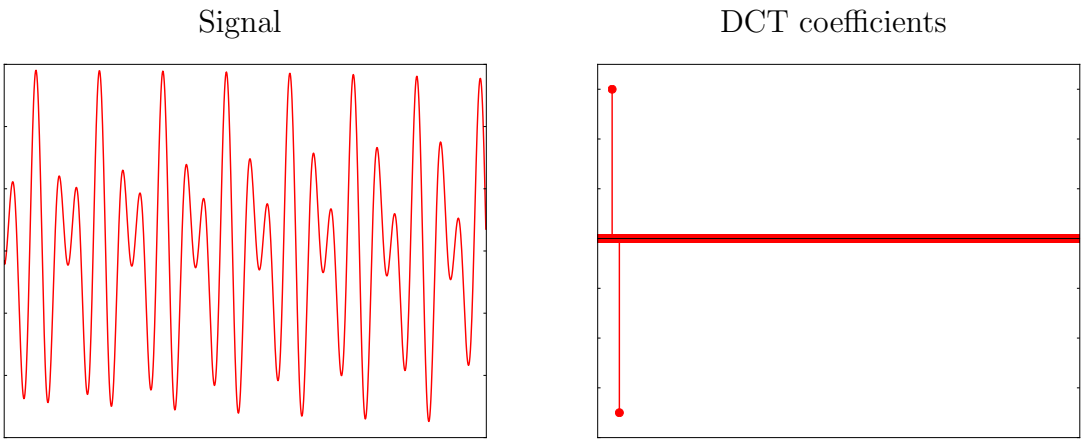


Figure 1: A signal that is sparse in the DCT dictionary.

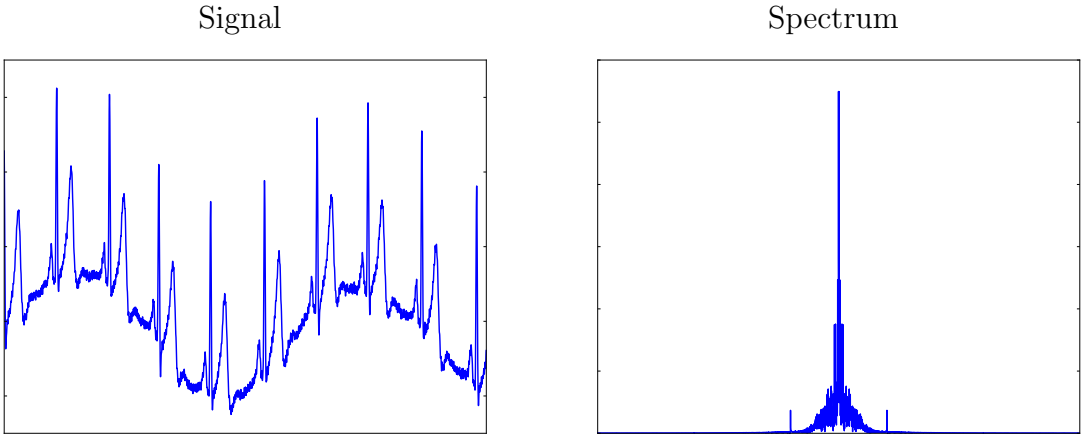


Figure 2: Electrocardiogram signal (left) and the magnitude of its spectrum (right).

It turns out that the dictionary is actually an orthonormal basis for $\mathbb{L}_2$, since

$$\lim_{n\to\infty} ||f(t) - S_n(t)|| = 0 \qquad \text{for all } f \in \mathbb{L}_2 \ . \tag{17}$$

This is a classical result, we refer to any text on Fourier analysis for the proof.

The discrete Fourier transform (DFT) is a discrete counterpart to the Fourier series. It maps vectors in $\mathbb{C}^n$ to their decomposition in terms of discrete sinusoids of the form

$$\phi_k = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 \\ e^{\frac{i2\pi k}{n}} \\ e^{\frac{i2\pi k2}{n}} \\ \cdots \\ e^{\frac{i2\pi k(n-1)}{n}} \end{bmatrix}, \quad 0 \leq k \leq n-1. \tag{18}$$

This dictionary is an orthonormal basis of C^n. The corresponding matrix is known as the DFT matrix,

$$F := \begin{bmatrix} \phi_0 & \phi_1 & \cdots & \phi_{n-1} \end{bmatrix}, \tag{19}$$

$$\text{DFT}\{x\} := Fx. \tag{20}$$

The fast Fourier transform (FFT) algorithm allows to compute the DFT in $\mathcal{O}(n \log n)$. This efficiency is crucial in practical applications. The discrete cosine transform (DCT) is a related transformation designed for real vectors; the corresponding atoms are shifted cosines instead of complex exponentials. Figure 1 shows a signal that is sparse in the DCT dictionary. Figure 2 shows the frequency representation of an electrocardiogram signal. The energy of its spectrum is highly concentrated in the low frequencies, as the signal fluctuates slowly. The representation allows to detect periodicities in the signal, which correspond to DFT coefficients with large magnitudes.

The DFT can be extended to two dimensions by considering two-dimensional sinusoidal atoms obtained by taking the outer product of the one-dimensional atoms defined by (18).

$$\phi^{\text{2D}}_{k_1,k_2} := \frac{1}{n} \begin{bmatrix} 1 & e^{\frac{i2\pi k_2}{n}} & \cdots & e^{\frac{i2\pi k_2(n-1)}{n}} \\ e^{\frac{i2\pi k_1}{n}} & e^{\frac{i2\pi (k_1+k_2)}{n}} & \cdots & e^{\frac{i2\pi (k_1+k_2(n-1))}{n}} \\ & & \cdots & \\ e^{\frac{i2\pi k_1(n-1)}{n}} & e^{\frac{i2\pi (k_1(n-1)+k_2)}{n}} & \cdots & e^{\frac{i2\pi (k_1(n-1)+k_2(n-1))}{n}} \end{bmatrix} \tag{21}$$

$$= \phi^{\text{1D}}_{k_1} \left(\phi^{\text{1D}}_{k_2}\right)^T. \tag{22}$$

To compute the 2D DFT of an image we take inner products with these 2D sinusoidal atoms, which form an orthonormal basis of $\mathbb{C}^{n\times n}$. The transform can be computed efficiently by applying 1D DFTs to the rows and columns of the array,

$$\text{DFT}^{\text{2D}}\{X\} := FXF \tag{23}$$

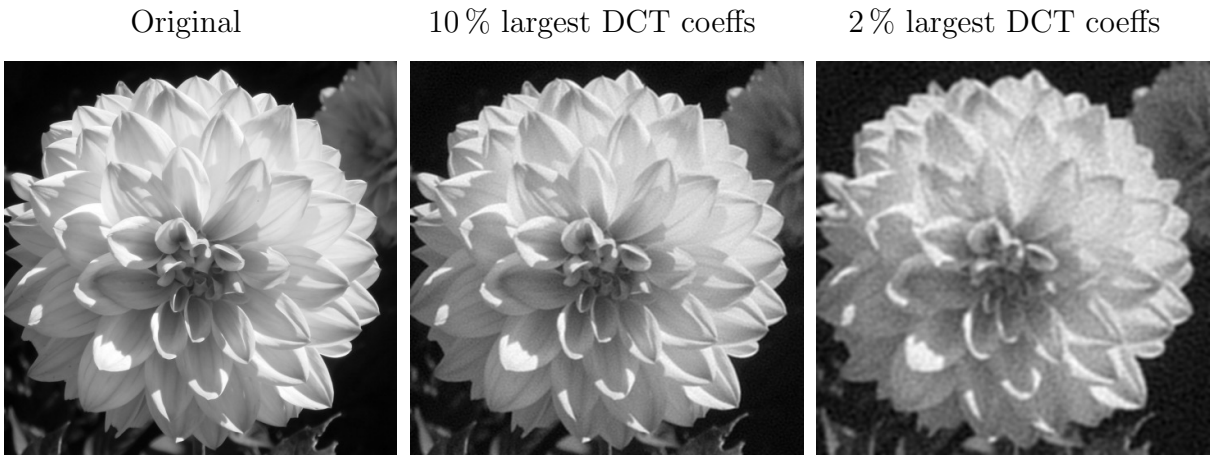


Figure 3: Discarding the smallest DCT coefficients of an image allows to compress it quite effectively.

The 2D frequency representation of images tends to be sparse. In particular, most of the energy of the image tends to be concentrated in the lower frequencies. This insight can be used to compress the image by just retaining the coefficients with larger magnitudes. Figure 3 shows that this simple scheme can be quite effective. The JPEG compression standard is based on a similar idea: high-frequency coefficients are discarded according to a perceptual model.

2.2 Short-time Fourier transform (STFT)

The Fourier series or the DFT provide global information about the periodicities of a signal, but they do not capture localized periodicities. However, the spectrum of speech, music and other sound signals changes with time. To analyze the changes in the spectrum we can compute the DFT of time segments of the signal. However, doing this in a naïve way may introduce spurious high-frequencies, as shown in Figure 4. Multiplying the time segment with a window that tapers off at the ends smoothens the transitions and avoids introducing high-frequency artifacts.

The short-time Fourier transform (STFT) is defined as the Fourier series coefficients of the pointwise product between a function and a shifted window $w : [0, 1] \to \mathbb{C}$ approximately localized in time and frequency

$$\text{STFT}\left\{f\right\}(k, \tau) := \int_0^1 f(t)\, \overline{w(t-\tau)} e^{-i2\pi kt}\, \mathrm{d}\, t. \tag{24}$$

Equivalently, the STFT coefficients are equal to the inner product between the signal and atoms of the form $\phi_{k,\tau}(t) := w(t-\tau)\, e^{i2\pi kt}$, which corresponds to copies of w shifted by τ

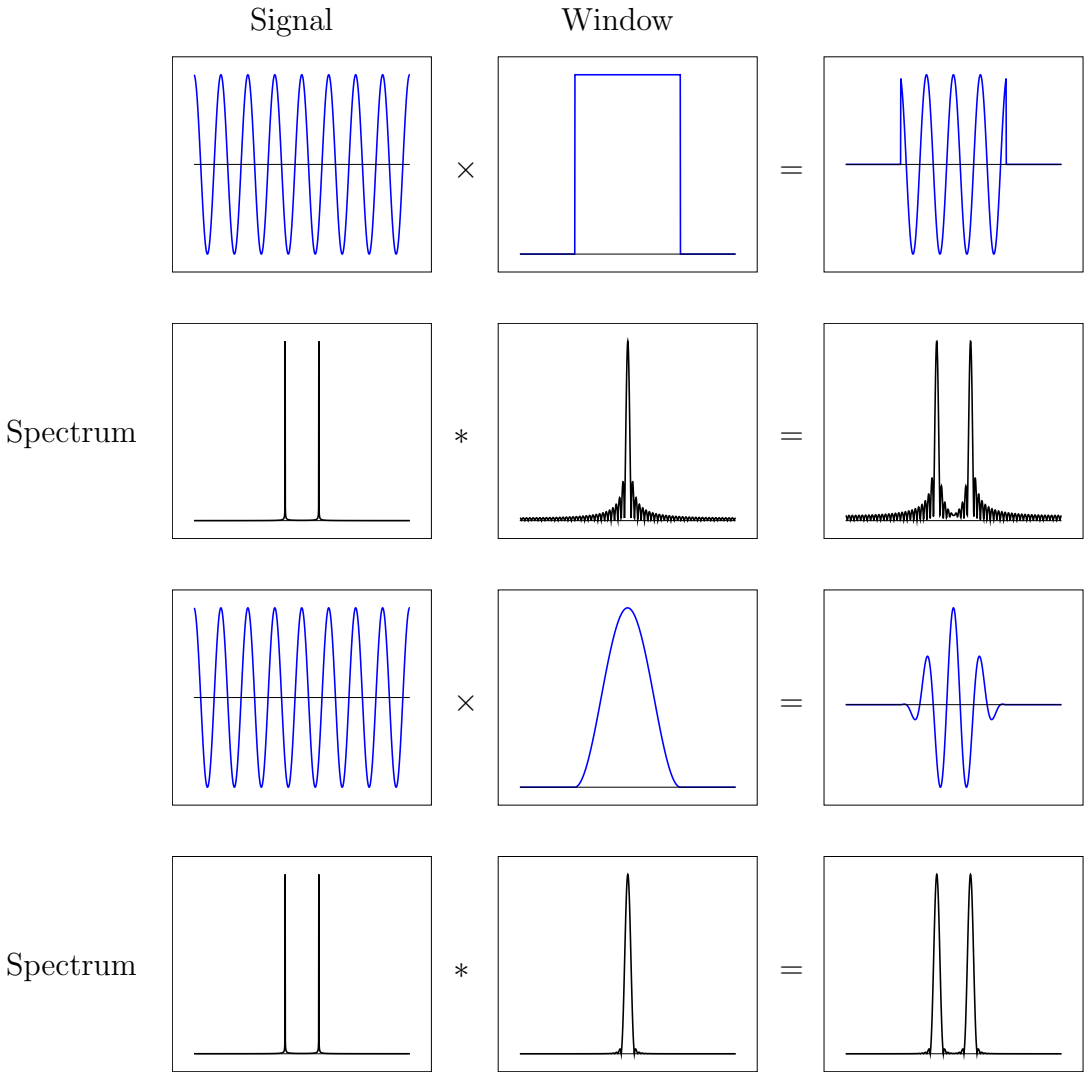


Figure 4: The spectrum of a time segment may contain spurious high-frequency content produced by the sudden transition at the ends of the segment. In the frequency domain, the spectrum is being convolved by a sinc function, which has a very heavy tail. Multiplying the signal by a localized window that has a faster decay in the frequency domain alleviates the problem.

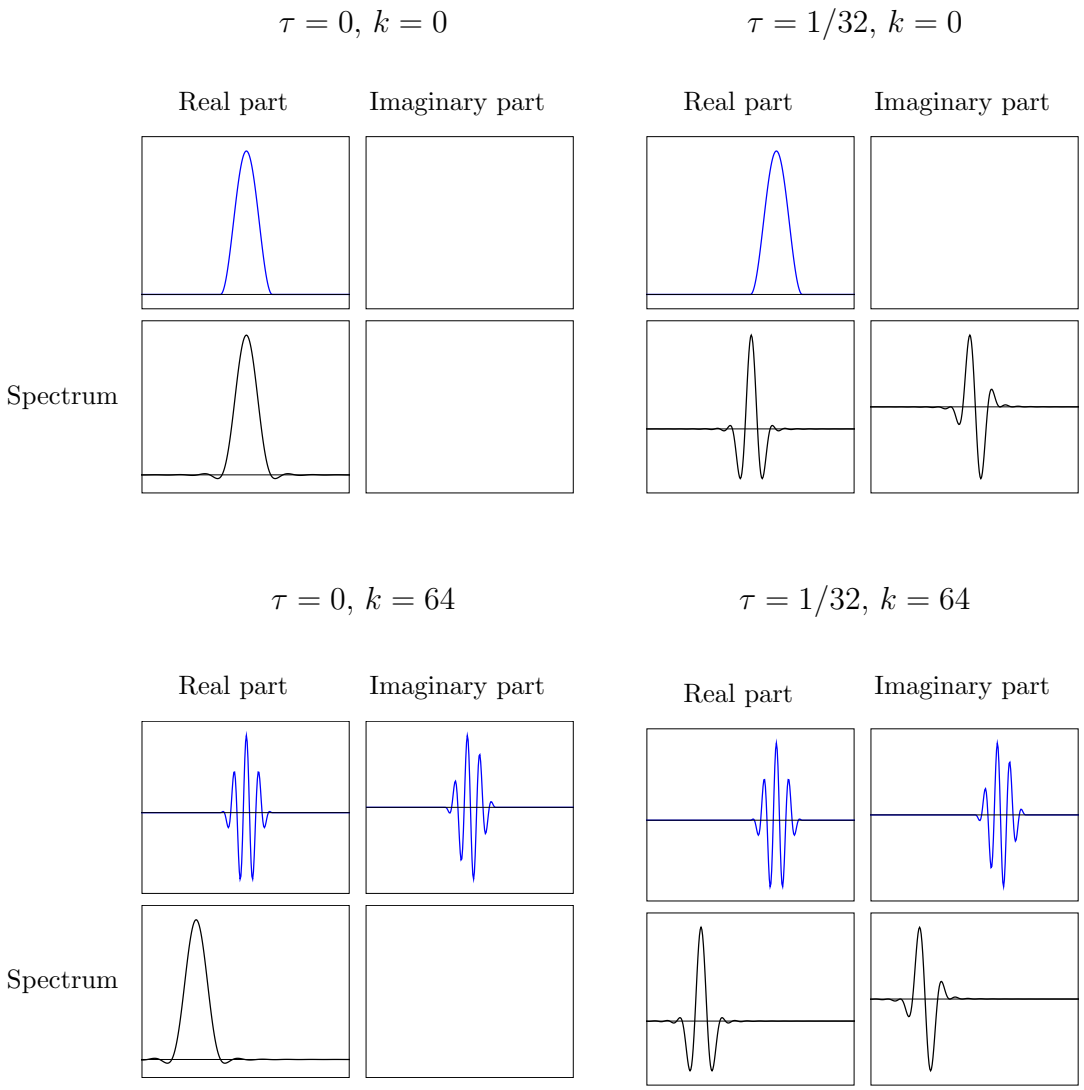


Figure 5: Atoms in the STFT dictionary.

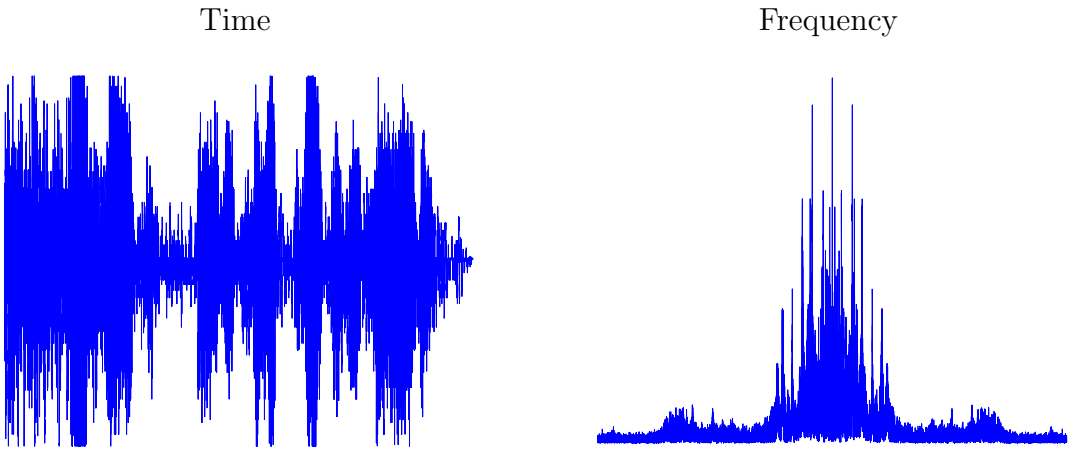


Figure 6: Time and frequency representation of a speech signal.

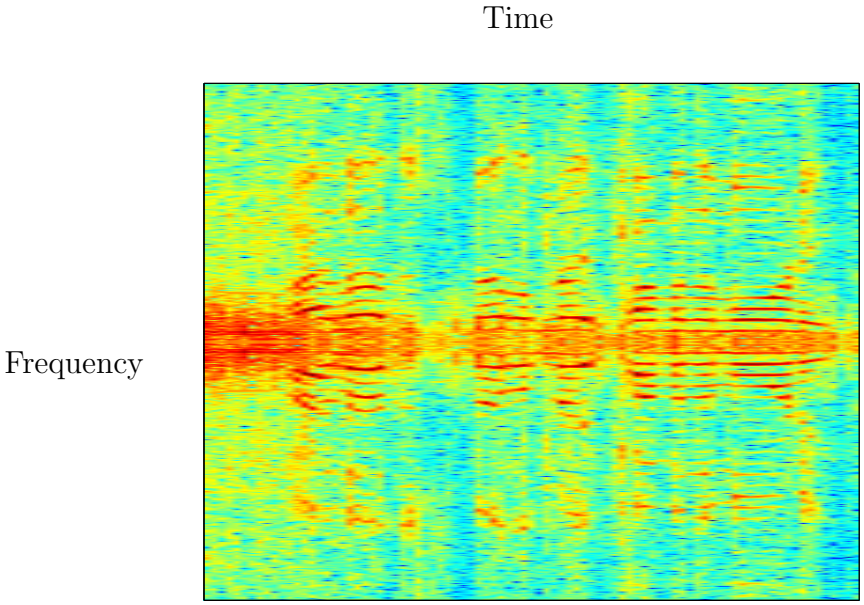


Figure 7: Spectrogram (log magnitude of STFT coefficients) of the speech signal in Figure 6.

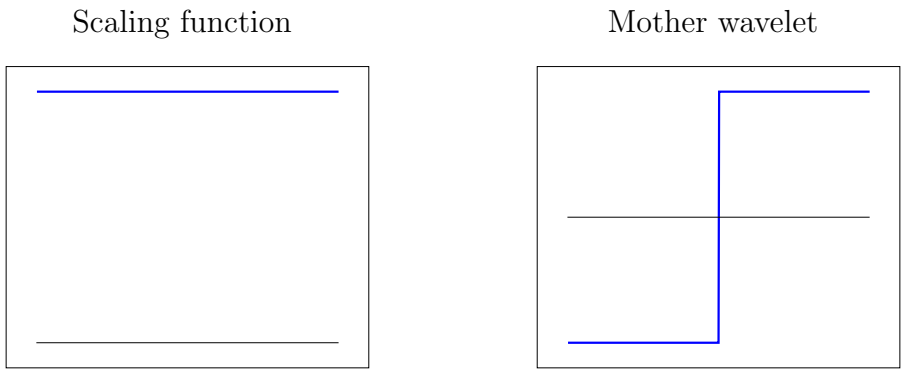


Figure 8: Scaling function and mother wavelet of the Haar wavelet transform.

in time and by k in frequency. Some examples are shown in Figure 5. Including dilations of w (in addition to time and frequency translations) yields a dictionary of Gabor atoms.

The discrete-time STFT consists of pointwise multiplication by a shifted window followed by a DFT. This is equivalent to computing $D^T x$, where $D \in \mathbb{C}^{n \times m}$, $m > n$ is an overcomplete dictionary. The corresponding analysis sparse model is very useful for speech analysis. The logarithm of the magnitude of the STFT coefficients, called an spectrogram, is widely used for sound processing. Figure 7 shows the spectrogram of a real speech signal. The time and frequency representation of the same signal are shown in Figure 6.

2.3 Wavelets

Wavelets are atoms that allow to capture signal structure at different scales. A wavelet ψ is a unit-norm, zero-mean function in $\mathbb{L}_2$. The wavelet transform of a function $f \in \mathbb{L}_2$ is defined as

$$\mathrm{W}\left\{f\right\}(s,\tau) := \frac{1}{\sqrt{s}} \int f(t)\, \overline{\psi\left(\frac{t-\tau}{s}\right)}\, \mathrm{d}\, t = \langle \phi_{s,\tau}, f \rangle \,. \tag{25}$$

The wavelet coefficients are obtained by taking the inner product with atoms that are shifted and dilated copies of the *mother* wavelet ψ

$$\phi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right). \tag{26}$$

The wavelet transform coefficients provide information about the signal at different scales. This rather vague statement can be made precise using the multiresolution framework of Mallat and Meyer.

Definition 2.1 (Multiresolution approximation)**.** *A multi resolution approximation is a sequence* $\{\mathcal{V}_j, j \in \mathbb{Z}\}$ *of closed subspaces of* $\mathbb{L}_2(\mathbb{R})$ *satisfying the following conditions.*

- *Dilating functions in* $\mathcal{V}_j$ *by 2 yields functions in* $\mathcal{V}_{j+1}$

$$f(t) \in \mathcal{V}_j \iff f\left(\frac{t}{2}\right) \in \mathcal{V}_{j+1}. \tag{27}$$

- *Approximations at a scale* 2^j *are always better than at* 2^{j+1}

$$\mathcal{V}_{j+1} \subset \mathcal{V}_j. \tag{28}$$

- $\mathcal{V}_j$ *is invariant to translations at the scale* 2^j

$$f(t) \in \mathcal{V}_j \iff f\left(t - 2^j k\right) \in \mathcal{V}_j \qquad \text{for all } k \in \mathbb{Z}. \tag{29}$$

- *As* $j \to \infty$ *the approximation loses all information*

$$\lim_{j \to \infty} \mathcal{V}_j = \{0\}. \tag{30}$$

- *As* $j \to -\infty$ *the approximation is perfect*

$$\lim_{j \to -\infty} \mathcal{V}_j = \mathbb{L}_2. \tag{31}$$

- *There exists a scaling function* $\zeta \in \mathcal{V}_0$ *such that*

$$\{\zeta_{0,k}(t) := \zeta(t-k),\ k \in \mathbb{Z}\} \tag{32}$$

  is an orthonormal basis for $\mathcal{V}_0$*.*

Under these conditions, we can interpret the projection $\mathcal{P}_{\mathcal{V}_j}(f)$ of a function f onto $\mathcal{V}_j$ as an approximation of f at scale 2^j. In a remarkable result, Mallat and Meyer prove that for any multiresolution approximation there exists a wavelet ψ such that

$$\mathcal{P}_{\mathcal{V}_j}(f) = \mathcal{P}_{\mathcal{V}_{j+1}}(f) + \sum_{k \in \mathbb{Z}} \left\langle \psi_{2^j,k}, f \right\rangle \psi_{2^j,k}. \tag{33}$$

The particular wavelet ψ that yields this orthonormal basis depends on the scaling function. Figure 8 shows the scaling function and the corresponding mother wavelet for the Haar wavelet transform. The dictionary of shifted wavelets dilated by 2^j $\left\{\psi_{2^j,k},\, k \in \mathbb{Z}\right\}$ is an orthonormal basis for $\mathcal{V}_j \cap \mathcal{V}_{j+1}^{\perp}$, i.e. the subspace in $\mathcal{V}_j$ that contains the functions that

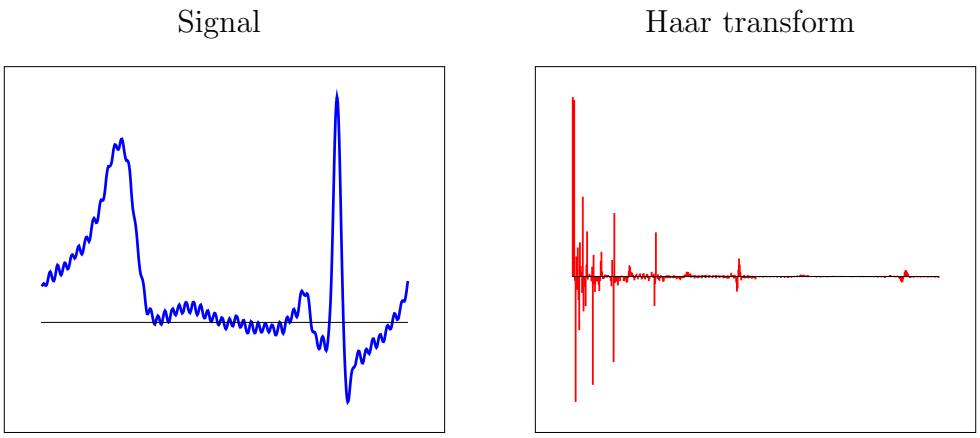


Figure 9: Haar wavelet coefficients (right) of an electrocardiogram signal (left).

are not available at coarser scales. As a result, $\left\{\zeta_{0,k}\left(t\right), \psi_{2^1,k}, \psi_{2^2,k}, \ldots, \psi_{2^j,k},\, k \in \mathbb{Z}\right\}$ is an orthonormal basis for $\mathcal{V}_j$.

Wavelet bases can be discretized to obtain orthonormal bases of $\mathbb{C}^n$ or $\mathbb{R}^n$. These discrete transforms can be computed in $\mathcal{O}\left(n\right)$. Figure 9 shows the Haar wavelet coefficients of an electrocardiogram signal. The corresponding multiresolution approximations at the different scales are shown in Figures 10, 11 and 12.

A signal-processing interpretation of the wavelet transform is that the scaling function acts as a low-pass filter, whereas the dilated and shifted wavelets act as band-pass filters in different bands. Many other wavelet bases apart from the Haar exist: Meyer, Daubechies, Battle-Lemarie, ... We refer the interested reader to [3] for more information. Chapter 7 provides a detailed and rigorous description of the construction of orthonormal wavelet bases from a multiresolution approximation.

Two-dimensional wavelets can be obtained by taking outer products of one-dimensional wavelets, as we did for the DFT.

$$\phi^{\text{2D}}_{s_1,s_2,k_1,k_2} := \phi^{\text{1D}}_{s_1,k_1} \left(\phi^{\text{1D}}_{s_2,k_2}\right)^T \tag{34}$$

The corresponding two-dimensional transform allows to obtain sparse representations of natural images. An example is shown in Figure 13. The wavelet coefficients at finer scales are mostly zero in most areas of the image. Figure 14 shows the sorted magnitudes of the coefficients on a logarithmic scale. A large majority of the coefficients are very small and can be discarded without significantly affecting the quality of the image. The JPEG 2000 compression standard is based this insight.

Finally we would like to mention that designing multidimensional transforms that are more effective at providing sparse representations for images has been a vibrant research subject

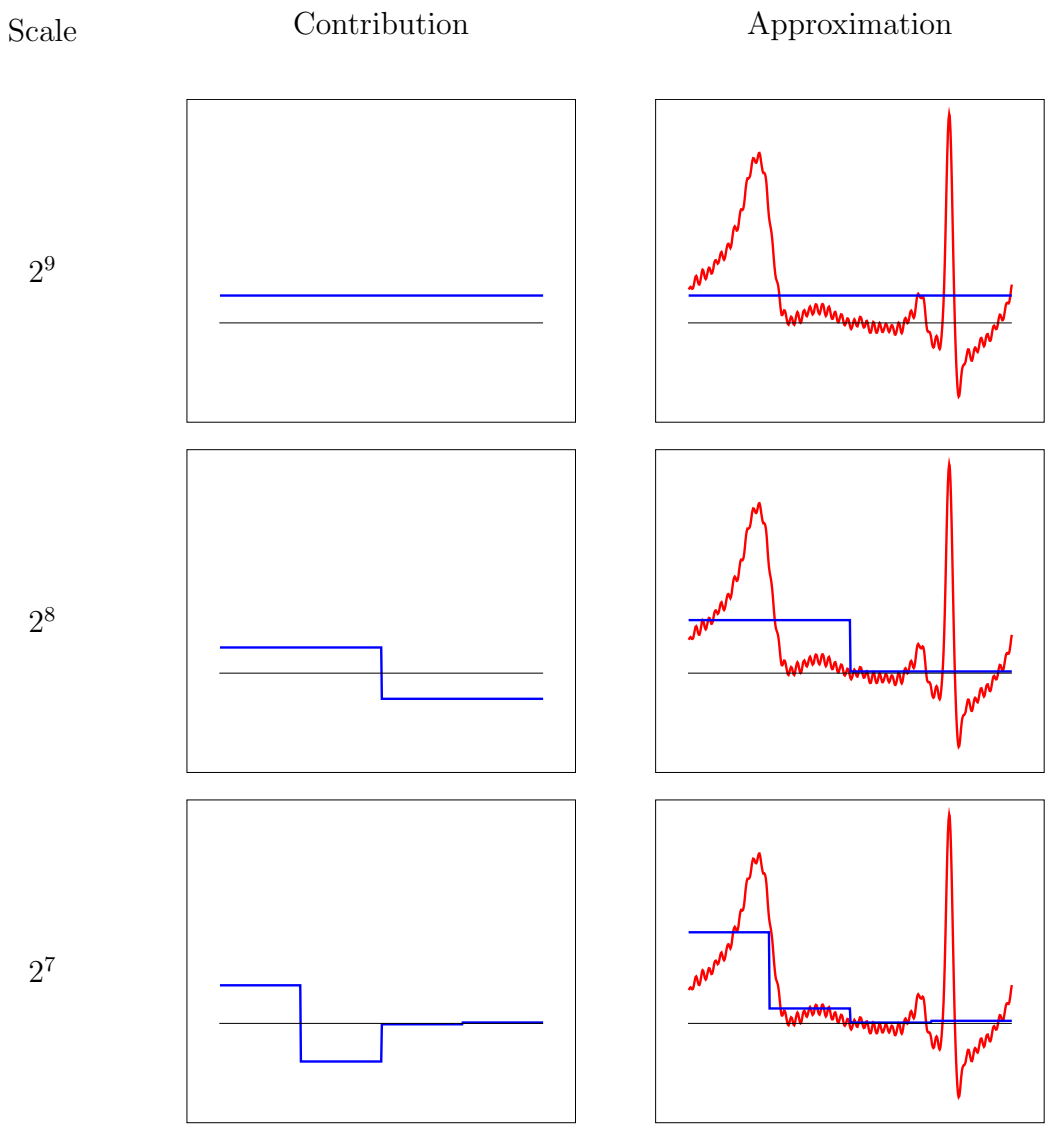


Figure 10: Approximation of the electrocardiogram signal in Figure 9 using a Haar multiresolution approximation at different scales (right). On the left, we can see the projection of the signal onto $\mathcal{V}_j \cap \mathcal{V}_{j+1}^{\perp}$, which captures the information in the signal at scale 2^j.

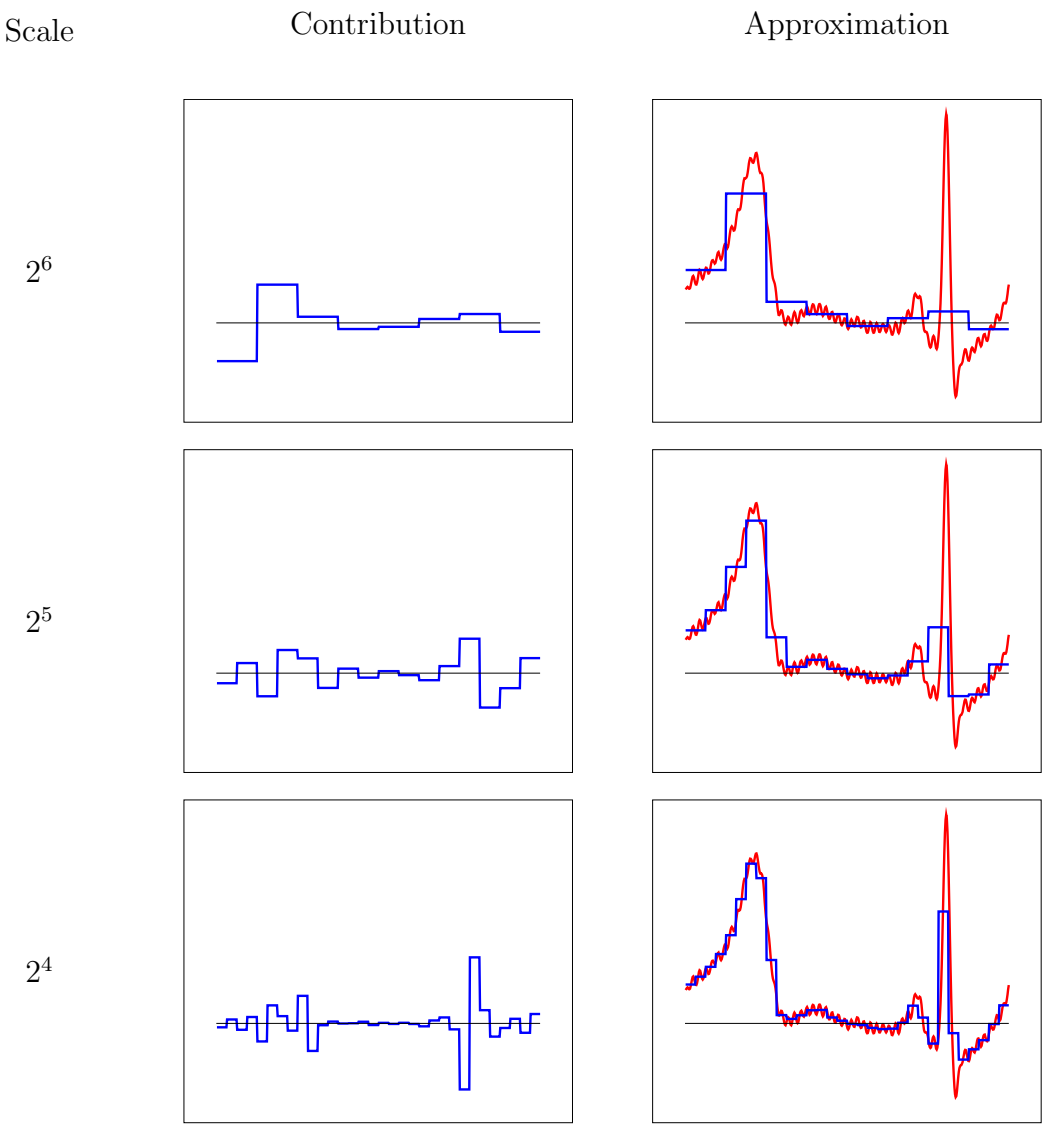


Figure 11: Approximation of the electrocardiogram signal in Figure 9 using a Haar multiresolution approximation at different scales (right). On the left, we can see the projection of the signal onto $\mathcal{V}_j \cap \mathcal{V}_{j+1}^{\perp}$, which captures the information in the signal at scale 2^j.

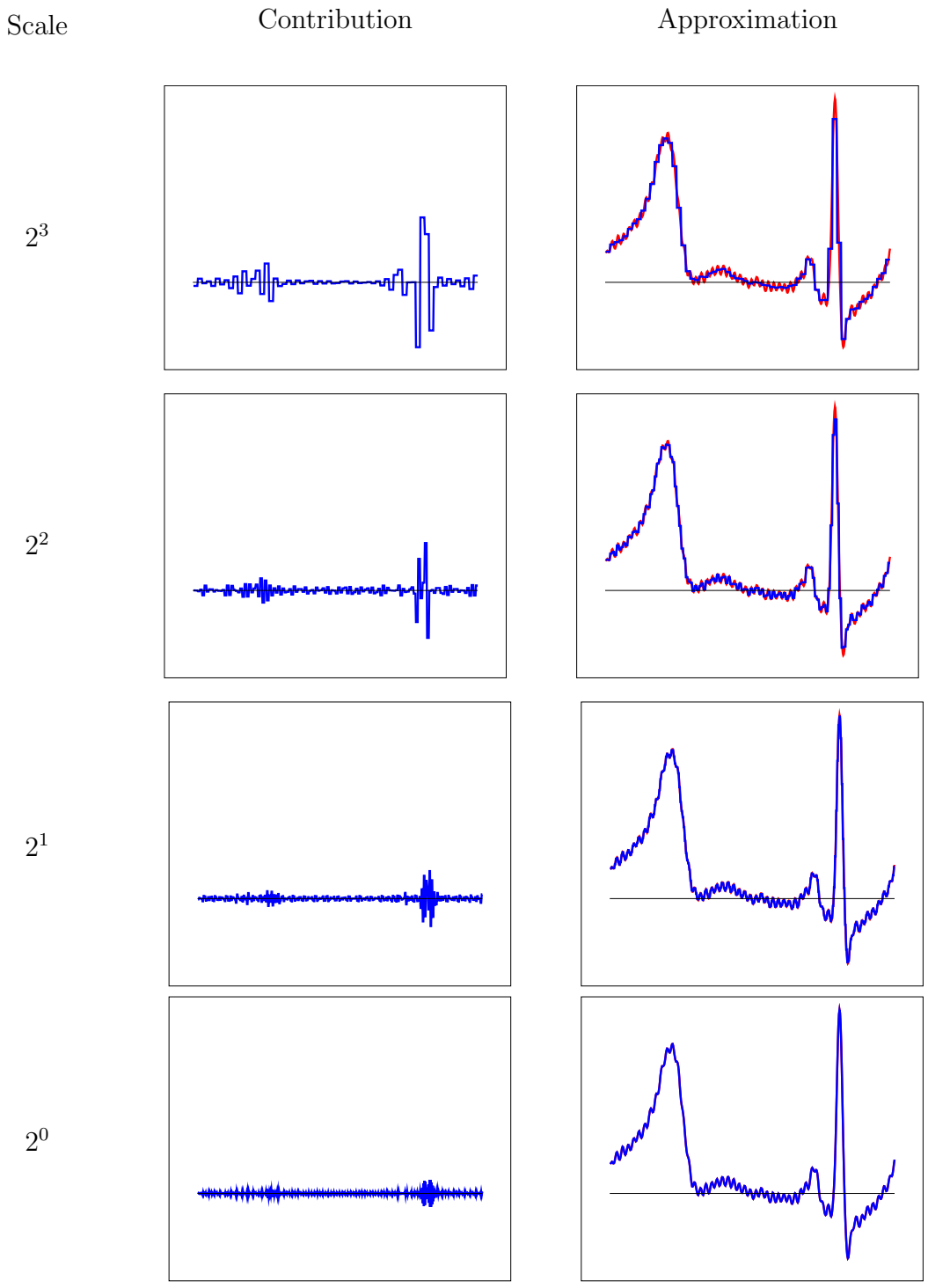


Figure 12: Approximation of the electrocardiogram signal in Figure 9 using a Haar multiresolution approximation at different scales (right). On the left, we can see the projection of the signal onto $\mathcal{V}_j \cap \mathcal{V}_{j+1}^{\perp}$, which captures the information in the signal at scale 2^j.

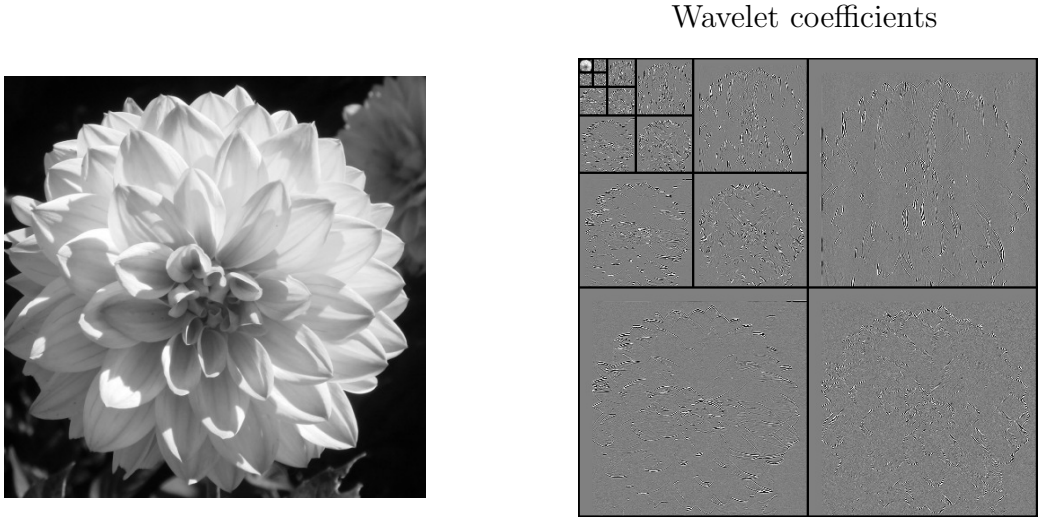


Figure 13: Coefficients in a wavelet basis (right) of a natural image (left).

for many years. Some of these extensions include the steerable pyramid, ridgelets, curvelets, and bandlets. We refer to Section 9.3 in [3] for more details.

3 The synthesis model

Overcomplete dictionaries provide greater flexibility for signal representation than orthonormal transforms, but this comes at a cost. Computing the corresponding coefficients and using the corresponding sparse models to process data is not as simple or computationally efficient. We consider an overcomplete dictionary D with linearly independent atoms $\{\phi_1, \ldots, \phi_m \in \mathbb{R}^m\}$ that are linearly independent

$$D := \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_m \end{bmatrix}, \qquad m > n. \tag{35}$$

The synthesis sparse model assumes that a signal $x \in \mathbb{R}^n$ can be represented as

$$x = Dc, \tag{36}$$

where c is sparse. Unfortunately, even if such a sparse c exists, it is not easy to compute it from x. There reason is that there are infinite choices of coefficient vectors c' such that $x = Dc'$ and most of them are not sparse at all!

A possible choice for the coefficient vector is given by the following lemma, which is proved in Section A.1 of the appendix.

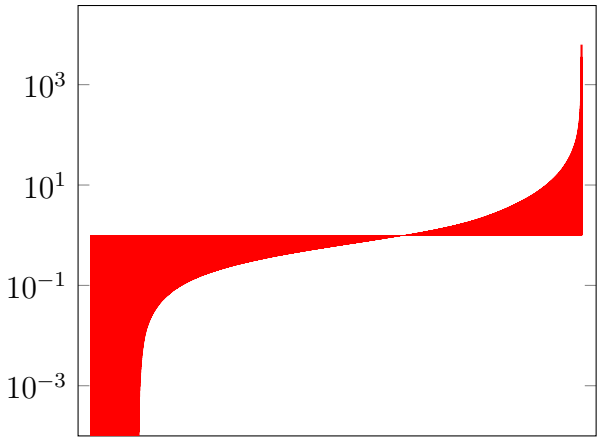


Figure 14: Sorted magnitude of the wavelet coefficients shown in Figure 13 plotted on a logarithmic scale. The image is highly compressible in the wavelet domain.

Lemma 3.1 (Minimum ℓ_2-norm solution)**.** *The coefficient vector with minimum ℓ_2 norm satisfying $x = Dc$ is*

$$c_{\ell_2} := D^T \left(D D^T \right)^{-1} x, \tag{37}$$

which is the projection of c onto the row space of D.

In the case of complex signals, the same result holds replacing D^T by D^*.

Unfortunately, the minimum ℓ_2-norm solution is often very dense. Let us consider a concrete example. We define the atoms of an overcomplete dictionary of sinusoids as

$$\phi_k(j) := \frac{1}{\sqrt{n}} e^{\frac{i 2\pi k j}{m}}, \quad 1 \leq k \leq m, \, 1 \leq j \leq n. \tag{38}$$

If $m = n$ this is an orthonormal DFT basis, but for $m > n$ the sinusoidal atoms are correlated and the dictionary is overcomplete. Figure 15 shows a signal that is sparse in this dictionary and the corresponding coefficient vector. Note that this signal would *not* be sparse in a DFT basis. As we can see in Figure 16, the minimum ℓ_2-norm coefficient vector is not sparse.

Ideally, we would like to find the coefficient vector with the smallest number of nonzeros that corresponds to the signal. More precisely, our aim is to solve

$$\text{minimize} \quad ||\tilde{c}||_0 \tag{39}$$

$$\text{subject to} \quad x = D\,\tilde{c} \tag{40}$$

Unfortunately, this optimization problem is computationally intractable. Intuitively, we cannot do much better than try out all the possibilities. However there are tractable methods that are able to produce sparse coefficients in practice. They are divided in two main classes.

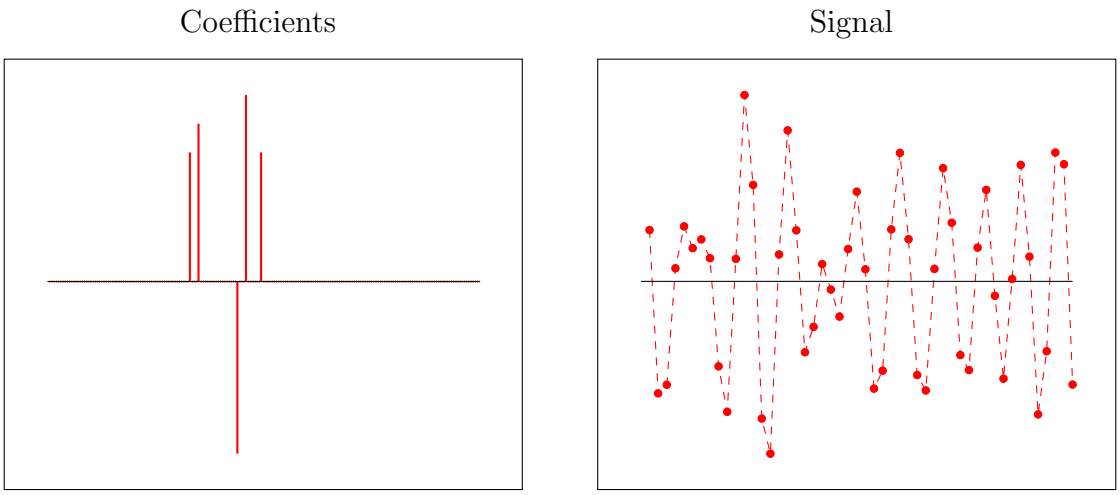


Figure 15: A signal that is sparse in an overcomplete dictionary of sinusoids (left) and its corresponding coefficients (right).

- Greedy methods which select atoms one by one.
- Methods based on solving a related convex program, usually ℓ_1-norm minimization.

In general, the performance of these methods deteriorates for dictionaries with more correlated atoms. This is not surprising because correlations *tangle* the contributions of the different atoms making more difficult to obtain a sparse solution efficiently.

3.1 Greedy methods

Matching pursuit (MP) [4] is a very simple method for obtaining a sparse coefficient vector. We initialize a residual vector to equal the signal. Then we iteratively choose the atom that is most correlated with the residual and subtract the component of the residual in that direction.

Algorithm 3.2 (Matching pursuit)**.** *Given a dictionary $D \in \mathbb{R}^{n\times m}$ (or $\mathbb{C}^{n\times m}$) and a signal $x \in \mathbb{R}^n$ (or $\mathbb{C}^n$), we initialize the residual and the approximation by setting,*

$$r^{(0)} := x, \tag{41}$$

$$\hat{x}^{(0)} := 0, \tag{42}$$

$$\mathcal{I}^{(0)} = \emptyset. \tag{43}$$

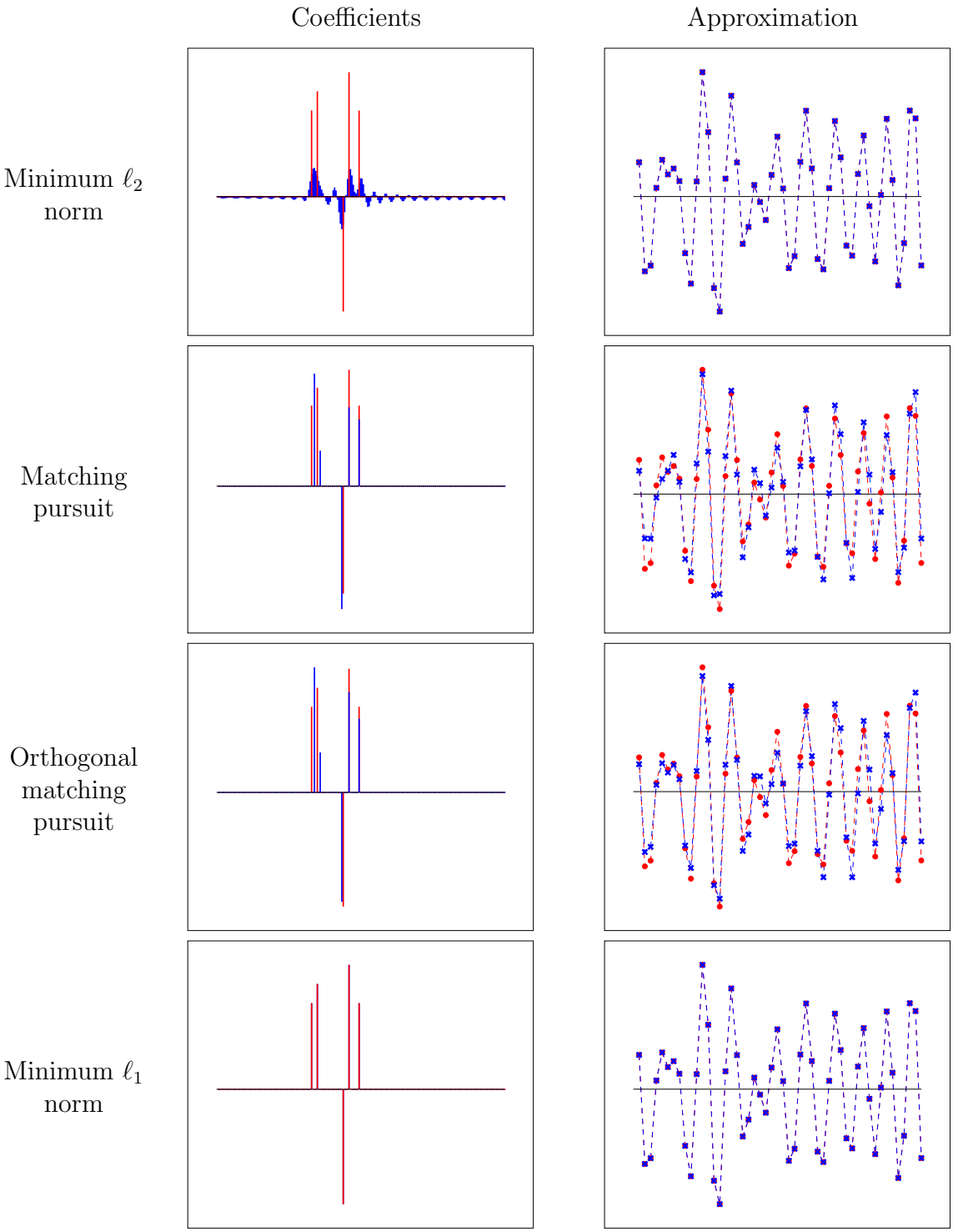

Coefficients
Approximation
Minimum ℓ_2 norm
Matching pursuit
Orthogonal matching pursuit
Minimum ℓ_1 norm

Figure 16: .

Then for a fixed number of iterations $k = 1, 2, \ldots, N$ we compute

$$\phi^{(k)} := \arg \max_{j \in \{1,2,\ldots,m\}/\mathcal{I}^{(k-1)}} \left| \left\langle r^{(k-1)}, \phi_j \right\rangle \right|, \tag{44}$$

$$\hat{x}^{(k)} := \hat{x}^{(k-1)} + \left\langle r^{(k-1)}, \phi^{(k)} \right\rangle \phi^{(k)}, \tag{45}$$

$$r^{(k)} := r^{(k-1)} - \left\langle r^{(k-1)}, \phi^{(k)} \right\rangle \phi^{(k)}, \tag{46}$$

$$\mathcal{I}^{(k)} := \mathcal{I}^{(k-1)} \cup \{j\} \quad \textit{where } j \textit{ is the index such that } \phi^{(k)} = \phi_j. \tag{47}$$

The output is a sparse approximation to the signal $\hat{x}^{(N)}$ and the corresponding coefficient indices $\mathcal{I}^{(N)}$.

If the atoms form an orthonormal basis, MP produces the exact coefficient vector in a number of iterations that is equal to its cardinality. However, for more correlated dictionaries it might choose wrong atoms (that may be very correlated with the actual atoms that form the signal), as shown in Figure 16.

The approximation obtained by MP at every iteration is not necessarily optimal in terms of ℓ_2-norm. Indeed, given a set of atoms

$$D_{\mathcal{I}^{(N)}} := \begin{bmatrix} \phi^{(1)} & \phi^{(2)} & \cdots & \phi^{(N)} \end{bmatrix} \tag{48}$$

where $N < m$, the coefficients that yield the best ℓ_2-norm approximation are equal to the least-squares estimate

$$c_{\text{ls}} := D^{\dagger}_{\mathcal{I}^{(k)}} x \tag{49}$$

$$= \left(D^T_{\mathcal{I}^{(N)}} D_{\mathcal{I}^{(N)}} \right)^{-1} D^T_{\mathcal{I}^{(N)}} x. \tag{50}$$

In contrast, MP computes

$$\left(c_{\text{MP}} \right)_k = \left\langle \phi^{(k)}, x - \sum_{j=1}^{k-1} \left(c_{\text{MP}} \right)_j \phi^{(j)} \right\rangle. \tag{51}$$

Setting the coefficients to equal (49) ensures that the residual and the approximation are orthogonal. This is the strategy followed by orthogonal matching pursuit (OMP) [5].

Algorithm 3.3 (Orthogonal matching pursuit)**.** *Given a dictionary $D \in \mathbb{R}^{n \times m}$ (or $\mathbb{C}^{n \times m}$) and a signal $x \in \mathbb{R}^n$ (or $\mathbb{C}^n$), we initialize the residual and the approximation by setting,*

$$r^{(0)} := x, \tag{52}$$

$$\mathcal{I}^{(0)} = \emptyset. \tag{53}$$

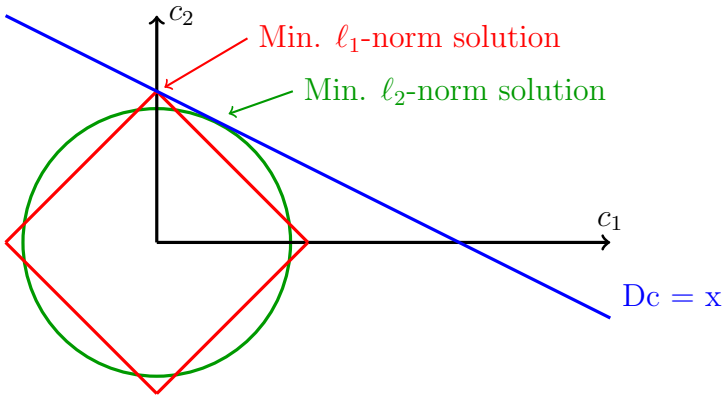


Figure 17: The minimum ℓ_1-norm solution is sparser than the minimum ℓ_2-norm solution because of the geometry of the ℓ_1-norm and ℓ_2-norm balls.

Then for a fixed number of iterations $k = 1, 2, \ldots, N$ *we compute*

$$\phi^{(k)} := \arg \max_{j \in \{1,2,\ldots,m\}/\mathcal{I}^{(k-1)}} \left| \left\langle r^{(k-1)}, \phi_j \right\rangle \right|, \tag{54}$$

$$\mathcal{I}^{(k)} := \mathcal{I}^{(k-1)} \cup \{j\} \quad \textit{where } j \textit{ is the index such that } \phi^{(k)} = \phi_j, \tag{55}$$

$$D_{\mathcal{I}^{(k)}} := \left[\phi^{(1)} \quad \phi^{(2)} \quad \cdots \quad \phi^{(k)} \right], \tag{56}$$

$$\hat{c}^{(k)} := D^{\dagger}_{\mathcal{I}^{(k)}} x, \tag{57}$$

$$\hat{x}^{(k)} := D_{\mathcal{I}^{(k)}} \hat{c}^{(k)}, \tag{58}$$

$$r^{(k)} := x - \hat{x}^{(k)}. \tag{59}$$

The output is a sparse approximation to the signal $\hat{x}^{(N)}$ *and the corresponding coefficient indices* $\mathcal{I}^{(N)}$.

OMP produces better approximations to the signal than MP, but it may still choose the wrong atoms if the dictionary is highly correlated, as is the case in Figure 16.

3.2 ℓ_1-norm minimization

As shown in Figure 16 minimizing the ℓ_2 norm of the coefficient vector does not tend to yield a sparse solution. However, minimizing the ℓ_1 norm, i.e. solving the problem

$$\text{minimize} \quad ||\tilde{c}||_1 \tag{60}$$

$$\text{subject to} \quad x = D\,\tilde{c} \tag{61}$$

often does. This approach is known as basis pursuit [2] in the literature. As we have seen in previous lectures, the optimization problem is convex and can be recast as a linear program.

It is therefore computationally tractable although significantly more computationally costly than greedy methods such as MP or OMP. ℓ_1-norm minimization achieves better results than these methods in some cases, see for instance Figure 16, but it may also fail to find sparse solutions in highly correlated dictionaries.

Figure 17 provides some geometric intuition as to why minimizing the ℓ_1 norm is better than minimizing the ℓ_2 norm when we aim to obtain sparse solutions under linear constraints. The ℓ_1-norm ball is more concentrated around the axes than the ℓ_2-norm ball. It is therefore more likely for the line representing the constraint $x = Dc$ to be tangent to the ball on an axis, where the solution has cardinality one instead of two. As a result, the minimum ℓ_1-norm solution is sparser than the minimum ℓ_2-norm solution. This intuition generalizes to higher dimensions.

4 Denoising

4.1 The denoising problem

The aim of denoising is to extract a signal from data that is corrupted by uninformative perturbations, which we call noise. We will focus on the additive noise model

$$\text{data} = \text{signal} + \text{noise}, \tag{62}$$

$$y = x + z. \tag{63}$$

In order to achieve denoising it is necessary to have some prior knowledge about the structure of the signal and the structure of the noise. As an example, in Figure 18, an electrocardiogram recording is corrupted by high-frequency perturbations. In the frequency domain, we can see a small peak at 60 Hz due to noise coming from the power grid. To eliminate this noise we enforce the prior that the signal of interest is essentially low pass by filtering out the high end of the spectrum.

4.2 Thresholding

Dictionaries and transforms such as the DFT, STFT and the wavelet transform allow to obtain sparse representations of images, music, speech and other signals. These dictionaries exploit structure that is expected to be present in the signal but not in the noise. As a result the noise component in the data will not have a sparse representation most of the time; it is *incoherent* with the dictionary atoms. This can be exploited to denoise the signal via thresholding.

Definition 4.1 (Hard thresholding)**.** *Let $x \in \mathbb{R}^n$. The hard-thresholding operator $\mathcal{H}_\eta : \mathbb{R}^n \to \mathbb{R}^n$ sets to zero any entries in x with magnitude smaller than a predefined real-valued*

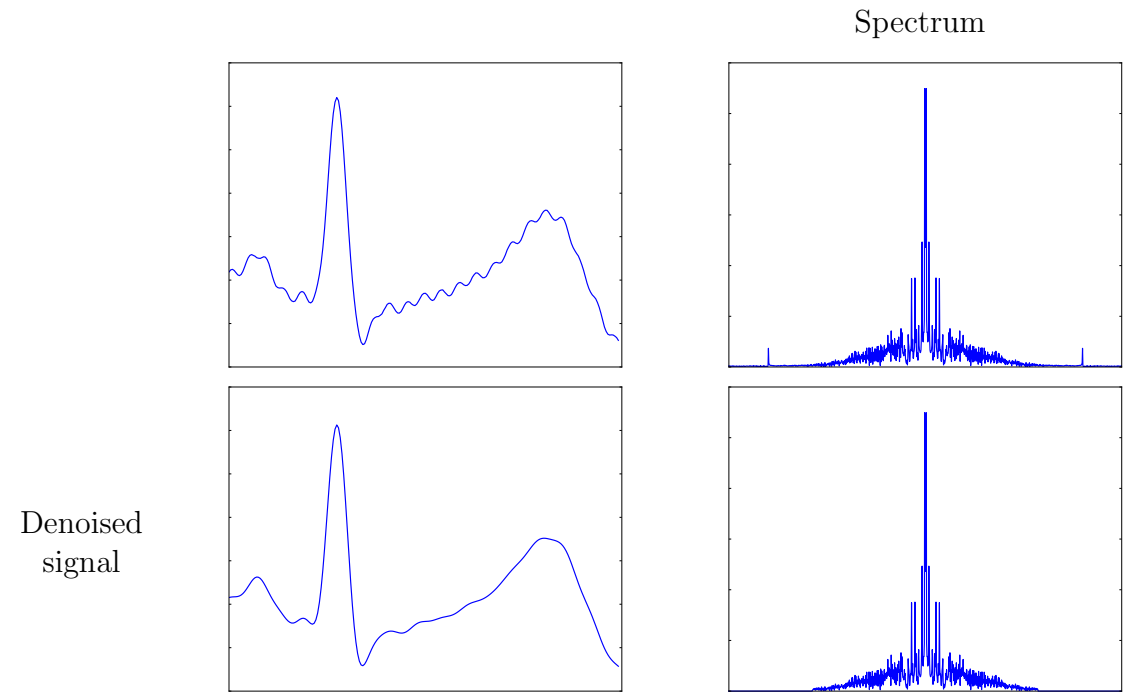


Figure 18: Electrocardiogram recording denoised via low-pass filtering.

threshold $\eta > 0$,

$$\mathcal{H}_\eta(x)_i := \begin{cases} x_i & \text{if } |x_i| > \eta, \\ 0 & \text{otherwise.} \end{cases} \tag{64}$$

The idea is very simple. We first map the signal to a domain where it is sparse, but the noise is dense. Then we discard all the coefficients below a certain threshold, as illustrated in Figure 19. In particular, if the signal is sparse in a basis, we threshold the coefficients $B^{-1}y$,

$$\hat{c} = \mathcal{H}_\eta\left(B^{-1}y\right) \tag{65}$$

$$= \mathcal{H}_\eta\left(c + B^{-1}z\right), \tag{66}$$

$$\hat{x} = B\hat{c}. \tag{67}$$

For the method to work $B^{-1}z$ must not be sparse. We can actually prove that this is the case if the basis is orthonormal and the noise is iid Gaussian.

Lemma 4.2 (Thresholding Gaussian noise)**.** *If z is an n-dimensional iid Gaussian with zero mean and variance σ^2, then for any orthogonal matrix $U \in \mathbb{R}^n$ $U^T z$ is iid Gaussian with zero mean and variance σ^2.*

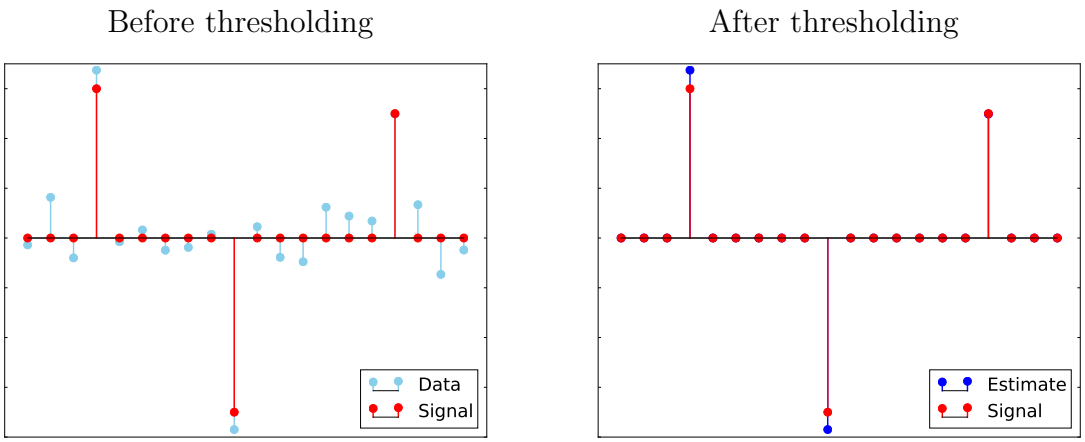


Figure 19: Denoising via hard thresholding.

Proof. By elementary properties of Gaussian random vectors, $U^T z$ is Gaussian with zero mean and covariance matrix

$$\Sigma = U^T \left(\sigma^2 \, \mathrm{I}\right) U = \sigma^2 U^T U = \sigma^2 \, \mathrm{I} \, . \tag{68}$$

□

Figures 20 and 21 show the results of applying this denoising method to the signals in Figures 1 and 13. Both signals are corrupted by additive Gaussian noise. In both cases, exploiting the sparse decomposition allows us to denoise the data very effectively.

Thresholding can also be applied in conjunction with a sparse analysis model. If we suspect that the inner products between a signal x and the atoms in a dictionary D are mostly zero then we can threshold $D^T y$ to denoise the coefficients,

$$\hat{c} = \mathcal{H}_\eta \left(D^T y\right) \tag{69}$$

$$= \mathcal{H}_\eta \left(D^T x + D^T z\right) , \tag{70}$$

$$\tag{71}$$

This method will be effective if $D^T z$ is dense (recall that z denotes the noise component). Recovering an approximation to the signal from the thresholded vector $\mathcal{H}_\eta \left(D^T y\right)$ requires applying a left inverse L that might introduce some distortion,

$$\hat{x} = L\hat{c}, \qquad LD^T = \mathrm{I} \, . \tag{72}$$

An alternative approach is to enforce the analysis model within an optimization problem, as we explain in Section **??**. Thresholding STFT coefficients is a popular denoising technique in speech processing. Figures 24 and **??** show an example with real data.

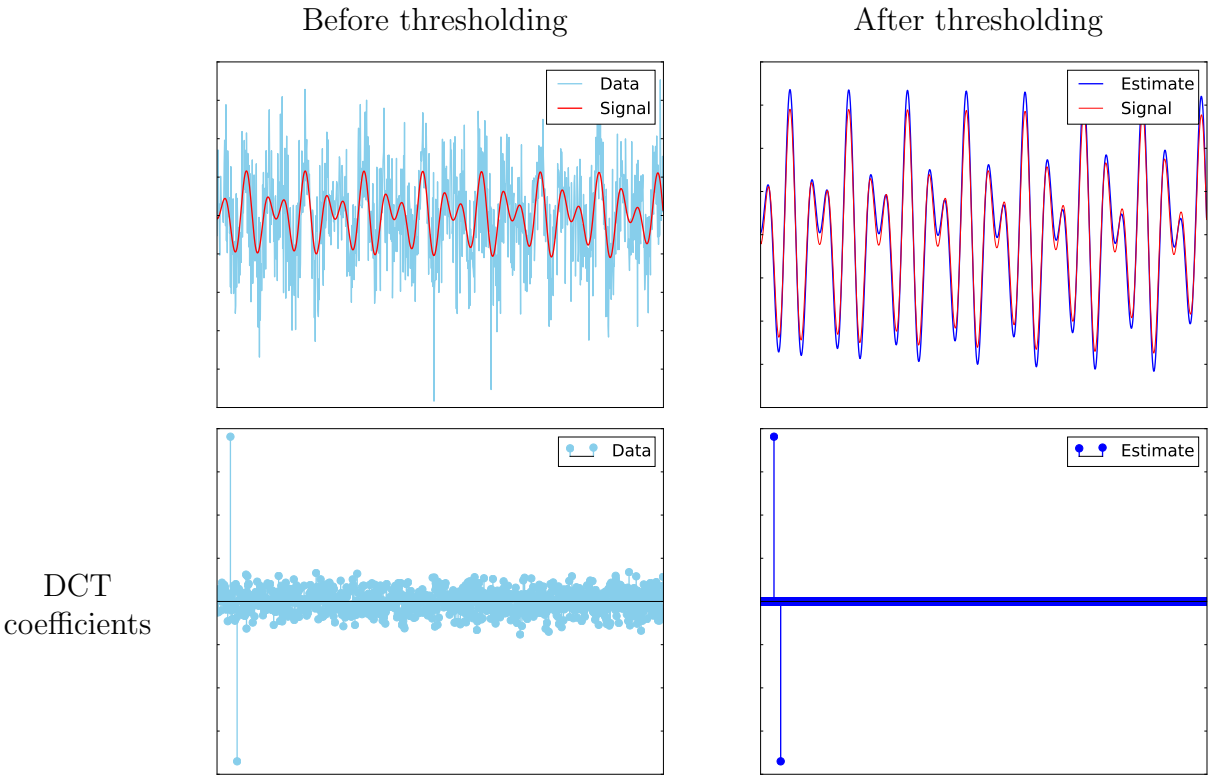


Figure 20: Denoising via hard thresholding in the DCT basis.

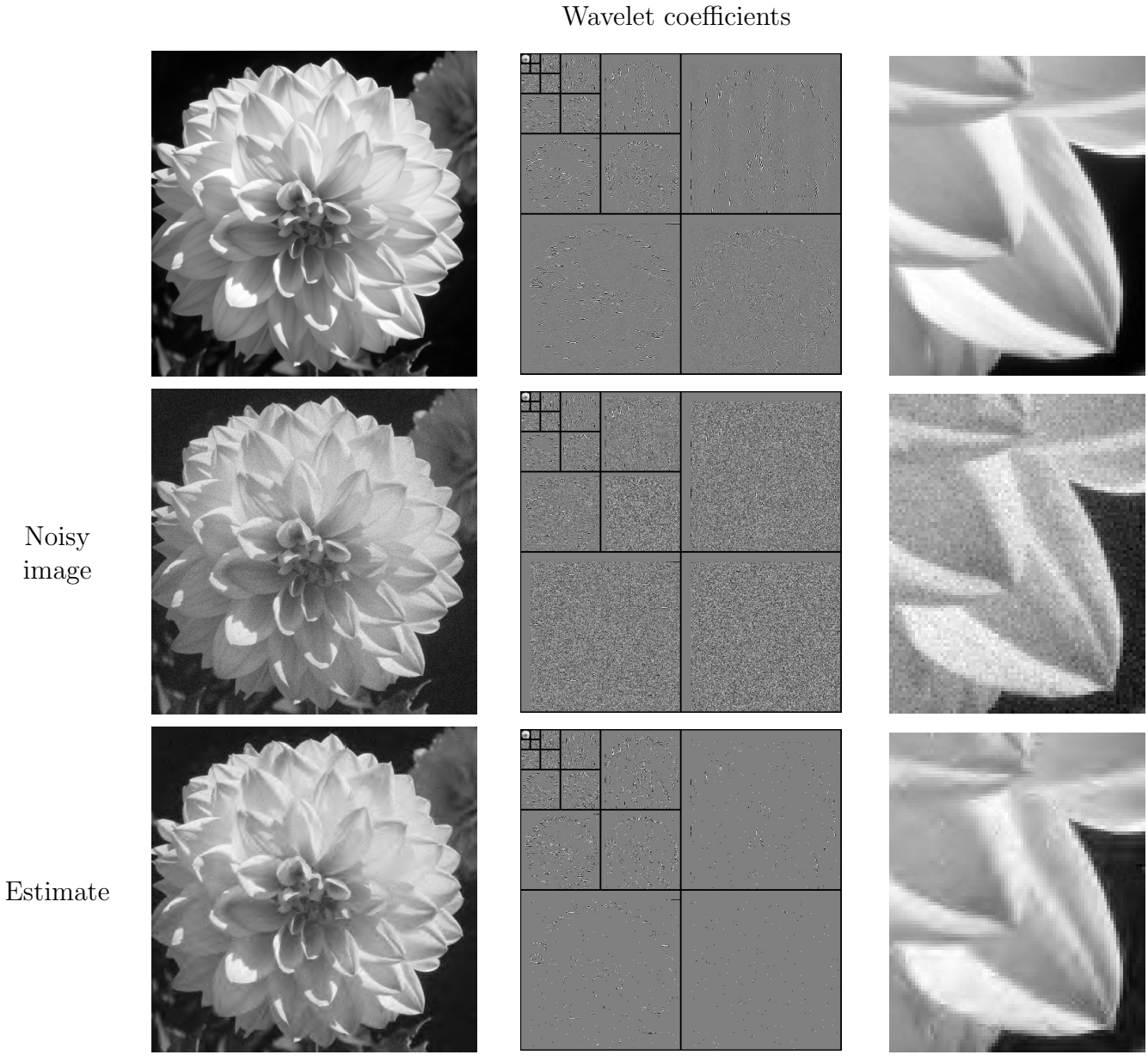


Figure 21: Denoising via hard thresholding in a biorthogonal wavelet basis.

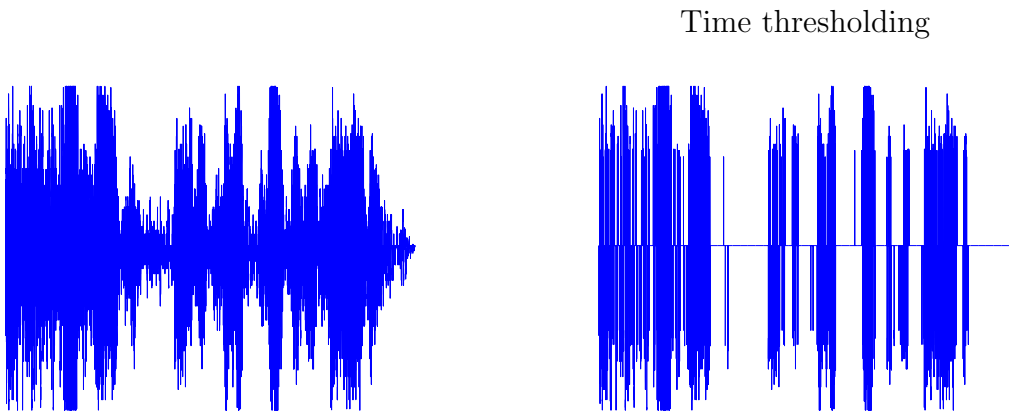


Figure 22: Time thresholding applied to the noisy data shown in Figure 6. The result sounds terrible because the thresholding eliminates parts of the speech.

4.3 Block thresholding

When we apply transforms that capture localized details of signals, such as the wavelet transform or the STFT, sparse representations tend to be highly structured. For example, nonzero wavelet coefficients are often clustered around edges. This is apparent in Figure 13. The reason is that several localized atoms are needed to reproduce sharp variations, whereas a small number of coarse-scale atoms suffice to represent smooth areas of the image.

The assumption that the coefficients of a signal are grouped together is called *group sparsity*. Thresholding-based denoising of group-sparse signals should take into account this structure. It is probably a good idea to threshold an isolated coefficient that is not too large, but a similar coefficient that lies near large nonzero coefficients is likely to contain useful information and should not be discarded.

Block thresholding exploits group sparsity by thresholding the ℓ_2 norm of groups of coefficients. The coefficients are partitioned into blocks $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k$. The blocks are then thresholded one by one.

$$\mathcal{B}_\eta\left(x\right)_i := \begin{cases} x_i & \text{if } i \in \mathcal{I}_j \quad \text{such that } \left|\left|x_{\mathcal{I}_j}\right|\right|_2 > \eta, \\ 0 & \text{otherwise.} \end{cases} \tag{73}$$

4.4 Speech denoising

We use a real speech denoising example to compare the effect of thresholding in different domains and of applying block thresholding. The recording shown in Figures 6 and 7 is a short snippet from the movie *Apocalypse Now* where one of the character talks over the noise of a helicopter. We denoise the data using the following methods (click on the links to hear

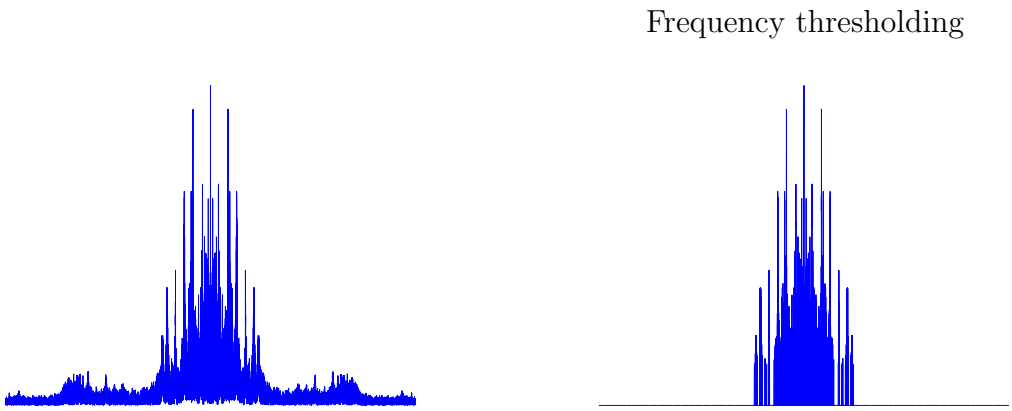


Figure 23: Frequency thresholding applied to the noisy data shown in Figure 6. The result is very low pitch because the thresholding eliminates the high frequencies of both the speech and the noise.

the result):

- Time thresholding: The result, which is plotted in Figure 22, sounds terrible because the thresholding eliminates parts of the speech.
- Frequency thresholding: The result has very low pitch because the thresholding eliminates the high frequencies of both the speech and the noise. The spectrum is shown in Figure 22 before and after thresholding.
- STFT thresholding: The result is significantly better but isolated STFT coefficients that are not discarded produce *musical noise* artifacts. The corresponding spectrogram is shown in Figure 24.
- STFT block thresholding: The result does not suffer from musical noise and retains some of the high-pitch speech. The corresponding spectrogram is shown in Figure 24.

The results are compared visually for a small time segment of the data in Figure 25.

4.5 Synthesis model

A discussed previously, overcomplete dictionaries are not invertible. This means that we cannot apply a linear transform to the noisy signal and threshold in order to exploit the synthesis model. However, we can apply the methods we studied in Section 3 for estimating synthesis coefficients in the noiseless case. In particular, we can adapt the approach based on ℓ_1-norm minimization by eliminating the equality constraint in Problem **??** and adding a

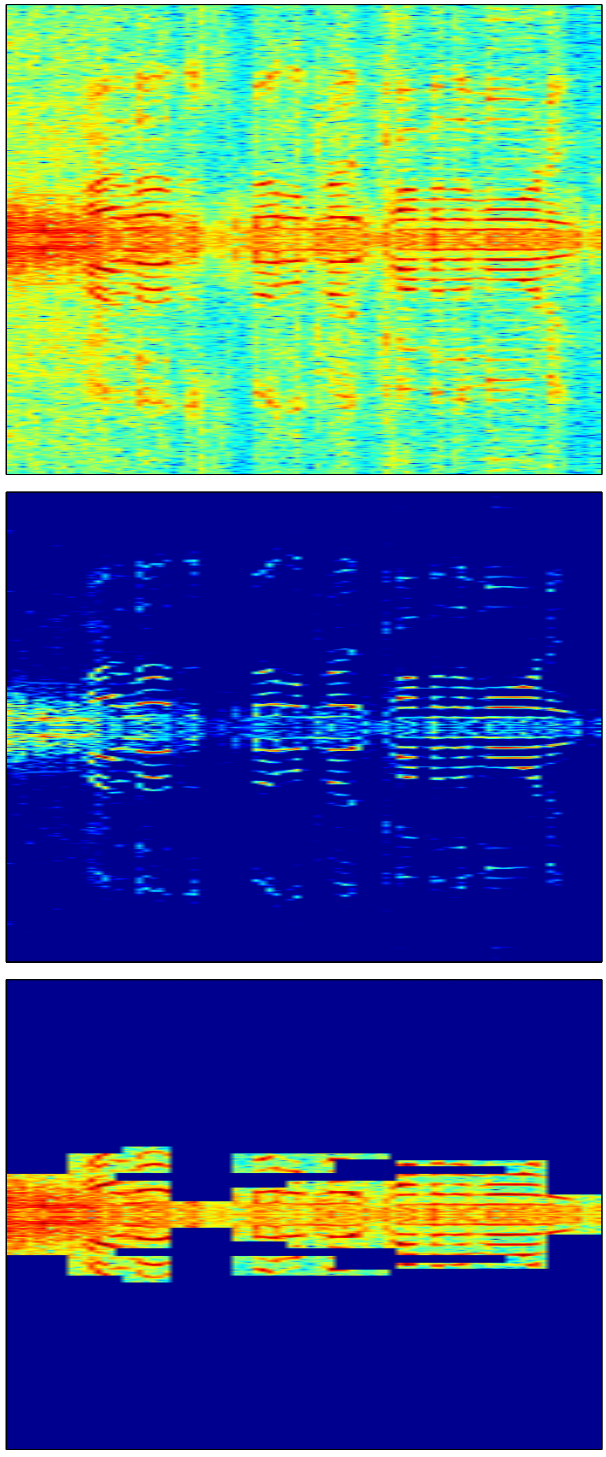


Figure 24: Spectrograms of the noisy signal (above) compared to the estimates obtained by simple thresholding (center) and block thresholding (bottom). The result of simple thresholding contains musical noise caused by particularly large STFT coefficients caused by the noise that were not thresholded. The result of block thresholding does not suffer from these artifacts.

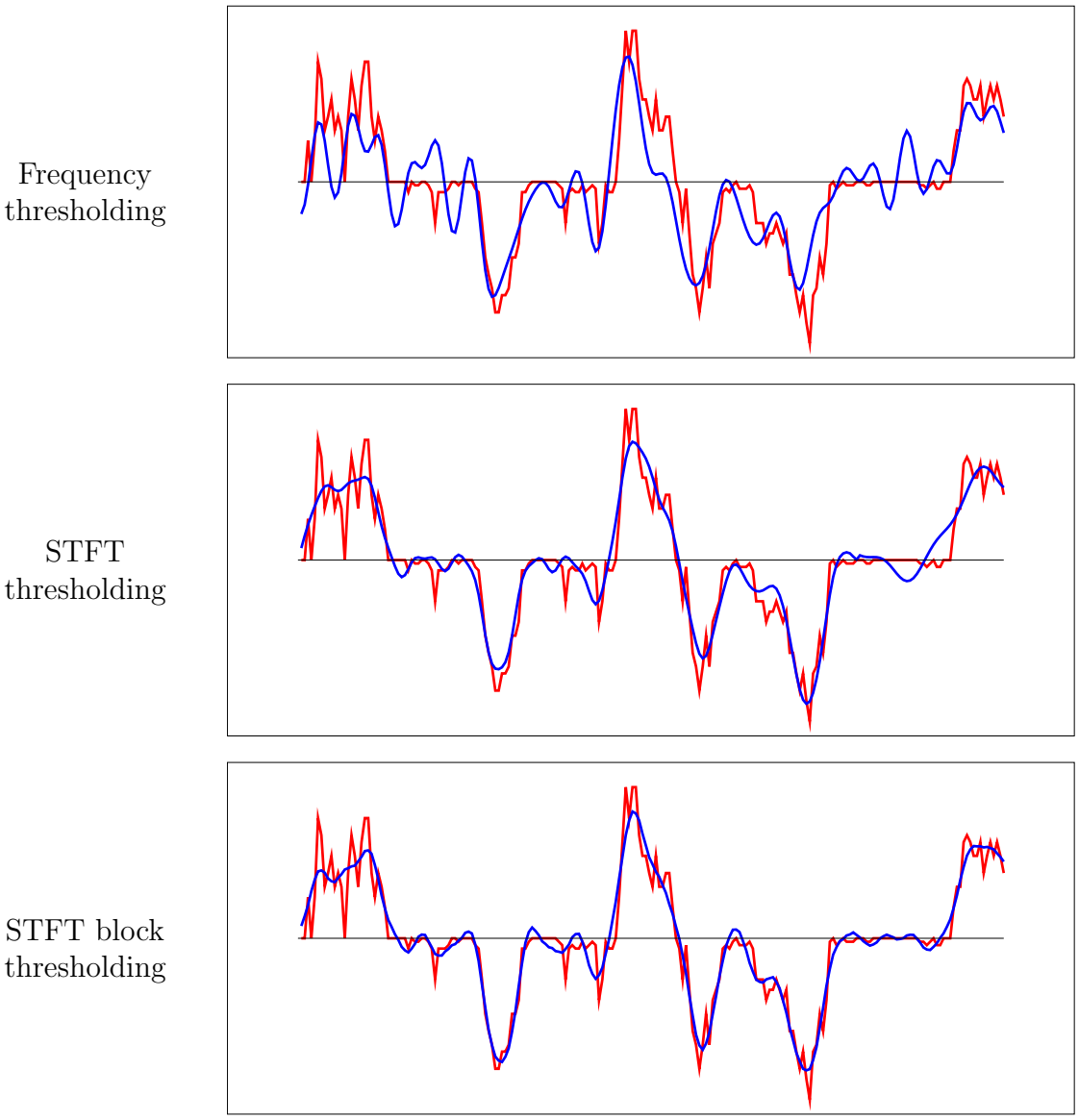


Figure 25: Comparison of the original noisy data (blue) with the denoised signal for the data shown in Figure 6. We compare frequency thresholding (above) and thresholding (center) and block thresholding (below) of STFT coefficients.

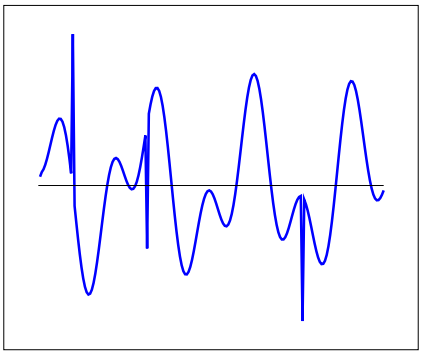


Figure 26: A signal consisting of a superposition of spikes and sinusoids.

data-fidelity term to the cost function. This is known as basis-pursuit denoising [2]. Thus, we estimate the coefficients by solving

$$\hat{c} = \arg\min_{\tilde{x} \in \mathbb{R}^m} ||y - D\tilde{c}||_2^2 + \lambda \, ||\tilde{c}||_1 \tag{74}$$

$$\hat{x} = D\hat{c}, \tag{75}$$

where $\lambda > 0$ is a regularization parameter that determines the tradeoff between the term that promotes sparsity and the term that promotes data fidelity.

The signal in Figure 26 is not sparse either in a basis of spiky atoms or sinusoidal atoms. However, it is sparse in a dictionary that contains *both* sinusoids and spikes,

$$x = Dc = \begin{bmatrix} I & F \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = a + Fb, \tag{76}$$

where $I \in \mathbb{R}^{n \times n}$ is the identity matrix and $F \in \mathbb{R}^{n \times n}$ is a DCT matrix. Figure 27 shows the result of applying ℓ_1-norm regularization to denoise a noisy version of the signal.

4.6 Analysis model

In Section 4.2 we explained how to apply thresholding to the coefficients of a noisy signal in an overcomplete dictionary. To retrieve an approximation of the signal we were forced to apply a left inverse matrix that could distort the result. A more principled way to enforce a sparse analysis model while denoising is to solve the ℓ_1-norm regularized problem

$$\hat{x} = \arg\min_{\tilde{x} \in \mathbb{R}^m} ||y - \tilde{x}||_2^2 + \lambda \left|\left|A^T \tilde{x}\right|\right|_1, \tag{77}$$

which can be interpreted as a tractable relaxation of the problem

$$\text{minimize} \quad \left|\left|D^T \tilde{x}\right|\right|_0 \tag{78}$$

$$\text{subject to} \quad y \approx \tilde{x}. \tag{79}$$

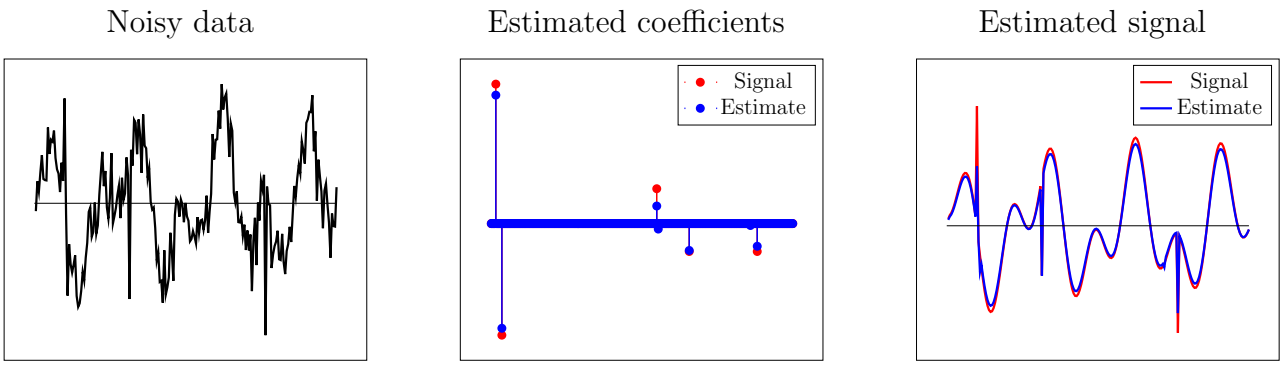


Figure 27: Denoising via ℓ_1-norm-regularized least squares.

Although this problem is very similar to the ℓ_1-norm regularized synthesis formulation (74) it is significantly more challenging to solve.

In image processing, an extremely popular analysis operator is the finite-differences operator. The reason is that images are often well approximated as piecewise constant, which means that they have sparse gradients. We define the total variation of an image Im as the ℓ_1-norm of the horizontal and vertical components of its gradient

$$\text{TV}\left(\text{Im}\right) := \left|\left|\nabla_x \, \text{Im}\right|\right|_1 + \left|\left|\nabla_y \, \text{Im}\right|\right|_1 . \tag{80}$$

If the image is corrupted by noise that is not piecewise constant, we can enforce the prior that the gradient is sparse by penalizing the total variation of the estimate. This is equivalent to applying ℓ_1-norm regularization with an analysis operator that computes the discretized gradient of the image (i.e. a finite-differences operator),

$$\widehat{\text{Im}} = \arg \min_{\widetilde{\text{Im}} \in \mathbb{R}^{n \times n}} \left|\left|Y - \widetilde{\text{Im}}\right|\right|_{\text{F}}^2 + \lambda \, \text{TV}\left(\widetilde{\text{Im}}\right) . \tag{81}$$

Figures 29, 30 and **??** display the results of applying total-variation denoising to a one-dimensional piecewise constant signal and to a real image. Small values of the regularization parameter do not denoise well, whereas large values produce cartoonish estimates. Medium values however allow to denoise quite effectively. We refer the interested reader to [1, 6] for more details on total-variation regularization.

References

The book *A wavelet tour of signal processing* by Mallat [3] is a great reference for the topics discussed in these notes. Numerical experiments by Gabriel Peyré illustrating many of the ideas that we have discussed are available here.

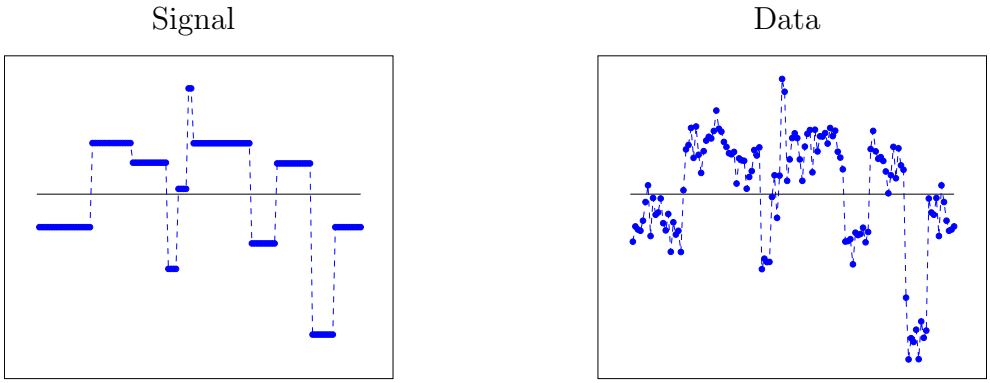


Figure 28: One-dimensional signal (left) and the corresponding noisy data (right).

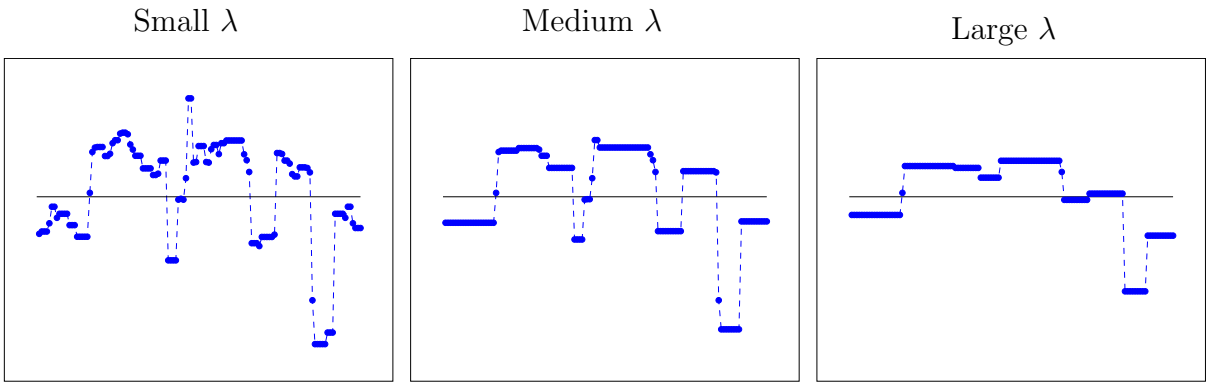


Figure 29: Total-variation denoising applied to the data in Figure 28 for different values of the regularization parameter.

Figure 30: Total-variation denoising applied to the image in Figure 21 for different values of the regularization parameter.

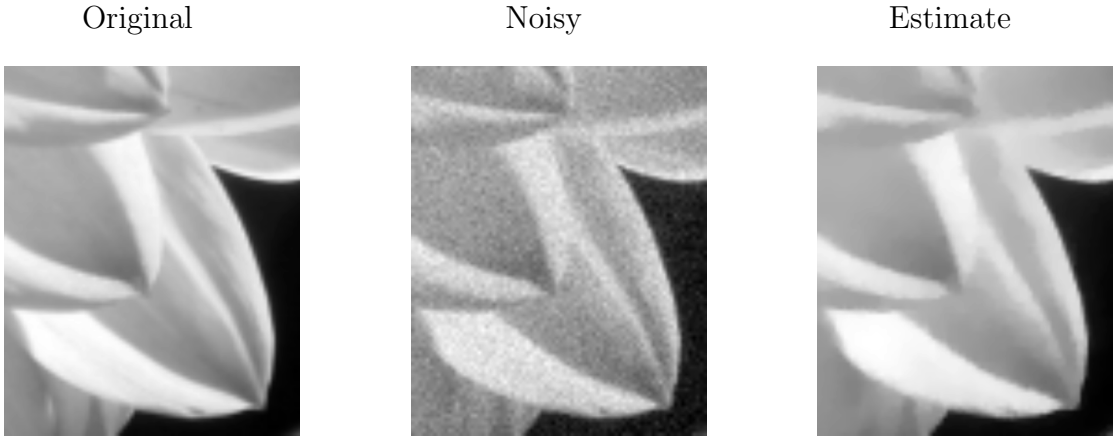


Figure 31: Total-variation denoising applied to the image in Figure 21 for different values of the regularization parameter.

[1] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.

[2] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.

[3] S. Mallat. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.

[4] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[5] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44. IEEE, 1993.

[6] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

A Proofs

A.1 Proof of Lemma 3.1

Consider the SVD of $D = USV^*$. The columns of $V \in \mathbb{R}^{m \times n}$ are an orthonormal basis for the row space of D row (D). Let us decompose any coefficient vector c' such that $x = Dc'$ as

$$c' = Vb + \mathcal{P}_{\text{row}(D)^\perp}\ (c') \tag{82}$$

where b is an n-dimensional vector. If $x = Dc'$, then

$$\begin{aligned} S^{-1}U^T x &= V^* c' && (83)\\ &= V^* \left(Vb + \mathcal{P}_{\text{row}(D)^\perp}\ (c') \right) && (84)\\ &= b, && (85) \end{aligned}$$

so b has the same value for any c' such that $x = Dc'$. We can decompose the norm of c' in the following way by Pythagoras's Theorem,

$$\begin{aligned} ||c'||_2^2 &= \left|\left|\mathcal{P}_{\text{row}(D)}\ (c')\right|\right|_2^2 + \left|\left|\mathcal{P}_{\text{row}(D)^\perp}\ (c')\right|\right|_2^2 && (86)\\ &= ||b||_2^2 + \mathcal{P}_{\text{row}(D)^\perp}\ (c')\,. && (87) \end{aligned}$$

Any solution such that $\mathcal{P}_{\text{row}(D)^\perp}\ (c') \neq 0$ will have norm greater than $||b||_2^2$ so the minimum norm solution is

$$\begin{aligned} Vb &= VS^{-1}U^T x && (88)\\ & D^T \left(DD^T\right)^{-1} x. && (89) \end{aligned}$$

Random projections

1 Introduction

Random projections are a useful tool in the analysis and processing of high-dimensional data. We will analyze two applications that use random projections to compress information embedded in high-dimensional spaces: dimensionality reduction and compressed sensing.

2 Dimensionality reduction

The goal of dimensionality-reduction techniques is to project high-dimensional data onto a lower-dimensional space while preserving as much information as possible. In these notes we will denote the dimension of the high-dimensional space by n and the dimension of the low-dimensional subspace by m. These methods are a basic tool in data analysis; some applications include visualization (especially if we project onto $\mathbb{R}^2$ or $\mathbb{R}^3$), denoising and decreasing the computational cost of processing the data. Indeed, the complexity of many algorithms depends directly on the ambient dimension and in big-data regimes applying even very simple machine-learning algorithms such as nearest-neighbour classification or least-squares regression may have a huge computational cost.

We will focus on dimensionality reduction via linear projections.

Definition 2.1 (Linear projection)**.** *The linear projection of $x \in \mathbb{R}^n$ onto a subspace $\mathcal{S} \subseteq \mathbb{R}^n$ is the point of $\mathcal{S}$ that is closest to x, i.e. the solution to the optimization problem*

$$\begin{aligned} &\text{minimize} && ||x - u||_2 && (1)\\ &\text{subject to} && u \in \mathcal{S}. && (2) \end{aligned}$$

The following simple lemma explains how to compute a projection using an orthonormal basis of the subspace that we want to project onto. The proof is in Section A.1 of the appendix.

Lemma 2.2. *Let U be a matrix whose columns are an orthonormal basis of a subspace $\mathcal{S} \subseteq \mathbb{R}^n$.*

$$\mathcal{P}_{\mathcal{S}}\,(x) = UU^T x. \tag{3}$$

Once we fix a projection and a corresponding matrix U the lower-dimensional representation of a vector $x \in \mathbb{R}^n$ is $U^T x \in \mathbb{R}^m$, so the dimensionality is reduced from n to m. An interesting problem is how to choose the low-dimensional subspace parametrized by U. The following sections describe two popular alternatives: principal component analysis and random projections.

2.1 Principal component analysis

Principal component analysis (PCA) is an adaptive dimensionality-reduction technique in which we first determine the directions of maximum variation in a dataset and then project onto them. PCA is based on the singular-value decomposition (SVD). The proof of the following fundamental result can be found in any graduate linear algebra textbook.

Theorem 2.3. *Without loss of generality let $m \leq n$. Every rank r real matrix $A \in R^{m \times n}$ has a unique singular-value decomposition of the form (SVD)*

$$A = \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \sigma_m \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \cdots \\ v_m^T \end{bmatrix} \tag{4}$$

$$= USV^T, \tag{5}$$

where the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$ are nonnegative real numbers, the matrix $U \in R^{m \times m}$ containing the left singular vectors is orthogonal, and the matrix $V \in R^{m \times n}$ containing the right singular vectors is a submatrix of an orthogonal matrix (i.e. its columns form an orthonormal set).

Algorithm 2.4 (Principal component analysis)**.** *Given k data vectors $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_k \in \mathbb{R}^n$, we apply the following steps.*

1. *Center the data,*

$$x_i = \tilde{x}_i - \frac{1}{k} \sum_{i=1}^{k} \tilde{x}_i, \qquad 1 \leq i \leq n. \tag{6}$$

2. *Group the centered data in a data matrix $X \in \mathbb{R}^{n \times k}$*

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_k \end{bmatrix}. \tag{7}$$

3. *Compute the SVD of X and extract the left singular vectors corresponding to the m largest singular values. These are the first m principal components.*

$\sigma_1/\sqrt{n} = 0.705,$
$\sigma_2/\sqrt{n} = 0.690$

$\sigma_1/\sqrt{n} = 0.9832,$
$\sigma_2/\sqrt{n} = 0.3559$

$\sigma_1/\sqrt{n} = 1.3490,$
$\sigma_2/\sqrt{n} = 0.1438$

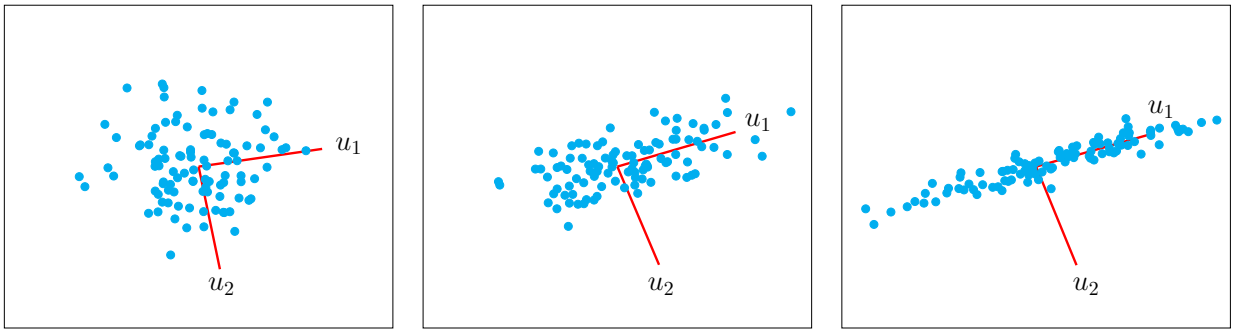


Figure 1: PCA of a dataset with $n = 100$ 2D vectors with different configurations. The two first singular values reflect how much energy is preserved by projecting onto the two first principal components.

Once the data are centered, the energy of their projection onto different directions in the ambient space reflects the variation of the dataset along those directions. PCA selects the directions that maximize the ℓ_2 norm of the projection and are mutually orthogonal. The span of the first m principal components is the subspace the best approximates the data in terms of ℓ_2-norm error, as established by the following theorem proved in Section A.2 of the appendix.

Theorem 2.5. *For any matrix $X \in \mathbb{R}^{n \times k}$ with left singular vectors $u_1, u_2, \ldots, u_n$ corresponding to the nonzero singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$,*

$$\sum_{i=1}^{k} \left|\left|\mathcal{P}_{\text{span}(u_1, u_2, \ldots, u_n)}\, x_i\right|\right|_2^2 \geq \sum_{i=1}^{k} ||\mathcal{P}_{\mathcal{S}}\, x_i||_2^2\,, \tag{8}$$

for any subspace $\mathcal{S}$ of dimension $m \leq \min\{n, k\}$.

Figure 1 illustrates PCA in 2D. Note how the singular values are proportional to the energy that lies in the direction of the corresponding principal component.

Figure 2 illustrates the importance of centering before applying PCA. Theorem 2.5 still holds if the data are not centered. However, the norm of the projection onto a certain direction no longer reflects the variation of the data. In fact, if the data are concentrated around a point that is far from the origin, the first principal component will tend be aligned in that direction. This makes sense as projecting onto that direction captures more energy. As a result, the principal components do not capture the directions of maximum variation *within* the cloud of data.

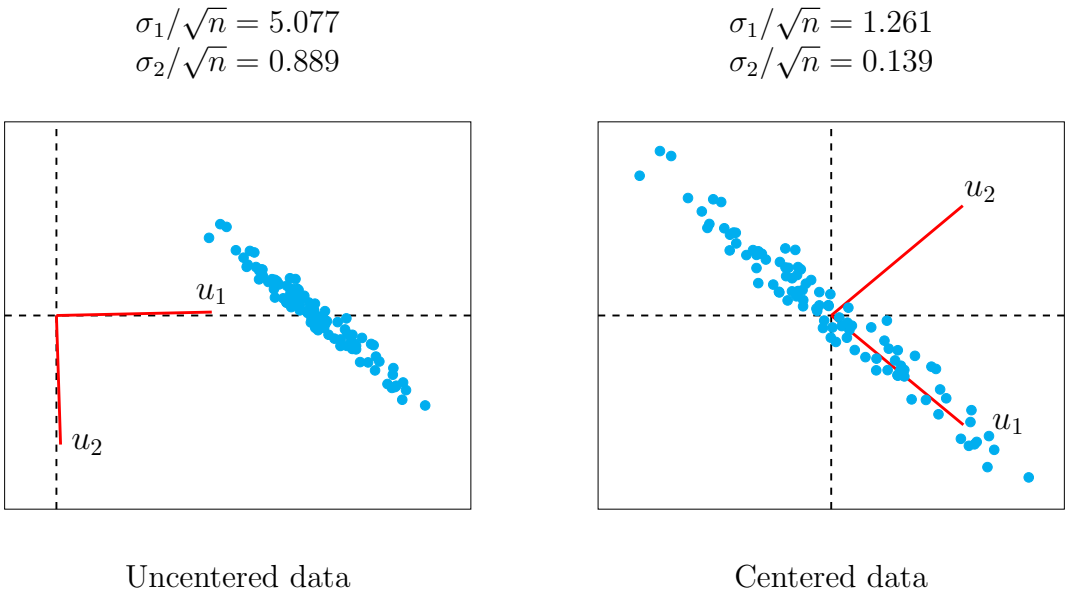


Figure 2: PCA applied to $n = 100$ 2D data points. On the left the data are not centered. As a result the dominant principal component u_1 lies in the direction of the mean of the data and PCA does not reflect the actual structure. Once we center, u_1 becomes aligned with the direction of maximal variation.

The file *seeds_dataset.txt* contains different geometric attributes (area, perimeter, compactness, length of kernel, width of kernel, asymmetry coefficient and length of kernel groove) of seeds belonging to three different varieties of wheat: Kama, Rosa and Canadian[1]. Figure 3 shows the projection of the data onto the first two and the last two principal components. The structure of the data is much better conserved in the first case, which allows to visualize the difference between the three seeds very clearly. Note however that the first principal components only guarantee that the energy in the projection will be preserved, not that the projection will be good for tasks such as classification.

2.2 Random projections

To apply PCA we need to process all of the data points beforehand in order to compute the projection. This may be too computationally costly if the dataset is very large or not possible at all if the aim is to project a stream of data in real time. For such cases we need a *non-adaptive* alternative to PCA that chooses the projection before actually seeing the data.

A simple method that tends to work well is to project onto a random subspace. In particular, if the data points are $x_1, x_2, \ldots \in \mathbb{R}^n$, we can obtain a random projection by multiplying the

[1]The data can be found at `https://archive.ics.uci.edu/ml/datasets/seeds`

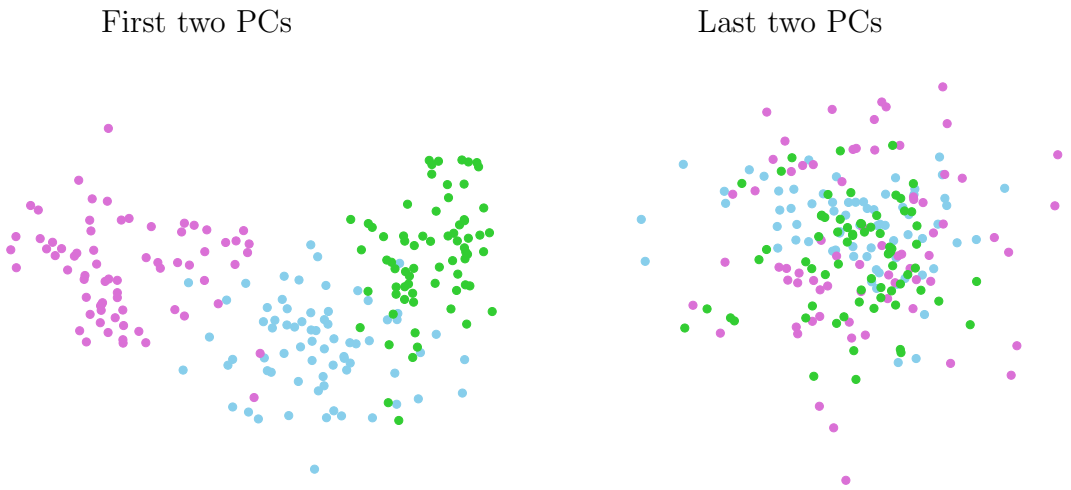


Figure 3: Projection of 7-dimensional vectors describing different wheat seeds onto the first two (left) and the last two (right) principal components of the dataset. Each color represents a variety of wheat.

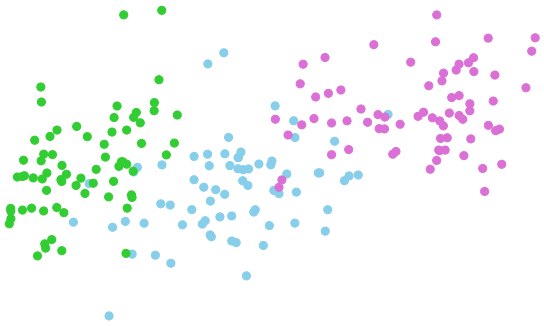


Figure 4: Approximate projection of 7-dimensional vectors describing different wheat seeds onto two random directions. Each color represents a variety of wheat.

data with a random matrix $A \in \mathbb{R}^{m \times n}$ to obtain $Ax_1, Ax_2, \ldots \in \mathbb{R}^m$. Strictly speaking, this is a linear projection only if A is a projection matrix with orthonormal rows (U^T in Lemma 2.2). However, in many cases the rows of random matrices are approximately orthogonal and consequently this procedure yields an approximate projection.

Figure 4 shows the result of applying such an approximate projection to the same data used in Figure 3 onto two random directions by multiplying the data with a 2×7 random Gaussian matrix. The structure of the data seems to be as well conserved by the projection as in the case of PCA.

Dimensionality-reduction techniques are useful if they preserve the information that we are interested in. In many cases, we would like the projection to conserve the distances between the different data points. This allows to apply algorithms such as nearest neighbors in the lower-dimensional space. The following lemma guarantees that random projections do not distort the distances between points with a certain probability. The result is striking because the lower bound on m– the dimension of the approximate projection– does not depend on n– the ambient dimension of the data– and its dependence on the number of points in the dataset is only logarithmic. Although we prove the result for a matrix with Gaussian entries, the result can be extended to other matrices that can be applied more efficiently [1].

Lemma 2.6 (Johnson-Lindenstrauss lemma)**.** *Let $\mathcal{S} := \{x_1, \ldots, x_k\}$ in $\mathbb{R}^n$. There exists a random function f such that for any pair of points x_i, x_j*

$$(1 - \epsilon) \left|\left|x_i - x_j\right|\right|_2^2 \leq \left|\left|f\left(x_i\right) - f\left(x_j\right)\right|\right|_2^2 \leq (1 + \epsilon) \left|\left|x_i - x_j\right|\right|_2^2, \tag{9}$$

with probability at least $\frac{1}{k}$ as long as

$$m \geq \frac{8 \log\left(k\right)}{\epsilon^2}. \tag{10}$$

The random function is of the form

$$f\left(x\right) := \frac{1}{\sqrt{m}} A x \tag{11}$$

and A is a matrix with iid Gaussian entries with zero mean and unit variance.

Proof. To establish the result we use the following proposition, proved in Section A.4 of the appendix, which establishes that the norm of a fixed vector is preserved by the random projection with high probability.

Proposition 2.7. *Let f be defined as in Lemma 2.6. For any fixed vector $v \in \mathbb{R}^n$*

$$\mathrm{P}\left((1 - \epsilon) \left|\left|v\right|\right|_2^2 \leq \left|\left|f\left(v\right)\right|\right|_2^2 \leq (1 + \epsilon) \left|\left|v\right|\right|_2^2\right) \geq 1 - 2 \exp\left(-\frac{m \epsilon^2}{8}\right). \tag{12}$$

We define the events

$$\mathcal{E}_{ij} = \left\{ (1-\epsilon) \left|\left|x_i - x_j\right|\right|_2^2 \leq \left|\left|f\left(x_i - x_j\right)\right|\right|_2^2 \leq (1+\epsilon) \left|\left|x_i - x_j\right|\right|_2^2 \right\}, \quad i \neq j, 1 \leq i, j \leq k.$$

By Proposition 2.7 applied to $v := x_i - x_j$ and condition (10)

$$\mathrm{P}\left(\mathcal{E}_{ij}^c\right) \leq \frac{2}{k^2}. \tag{13}$$

There are $\binom{k}{2}$ different pairs of points. The union bound yields

$$\begin{aligned}
\mathrm{P}\left(\bigcap_{i,j} \mathcal{E}_{ij}\right) &= 1 - \mathrm{P}\left(\bigcup_{i,j} \mathcal{E}_{ij}^c\right) && (14)\\
&\geq 1 - \sum_{i,j} \mathrm{P}\left(\mathcal{E}_{ij}^c\right) && (15)\\
&\geq 1 - \binom{k}{2} \frac{2}{k^2} && (16)\\
&\geq \frac{1}{k} && (17)
\end{aligned}$$

where the last inequality follows from condition (10). This completes the proof. □

3 Compressed sensing

Compressed sensing allows to recover sparse signals from randomized measurements by minimizing their ℓ_1 norm. In this section we first illustrate the application of these ideas to MRI and then provide a theoretical analysis.

3.1 Compressed sensing in MRI

Magnetic resonance imaging (MRI) is a popular medical imaging technique used in radiology. MRI data can be modeled as samples from the 2D or 3D Fourier transform of the object that is being imaged, for example a slice of a human brain. An estimate of the corresponding image can be obtained by computing the inverse Fourier transform of the data, as shown in Figure 5.

An important challenge in MRI is to reduce measurement time, which can be achieved by undersampling the data. Consider a 1D version of the problem, where the signal is an n-dimensional vector x and the DFT matrix is denoted by F. We model the undersampled data y as

$$y = F_\Omega \, x, \tag{18}$$

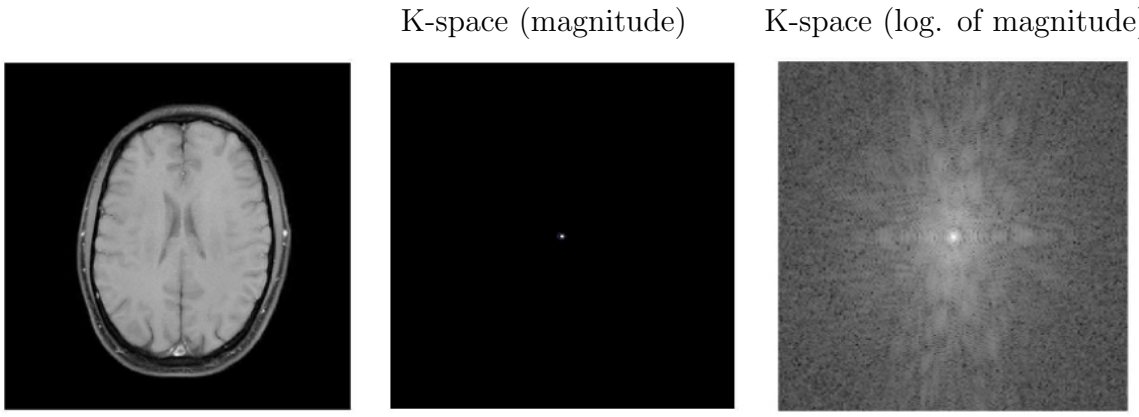


Figure 5: Image of a brain obtained by MRI, along with the magnitude of its 2D-Fourier or k-space representation and the logarithm of this magnitude.

where F_Ω is a submatrix of F obtained by choosing m rows indexed by the set Ω. Since $m < n$ the system is underdetermined and has infinite solutions.

A possible estimate for the image is the solution x_{ℓ_2} with minimum ℓ_2 norm satisfying $y = F_\Omega \, x_{\ell_2}$. By Lemma 3.1 in Lecture Notes 4 this estimate equals

$$x_{\ell_2} := F_\Omega^T \left(F_\Omega F_\Omega^T \right)^{-1} x, \tag{19}$$

where x is the original signal. Equivalently, x_{ℓ_2} is the projection of x onto the row space of F_Ω. This projection provides some insight as to the effect of different sampling strategies. Figure 6 shows x_{ℓ_2} for two undersampling patterns in 2D: regular undersampling in one direction and random undersampling. The corresponding artifacts are very different. Regular undersampling produces coherent aliasing– the reconstruction is a superposition of shifted copies of the image– whereas random undersampling produces aliasing that essentially looks like noise.

MRI signals are sparse in different transform domains. For example, the brain image in Figure 5 is sparse in the wavelet domain. This is good news if we are trying to recover an image from undersampled data. We cannot estimate more than m parameters from m measurements, so in principle it is hopeless to try to estimate an n-dimensional signal from data given by 18. However, if the signal is sparse in some domain and can be parametrized by s coefficients for instance, where $s < m$, then recovery may be possible.

As we discussed in the previous lecture, ℓ_1-norm minimization is an effective tool for obtaining sparse solutions to underdetermined linear equations. Figure 7 shows the result of applying ℓ_1-norm minimization to recover an image from the data corresponding to the images shown

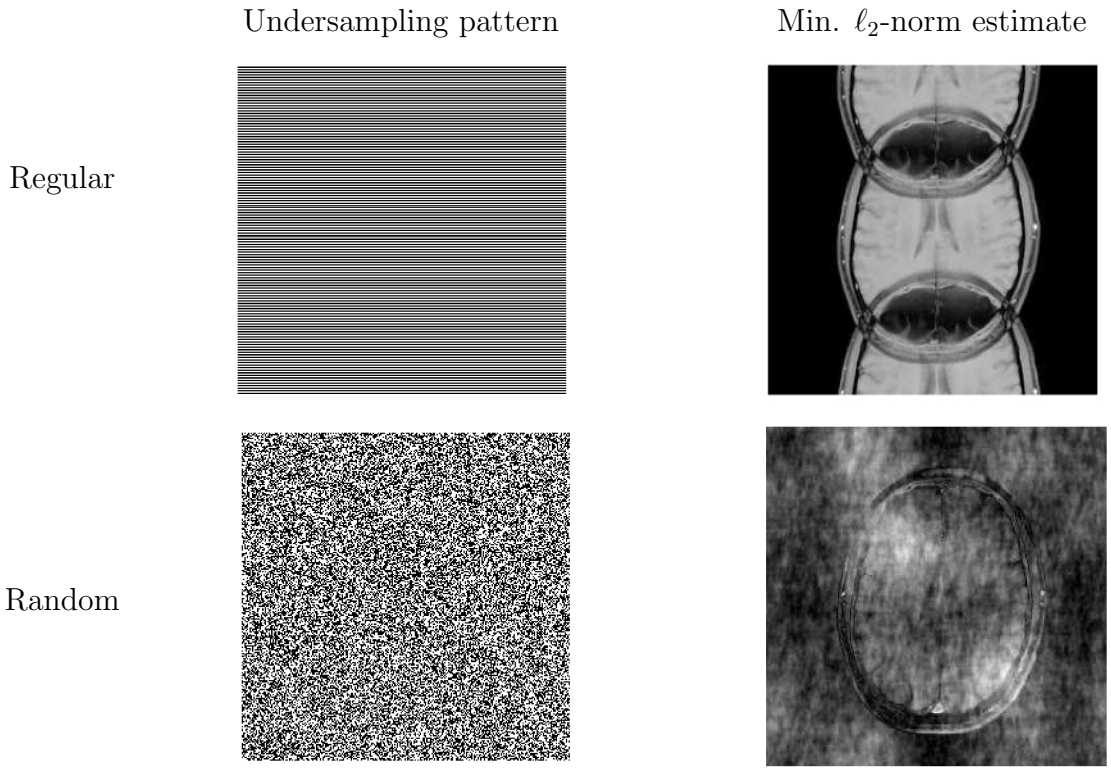


Figure 6: Two different sampling strategies in 2D k space: regular undersampling in one direction (top) and random undersampling (bottom). The original data is the same as in Figure 5. On the right we see the corresponding minimum-ℓ_2-norm estimate for each undersampling pattern.

Figure 7: Minimum-ℓ_1-norm for the two undersampling patterns shown in Figure 6.

in Figure 6. Since the image is assumed to be sparse in the wavelet domain, we solve

$$\begin{aligned} &\text{minimize} && ||\tilde{c}||_1 && (20)\\ &\text{subject to} && y = F_\Omega W \tilde{c} && (21) \end{aligned}$$

where W is the matrix corresponding to an orthonormal wavelet transform. For regular undersampling, then the estimate is essentially the same as the minimum-ℓ_2-norm estimate. This is not surprising, since the minimum-ℓ_2-norm estimate is also sparse in the wavelet domain because it is equal to a superposition of two shifted copies of the image. In contrast, ℓ_1-norm minimization recovers the original image perfectly when coupled with random projections. Intuitively, ℓ_1-norm minimization *cleans up* the noisy aliasing caused by random undersampling. In the next section we provide a theoretical characterization of this phenomenon.

3.2 Exact recovery

In our theoretical analysis we will focus on one-dimensional sparse signals, but most of the ideas extend to the case where the signal is multidimensional or sparse in a transform domain instead. The measurement model is given by

$$y = Ax \tag{22}$$

where $x \in \mathbb{R}^n$ is the sparse signal which has s nonzeros, $y \in \mathbb{R}^m$ the data and $A \in \mathbb{R}^{m \times n}$ the random matrix that models the measurement process. In MRI the rows of A are chosen at random from a DFT matrix. Here we will mostly assume that the entries are sampled independently at random from a Gaussian distribution. This model is not of great practical interest because Gaussian measurements do not arise in applications and multiplication with dense Gaussian matrices is computationally expensive. However, the Gaussian assumption makes the proofs significantly simpler and allows to illustrate ideas that are readily applicable to more useful cases such as random Fourier measurements.

Clearly, without any assumptions on x the underdetermined linear system (22) cannot be solved to retrieve x if $m < n$. It is necessary to use the sparsity assumption. If $s < n$ and the sparsity pattern T (the indices of the entries of x that are nonzero) is known, then the problem is actually overdetermined. Indeed, let x_T be the subvector of nonzero entries of x and $A_T \in \mathbb{R}^{m \times s}$ the submatrix of A consisting of the columns indexed by T. We have

$$y = Ax = A_T x_T. \tag{23}$$

As long as the columns of A_T are independent and $m \geq s$ we can easily find the original vector by applying any left inverse of A_T to the data. In particular, choosing the pseudoinverse,

$$A^\dagger y = \left(A_T^T A_T\right)^{-1} A_T^T y = x. \tag{24}$$

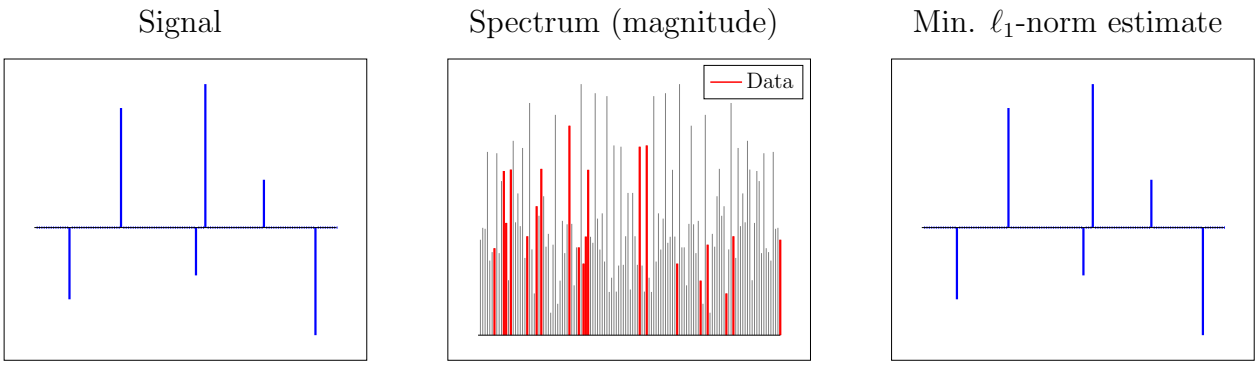


Figure 8: Minimizing the ℓ_1 norm of the estimate (right) allows to estimate the original signal (left) exactly from a small number of random samples of its spectrum (center).

The bad news is that we don't know T! Trying every possible submatrix of s columns is not feasible computationally for even very small values of s. Instead, we apply ℓ_1 norm minimization to favor sparse estimates. We recover x by solving the convex program

$$\begin{aligned} &\text{minimize} \quad ||\tilde{x}||_1 &(25)\\ &\text{subject to} \quad y = A\tilde{x}. &(26)\end{aligned}$$

Figure 8 shows an example where this procedure allows to reconstruct a sparse signal from random Fourier data. The following theorem establishes that ℓ_1-norm minimization achieves exact recovery of x with high probability as long as the number of measurements m is proportional to the sparsity of the signal s up to logarithmic factors. The result is remarkable. Random measurements allow to sample the signal at a rate that is essentially independent of the ambient dimension and only depends on how compressible it is.

Theorem 3.1. *Assume there exists a signal $x \in \mathbb{R}^n$ with s nonzeros such that*

$$Ax = y \tag{27}$$

for a random matrix $A \in \mathbb{R}^{m\times n}$ with iid Gaussian entries with zero mean and unit variance. The solution to Problem (106) is equal to x with probability at least $1 - \frac{1}{n}$ as long as the number of measurements satisfies

$$m \geq Cs \log n, \tag{28}$$

for a fixed numerical constant C.

An important generalization of this result establishes that A can be formed by taking random rows from any unitary matrix $U \in \mathbb{C}^{n\times n}$ (unitary is a fancy word for a matrix with orthonormal columns). In that case, the number of measurements that are necessary for exact recovery also depends on the coherence μ of the measurements, defined as

$$\mu(U) := \sqrt{n} \max_{1\leq i\leq n, 1\leq j\leq m} |U_i j|. \tag{29}$$

Intuitively the coherence measures how localized or *spiky* the rows of U are. If the rows are too localized, they might miss entries in the sparse vector. Recall that we don't know the support T beforehand, otherwise we would definitely use measurements that are localized on T! Exact recovery via ℓ_1-norm minimization is achieved with high probability for this kind of measurements if

$$m \geq C\mu\left(U\right) s \log n. \tag{30}$$

We refer to [6] for a proof of this result (see also [4]). In the case of the DFT $\mu = 1$ since the rows are complex exponentials divided by $\sqrt{n}$.

To prove Theorem 3.1 we use the following result, which establishes that the existence of a certain vector implies exact recovery. We defer the proof to Section A.6 in the appendix.

Lemma 3.2. *Let T be the indices of the nonzero entries in x. If A_T is full rank and there exists a vector $v \in \mathbb{R}^m$ such that*

$$\left(A^T v\right)_i = \operatorname{sign}\left(x_i\right) \qquad \text{if } x_i \neq 0 \tag{31}$$

$$\left|\left|\left(A^T v\right)_i\right|\right|_\infty < 1 \qquad \text{if } x_i = 0 \tag{32}$$

then x is the unique solution to Problem (106)*.*

By this lemma, all we need to do to establish exact recovery is show that for any subset T with cardinality s there exists a vector v such that $q := A^T v$ is equal to the sign of x on T and has magnitude strictly bounded by one on T^{c}. v is commonly known as a *dual certificate* in the literature. The reason is that it certifies optimality and is feasible for the dual problem of Problem (106), derived in Section A.7 of the appendix.

Lemma 3.3 (Dual problem)**.** *The dual problem to Problem* (106) *is*

$$\text{maximize} \quad y^T \tilde{v} \tag{33}$$

$$\text{subject to} \quad \left|\left|A^T \tilde{v}\right|\right|_\infty \leq 1, \tag{34}$$

where the dual variable $\tilde{v}$ has dimension n.

The dual certificate is feasible for this problem and achieves a cost-function value of $\left|\left|x\right|\right|_1$, since

$$y^T v = x^T A^T v \tag{35}$$

$$= \sum_{i \in T} x_i \operatorname{sign}\left(x_i\right) \tag{36}$$

$$= \left|\left|x\right|\right|_1. \tag{37}$$

By weak duality (Corollary 4.8 in Lecture Notes 2) this implies that x is a solution to the primal problem (note that this does not prove that it is the unique solution but Lemma 3.2 does). In the next section we will construct the dual certificate that establishes Theorem 3.1.

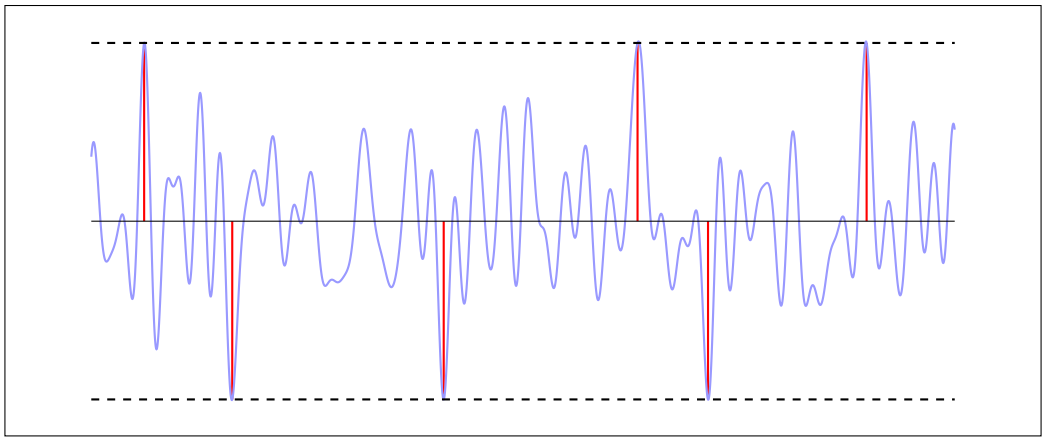


Figure 9: Subgradient $q_{\ell_2} := A^T v_{\ell_2}$ (blue) corresponding to the minimum-ℓ_2-norm dual certificate v_{ℓ_2} for a compressed-sensing problem with random Fourier measurements. q_{ℓ_2} interpolates the sign of the signal (red) on its support.

3.3 Dual certificate

In this section we analyze a dual-certificate candidate and show that it satisfies the conditions in Lemma 3.2 and hence proves Theorem 3.1 with high probability. In particular, we set v_{ℓ_2} to be the solution to the following minimization problem

$$\begin{aligned} &\text{minimize} && ||\tilde{v}||_2 && (38)\\ &\text{subject to} && A_T^T \tilde{v} = \text{sign}\left(x_T\right). && (39) \end{aligned}$$

In words, v_{ℓ_2} is the minimum-ℓ_2-norm vector such that $q_{\ell_2} := A^T v_{\ell_2}$ interpolates the sign of x on T. Ideally we would like to minimize the ℓ_∞ norm of the subgradient $A^T v$ on T^c instead of its ℓ_2 norm, but the former problem does not have a closed-form solution. From Lemma 3.1 in Lecture Notes 4 the solution to Problem (38) is

$$v_{\ell_2} = A_T \left(A_T^T A_T\right)^{-1} \text{sign}\left(x_T\right) \tag{40}$$

as long as A_T is full rank. Having an explicit expression for the solution allows us to analyze the construction for any arbitrary support and sign pattern. We will do this under the assumption that the entries of the measurement matrix A are Gaussian, but this technique has also been applied to Fourier [7] and other random measurements [4] (see also [6] for a more recent proof technique based on approximate dual certificates that provides better guarantees). Figure 9 shows q_{ℓ_2} for the case of random Fourier measurements.

To characterize v_{ℓ_2} first we need to establish that A_T is full rank, since otherwise $A_T^T A_T$ would not have an inverse. The following proposition shows that the random matrix preserves the norm of sparse vectors supported on a fixed set T with a certain probability. We defer the proof to Section 3.4.

Proposition 3.4. *Fix a set $T \subset \{1, 2, \ldots, n\}$ such that $|T| \leq s$. For any unit-norm vector x with support T*

$$1 - \epsilon \leq \frac{1}{\sqrt{m}} \left|\left|Ax\right|\right|_2 \leq 1 + \epsilon \tag{41}$$

with probability at least

$$1 - 2\left(\frac{12}{\epsilon}\right)^s \exp\left(-\frac{m\epsilon^2}{32}\right). \tag{42}$$

Setting $\epsilon = 1/2$ implies that the minimum singular value of A_T is lower bounded,

$$\sigma_{\min}\left(A_T\right) \geq \frac{\sqrt{m}}{2}, \tag{43}$$

with probability at least $1 - \exp\left(-\frac{Cm}{s}\right)$ for a certain constant C. This implies $A_T^T A_T$ is invertible and consequently that condition (31) is satisfied,

$$\begin{aligned} \left(q_{\ell_2}\right)_T &= A_T^T A_T \left(A_T^T A_T\right)^{-1} \operatorname{sign}\left(x_T\right) && (44)\\ &= \operatorname{sign}\left(x_T\right). && (45) \end{aligned}$$

All is left is to bound q_{ℓ_2} on T^c. We define

$$w := A_T \left(A_T^T A_T\right)^{-1} \operatorname{sign}\left(x_T\right). \tag{46}$$

Since the entries of A are all independent, and w only depends on A^T, it is independent of any column of A_{T^c}. This means that for each $i \in T^c$

$$\begin{aligned} \left(q_{\ell_2}\right)_i &= A_i^T A_T \left(A_T^T A_T\right)^{-1} \operatorname{sign}\left(x_T\right) && (47)\\ &= A_i^T w && (48) \end{aligned}$$

where A_i and w are independent.

By (43), we can bound the norm of w

$$\left|\left|w\right|\right|_2 \leq \frac{\left|\left|\operatorname{sign}\left(x_T\right)\right|\right|_2}{\sigma_{\min}\left(A_T\right)} \leq 2\sqrt{\frac{s}{m}} \tag{49}$$

with probability $1 - \exp\left(-\frac{Cm}{s}\right)$.

Conditioned on w, $A_i^T w$ is Gaussian with mean 0 and variance $\left|\left|w\right|\right|_2^2$. We have

$$\begin{aligned} \mathrm{P}\left(\left|A_i^T w\right| \geq 1 | w = w'\right) &\leq \mathrm{P}\left(|u| > \frac{1}{\left|\left|w'\right|\right|_2}\right) && (50)\\ &\leq 2\exp\left(-\frac{1}{2\left|\left|w'\right|\right|_2^2}\right), && (51) \end{aligned}$$

where u has mean 0 and variance 1. We have applied the following deviation bound, proved in Section A.9 of the appendix.

Lemma 3.5. *For a Gaussian random variable u with zero mean and unit variance and any $t > 0$*

$$\mathrm{P}\left(|u| \geq t\right) \leq 2\exp\left(-\frac{t^2}{2}\right). \tag{52}$$

Let us define the event

$$\mathcal{E} := \left\{||w||_2 \leq 2\sqrt{\frac{s}{m}}\right\}. \tag{53}$$

(51) implies that

$$\mathrm{P}\left(\left|A_i^T w\right| \geq 1 \middle| \mathcal{E}\right) \leq 2\exp\left(-\frac{m}{8s}\right). \tag{54}$$

As a result,

$$\mathrm{P}\left(\left|A_i^T w\right| \geq 1\right) \leq \mathrm{P}\left(\left|A_i^T w\right| \geq 1 \middle| \mathcal{E}\right) + \mathrm{P}\left(\mathcal{E}^c\right) \tag{55}$$

$$\leq \exp\left(-\frac{Cm}{s}\right) + 2\exp\left(-\frac{m}{8s}\right). \tag{56}$$

By the union bound,

$$\mathrm{P}\left(\bigcup_{i \in T^c} \left\{\left|A_i^T w\right| \geq 1\right\}\right) \leq n\left(\exp\left(-\frac{Cm}{s}\right) + 2\exp\left(-\frac{m}{8s}\right)\right). \tag{57}$$

We can consequently choose a constant C' so that if the number of measurements satisfies

$$m \geq C' s \log n \tag{58}$$

we have exact recovery with probability $1 - \frac{1}{n}$.

3.4 Bounds on singular values of random matrices

In this section we provide the proof of Proposition 3.4, which illustrates a useful method to bound the action of a linear operator on a set of infinite cardinality. Let $\mathcal{X}_T$ be the set of unit-norm vectors x with support T. By Proposition 2.7 we have that for any fixed unit-norm vector v

$$(1 - \epsilon) \leq \frac{1}{m}\left||Av\right||_2^2 \leq (1 + \epsilon) \tag{59}$$

with probability $1 - 2\exp\left(-\frac{m\epsilon^2}{8}\right)$. This does not immediately imply the result that we want: the bounds must hold for all vectors in $\mathcal{X}_T$ and we cannot just apply the union bound because the set has infinite cardinality. To overcome this obstacle, we first apply the union bound on a subset of $\mathcal{X}_T$ called an ϵ-net instead.

Definition 3.6 (Net). *An ϵ-net of a set $\mathcal{X}$ is a subset $\mathcal{N}_\epsilon \subseteq \mathcal{X}$ such that for every point $y \in \mathcal{X}$ there exists $z \in \mathcal{N}_\epsilon$ for which*

$$||y - z||_2 \leq \epsilon. \tag{60}$$

Definition 3.7 (Covering number). *The covering number $\mathcal{N}(\mathcal{X}, \epsilon)$ of a set $\mathcal{X}$ at scale ϵ is the minimal cardinality of an ϵ-net of $\mathcal{X}$, or equivalently the minimal number of balls of radius ϵ with centers in $\mathcal{X}$ required to cover $\mathcal{X}$.*

The following proposition, proved in Section A.8 of the appendix, provides a bound for the covering number of the s-dimensional sphere $\mathcal{S}^{s-1}$.

Proposition 3.8 (Covering number of a sphere). *The covering number of the s-dimensional sphere $\mathcal{S}^{s-1}$ at scale ϵ satisfies*

$$\mathcal{N}\left(\mathcal{S}^{k-1}, \epsilon\right) \leq \left(\frac{2+\epsilon}{\epsilon}\right)^s. \tag{61}$$

It is not difficult to see that every vector $x' \in \mathbb{R}^n$ with support T can be mapped to a vector $x'_T \in \mathbb{R}^s$ by choosing the entries in T and that $||x'||_2 = ||x'_T||_2$. As a result, Proposition 3.8 provides a bound on the covering number of $\mathcal{X}_T$.

Corollary 3.9. *If $|T| \leq s$, the covering number of $\mathcal{X}_T$ satisfies*

$$\mathcal{N}\left(\mathcal{X}_T, \epsilon\right) \leq \left(\frac{3}{\epsilon}\right)^s. \tag{62}$$

Let $\epsilon_1 := \epsilon/4$ and $\epsilon_2 := \epsilon/2$. Consider an ϵ_1-net $\mathcal{N}_{\epsilon_1}$ of $\mathcal{X}_T$. We define the event

$$\mathcal{E}_{u,\epsilon_2} := \left\{ (1-\epsilon_2)\,||u||_2^2 \leq \left|\left|\frac{1}{k} Au\right|\right|_2^2 \leq (1+\epsilon_2)\,||u||_2^2 \right\}. \tag{63}$$

By Proposition 2.7 and the union bound we have

$$\begin{aligned}
\mathrm{P}\left(\bigcup_{u \in \mathcal{N}_{\epsilon_1}} \mathcal{E}^c_{u,\epsilon_2}\right) &\leq \sum_{u \in \mathcal{N}_{\epsilon_1}} \mathrm{P}\left(\mathcal{E}^c_{u,\epsilon_2}\right) && (64)\\
&\leq |\mathcal{N}_{\epsilon_1}|\,\mathrm{P}\left(\mathcal{E}^c_{u,\epsilon_2}\right) && (65)\\
&\leq 2\left(\frac{12}{\epsilon}\right)^s \exp\left(-\frac{m\epsilon^2}{32}\right). && (66)
\end{aligned}$$

Now, to finish the proof we need to show that the bound on the elements of the net can be used to bound every other element. Let

$$\frac{\sigma_{\max}(A_T)}{\sqrt{m}} := 1 + \alpha \tag{67}$$

where $\sigma_{\max}(A_T)$ denotes the largest singular value of A_T. For all $x' \in \mathcal{X}_T$

$$\frac{1}{\sqrt{m}} ||Ax'||_2 = \frac{1}{\sqrt{m}} ||A_T x'_T||_2 \tag{68}$$

$$\leq 1 + \alpha. \tag{69}$$

For any $x' \in \mathcal{X}_T$, there is a $u \in \mathcal{N}(\mathcal{X}, \epsilon_1)$ such that $||x' - u||_2 \leq \epsilon/4$. This implies

$$\frac{1}{\sqrt{m}} ||Ax'||_2 \leq \frac{1}{\sqrt{m}} ||Au||_2 + \frac{1}{\sqrt{m}} ||A(x' - u)||_2 \tag{70}$$

$$\leq 1 + \frac{\epsilon}{2} + \frac{(1+\alpha)\epsilon}{4}. \tag{71}$$

Since by definition $\sigma_{\max}(A_T)$ is the smallest upper bound for the norm of $||A_T x'_T||_2 = ||Ax'||_2$ such that $||x'_T||_2 = 1$ (which holds because $x' \in \mathcal{X}_T$), we have

$$1 + \alpha \leq 1 + \frac{\epsilon}{2} + \frac{(1+\alpha)\epsilon}{4}, \tag{72}$$

so that

$$\alpha \leq \frac{3\epsilon}{4 - \epsilon} \leq \epsilon. \tag{73}$$

The lower bound on the singular value follows immediately

$$\frac{1}{\sqrt{m}} ||Ax||_2 \geq \frac{1}{\sqrt{m}} (||Au||_2 - ||A(x - u)||_2) \tag{74}$$

$$\geq 1 - \frac{\epsilon}{2} - \frac{(1+\alpha)\epsilon}{4} \tag{75}$$

$$= 1 - \frac{\epsilon}{2} - \frac{(1+\epsilon)\epsilon}{4} \tag{76}$$

$$\geq 1 - \epsilon. \tag{77}$$

3.5 Robustness to noise

In the previous sections we have shown that ℓ_1-norm minimization allows to recover sparse signals exactly with high probability in the absence of noise. In this section we consider recovery from data that is perturbed by noise, as is usually the case in practice. In more detail, given an s-sparse signal $x \in \mathbb{R}^n$ we have access to

$$y = Ax + z \tag{78}$$

where $A \in \mathbb{R}^{m \times n}$ is a random matrix and $z \in \mathbb{R}^m$ is an additive noise term. A necessary condition for *any* recovery method to be stable is that A should preserve the energy of the

sparse signal; if the signal is in the null space (or almost in the null space) of the measurement operator then we cannot hope to reconstruct it in the presence of noise. A matrix that satisfies the restricted isometry property (RIP) preserves the energy of *any* random vector; it essentially behaves as an isometry when acting upon these class of vectors.

Definition 3.10 (Restricted isometry property)**.** *If a matrix M satisfies the restricted isometry property with constant ϵ_s then for any s-sparse vector x*

$$(1-\epsilon_s)\left||x\right||_2 \leq \left||Mx\right||_2 \leq (1+\epsilon_s)\left||x\right||_2. \tag{79}$$

The RIP guarantees that the problem of recovering sparse vectors from the corresponding linear measurements is well posed. If M satisfies the RIP for a sparsity level $2s$ then there cannot be two sparse vectors x_1 and x_2 that are very different and yet produce similar measurements y_1 and y_2, since

$$\left||y_2 - y_1\right||_2 = M\left(x_2 - x_1\right) \tag{80}$$

$$\geq (1-\epsilon_{2s})\left||x_2 - x_1\right||_2. \tag{81}$$

Gaussian matrices satisfy the restricted isometry property with high probability, as established in the following theorem. Random Fourier measurements also satisfy the RIP [8, 11].

Theorem 3.11 (Restricted isometry property for Gaussian matrices)**.** *Let $A \in \mathbb{R}^{m\times n}$ be a random matrix with iid Gaussian entries with zero mean and unit variance. $\frac{1}{\sqrt{m}}A$ satisfies the restricted isometry property with a constant ϵ_s with probability $1-\frac{C_2}{n}$ as long as the number of measurements*

$$m \geq \frac{C_1 s}{\epsilon_s^2}\log\left(\frac{n}{s}\right) \tag{82}$$

for two fixed constants $C_1, C_2 > 0$.

Proof. By Proposition 3.4 we have that for a fixed support T,

$$(1-\epsilon)\left||x\right||_2 \leq \frac{1}{\sqrt{m}}\left||Ax\right||_2 \leq (1+\epsilon)\left||x\right||_2 \tag{83}$$

for any x with support T with probability at least

$$1 - 2\left(\frac{12}{\epsilon}\right)^s \exp\left(-\frac{m\epsilon^2}{32}\right). \tag{84}$$

There are

$$\binom{n}{s} \leq \left(\frac{en}{s}\right)^s \tag{85}$$

possible supports so by the union bound the result holds with probability at least

$$1 - 2\left(\frac{en}{s}\right)^s \left(\frac{12}{\epsilon}\right)^s \exp\left(-\frac{m\epsilon^2}{32}\right) = 1 - \exp\left(\log 2 + s + s\log\left(\frac{n}{s}\right) + s\log\left(\frac{12}{\epsilon}\right) - \frac{m\epsilon^2}{2}\right)$$
$$\leq 1 - \frac{C_2}{n} \tag{86}$$

for some constant C_2 as long as m satisfies (82). □

The RIP not only establishes that the recovery problem is well posed at least in principle, it also implies that ℓ_1-norm minimization achieves stable recovery in the presence of noise. Let us assume that we know an upper bound for the ℓ_2 norm of the noise in model (78). We can then relax the equality constraint in Problem (106) to an inequality constraint that takes into account the noise level.

$$\text{minimize} \quad ||\tilde{x}||_1 \tag{87}$$
$$\text{subject to} \quad ||A\tilde{x} - y||_2 \leq \sigma. \tag{88}$$

If the RIP constant for a matrix is $\epsilon_{2s} < \sqrt{2} - 1$ and x is s-sparse the solution $\hat{x}$ to the relaxed problem satisfies

$$||\hat{x} - x||_2 \leq C_0\, \sigma \tag{89}$$

for a certain constant. In fact, even if the original vector x is not sparse the solution will will satisfy

$$||\hat{x} - x||_2 \leq C_0\, \epsilon_0 + C_1 \frac{||x - x_s||_1}{\sqrt{s}} \tag{90}$$

where x_s contains the s entries of x with largest magnitude. We refer to [5] for the proof of this result.

4 Sampling

In this section we describe an application of compressed sensing to recovering signals that have a sparse spectrum.

4.1 Nyquist-Shannon sampling theorem

Consider a bandlimited signal $g \in \mathbb{L}_2([0,1])$, which is bandlimited. Its spectrum is equal to zero beyond a certain cut-off frequency f,

$$g(t) := \sum_{k=-f}^{f} c_k \exp(i2\pi kt). \tag{91}$$

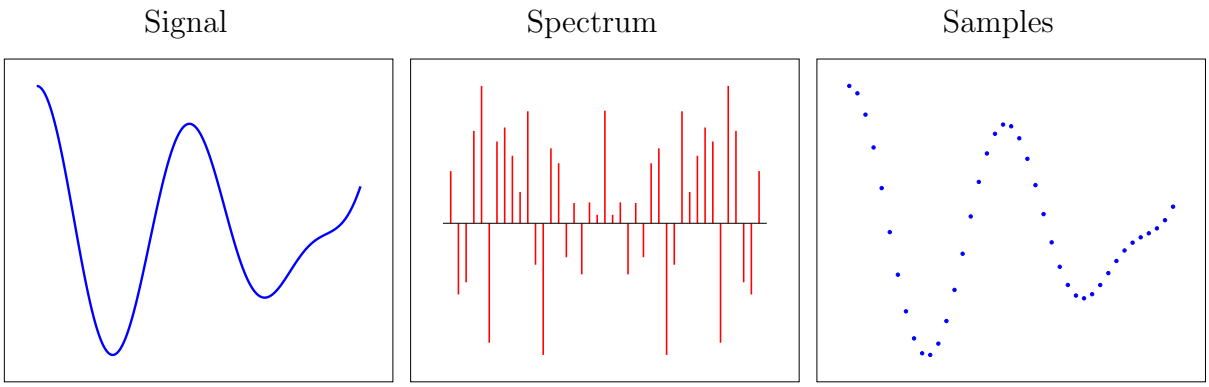


Figure 10: Bandlimited signal (left), corresponding spectrum (center) and regular samples (right).

Our aim is to estimate the signal from samples taken at regular intervals $g(0)$, $g\left(\frac{1}{n}\right)$, $g\left(\frac{2}{n}\right)$, $\ldots$, $g\left(\frac{n-1}{n}\right)$, where the sampling rate is determined by n. Such situations arise when we need to store an analog signal such as music or speech digitally. Two important questions are:

1. What sampling rate is necessary to preserve all the information in the signal?
2. How can we reconstruct the original signal from the sample?

Figure 10 shows an example of a bandlimited signal, its spectrum and the corresponding samples.

In order to simplify notation we define

$$a_{-f:f}(t)_k := \begin{bmatrix} \exp(-i2\pi(-f)t) \\ \exp(-i2\pi(-f+1)t) \\ \cdots \\ \exp(-i2\pi(f-1)t) \\ \exp(-i2\pi ft) \end{bmatrix}. \tag{92}$$

We can now easily express g in terms of $a_{-f:f}$ and the vector of Fourier coefficients c,

$$g(t) := \sum_{k=-f}^{f} c_k \exp(i2\pi kt) = a_{-f:f}(t)^* c. \tag{93}$$

For a fixed sampling rate the samples provide a system of n linear equations

$$F^* c = \begin{bmatrix} a_{-f:f}(0)^* \\ a_{-f:f}\left(\frac{1}{n}\right)^* \\ a_{-f:f}\left(\frac{2}{n}\right)^* \\ \cdots \\ a_{-f:f}\left(\frac{n-1}{n}\right)^* \end{bmatrix} c = \begin{bmatrix} g(0) \\ g\left(\frac{1}{n}\right) \\ g\left(\frac{2}{n}\right) \\ \cdots \\ g\left(\frac{n-1}{n}\right) \end{bmatrix}. \tag{94}$$

Since there are $2f+1$ unknowns we need $n \geq 2f+1$ to guarantee that the linear system has a unique solution. Setting $n = 2f+1$, we can check that the vectors

$$\frac{1}{\sqrt{n}} a_{-f:f}(0), \ \frac{1}{\sqrt{n}} a_{-f:f}\left(\frac{1}{n}\right), \ldots, \frac{1}{\sqrt{n}} a_{-f:f}\left(\frac{n-1}{n}\right) \tag{95}$$

form an orthonormal basis (they actually form a DFT matrix!), so applying the adjoint matrix allows to invert the system

$$c = \frac{1}{n} F F^* c = \frac{1}{n} F \begin{bmatrix} g(0) \\ g\left(\frac{1}{n}\right) \\ g\left(\frac{2}{n}\right) \\ \cdots \\ g\left(\frac{n-1}{n}\right) \end{bmatrix} = \frac{1}{n} \sum_{j=0}^{n} g\left(\frac{j}{n}\right) a_{-f:f}\left(\frac{j}{n}\right). \tag{96}$$

In order to interpret this linear inversion in traditional signal processing terms, let us define the periodized sinc or Dirichlet kernel

$$D_f(t) := \frac{1}{n} \sum_{k=-f}^{f} e^{-i2\pi kt} \tag{97}$$

$$= \frac{\sin(\pi n t)}{n \sin(\pi t)}. \tag{98}$$

A plot of this function is shown in Figure 11. We can easily express the result of shifting D by τ in terms of $a_{-f:f}$

$$D_f(t-\tau) = \frac{1}{n} \sum_{k=-f}^{f} e^{-i2\pi k(t-\tau)} \tag{99}$$

$$= \frac{1}{n} a_{-f:f}(t)^* a_{-f:f}(\tau). \tag{100}$$

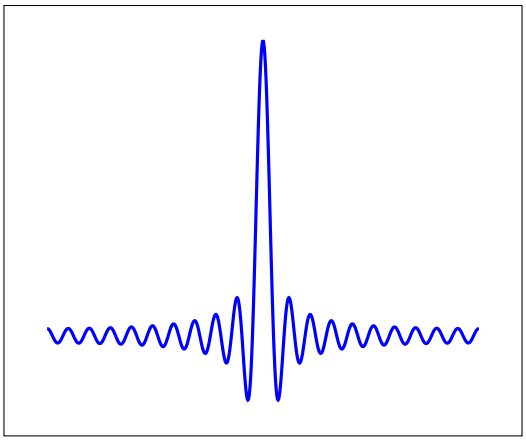


Figure 11: Periodized sinc or Dirichlet kernel.

This allows us to establish that obtaining the Fourier coefficients through linear inversion and then using them to reconstruct the signal is equivalent to interpolation with shifted sinc functions, as sketched in Figure 12.

$$g(t) = a_{-f:f}(t)^* c \tag{101}$$

$$= \frac{1}{n}\sum_{j=0}^{n} g\left(\frac{j}{n}\right) a_{-f:f}(t)^* a_{-f:f}\left(\frac{j}{n}\right) \qquad \text{by (96)} \tag{102}$$

$$= \sum_{j=0}^{n} g\left(\frac{j}{n}\right) D_f\left(t - \frac{j}{n}\right). \tag{103}$$

We conclude that the sampling rate should be twice the cut-off frequency and that reconstruction can be carried out by interpolating the samples with a sinc function. This is known as the Nyquist-Shannon sampling theorem.

4.2 Compressive sampling

We now consider the problem of sampling a signal $g \in \mathbb{L}_2([0,1])$ with a sparse spectrum

$$g(t) := \sum_{k \in \mathcal{S}} c_k \exp(i2\pi kt) \tag{104}$$

and then recovering it from its samples. The signal consists of s frequency components and has a cut-off frequency of f. According to the Shannon-Nyquist sampling theorem, we need $2f+1$ samples in order to preserve the information in the signal and recover it applying

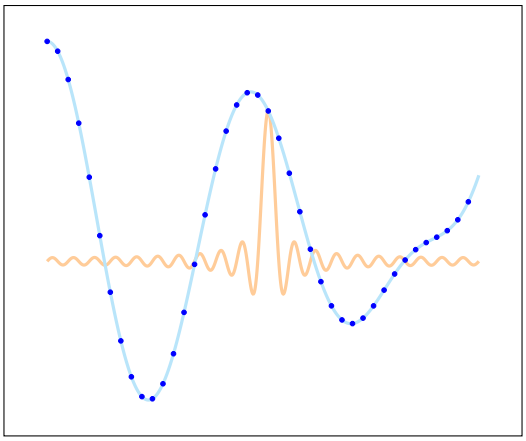


Figure 12: Reconstruction via linear inversion is equivalent to interpolation with a sinc function.

sinc interpolation. In contrast, compressed sensing theory implies that recovery via ℓ_1-norm minimization is possible from $\mathcal{O}\left(s \log\left(2f+1\right)\right)$ randomized samples of the form

$$F_\Omega^* c = \begin{bmatrix} \cancel{a_{-f:f}\left(0\right)^*} \\ a_{-f:f}\left(\frac{1}{n}\right)^* \\ \cancel{a_{-f:f}\left(\frac{2}{n}\right)^*} \\ \cdots \\ a_{-f:f}\left(\frac{n-1}{n}\right)^* \end{bmatrix} c = \begin{bmatrix} \cancel{g\left(0\right)} \\ g\left(\frac{1}{n}\right) \\ \cancel{g\left(\frac{2}{n}\right)} \\ \cdots \\ g\left(\frac{n-1}{n}\right) \end{bmatrix} := y_\Omega, \tag{105}$$

where Ω is a random subset of $\{1, 2, \ldots, 2f+1\}$. In detail, we can estimate the Fourier coefficients c by solving

$$\begin{aligned} &\text{minimize} && ||\tilde{c}||_1 && (106) \\ &\text{subject to} && y_\Omega = F_\Omega^* \tilde{c} && (107) \end{aligned}$$

and then reconstruct g using (96). If $s \ll 2f+1$ this procedure allows to reduce the number of measurements significantly with respect to traditional Nyquist sampling. Figure 13 shows an example which compares the two sampling schemes.

References

The proof of the Johnson-Lindenstrauss lemma is based on [9]. The proofs of exact recovery via ℓ_1-norm minimization and of the restricted isometry property of Gaussian matrices are

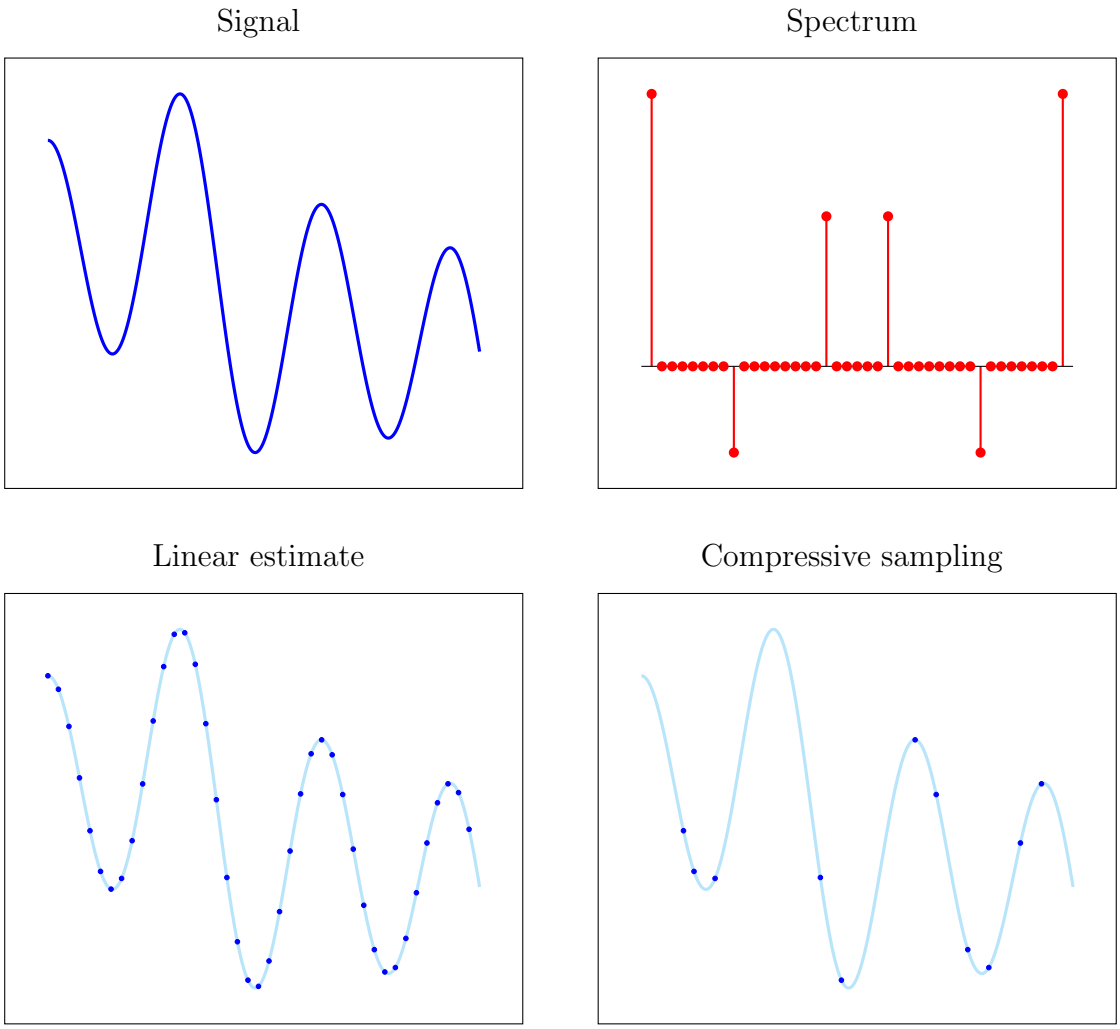


Figure 13: Example showing the samples that are necessary to recover a signal (top left) with a sparse spectrum (top right). Compressive sampling (bottom right) requires significantly less samples than traditional Nyquist sampling (bottom left).

based on arguments in [3] and [2]. For further reading on mathematical tools used to analyze compressed sensing we refer to [12] and [10].

[1] D. Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4):671–687, 2003.

[2] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[3] E. Candès and B. Recht. Simple bounds for recovering low-complexity models. *Mathematical Programming*, 141(1-2):577–589, 2013.

[4] E. Candès and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse problems*, 23(3):969, 2007.

[5] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9):589–592, 2008.

[6] E. J. Candès and Y. Plan. A probabilistic and ripless theory of compressed sensing. *IEEE Transactions on Information Theory*, 57(11):7235–7254, 2011.

[7] E. J. Candès, J. K. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[8] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: universal encoding strategies? *IEEE Transactions in Information Theory*, 52:5406–5425, 2006.

[9] S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

[10] S. Foucart and H. Rauhut. *A mathematical introduction to compressive sensing*, volume 1. 2013.

[11] M. Rudelson and R. Vershynin. On sparse reconstruction from fourier and gaussian measurements. *Communications on Pure and Applied Mathematics*, 61(8):1025–1045, 2008.

[12] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

A Proofs

A.1 Proof of Lemma 2.2

$x - UU^T x$ is orthogonal to any vector $y := Uc \in \mathcal{S}$ since

$$y^T(x - UU^T x) = c^T U^T x - c^T U^T U U^T x \tag{108}$$
$$= c^T U^T x - c^T U^T x \tag{109}$$
$$= 0. \tag{110}$$

By Pythagoras's theorem this implies

$$||x - y||_2^2 = \left|\left|x - UU^T x + UU^T x - y\right|\right|_2^2 \tag{111}$$
$$= \left|\left|x - UU^T x\right|\right|_2^2 + \left|\left|UU^T x - y\right|\right|_2^2 \tag{112}$$

which is minimized by $y = UU^T x$.

A.2 Proof of Theorem 2.5

We will need the following lemma, proved in Section A.3.

Lemma A.1. *For any matrix $X \in \mathbb{R}^{n \times k}$, where $k > n$, with left singular vectors $u_1, u_2, \ldots, u_n$ corresponding to the nonzero singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$,*

$$\sigma_1 = \max_{||u||_2 = 1} \left|\left|X^T u\right|\right|_2, \tag{113}$$
$$u_1 = \arg\max_{||u||_2 = 1} \left|\left|X^T u\right|\right|_2, \tag{114}$$
$$\sigma_m = \max_{\substack{||u||_2 = 1 \\ u \perp u_1, \ldots, u_{m-1}}} \left|\left|X^T u\right|\right|_2, \quad 2 \leq m \leq n, \tag{115}$$
$$u_m = \arg\max_{\substack{||u||_2 = 1 \\ u \perp u_1, \ldots, u_{m-1}}} \left|\left|X^T u\right|\right|_2, \quad 2 \leq m \leq n. \tag{116}$$

We will prove the result by induction on m. The base case $m = 1$ follows immediately from (114). To complete the proof we need to show that if the result is true for $m - 1 \geq 1$ (the induction hypothesis) then it also holds for m.

Let $\mathcal{S}$ be an arbitrary subspace of dimension m. We choose an orthonormal basis for the subspace $b_1, b_2, \ldots, b_m$ such that b_m is orthogonal to $u_1, u_2, \ldots, u_{m-1}$. Note that such a vector must exist because otherwise $\mathcal{S}$ cannot have dimension m.

By the induction hypothesis,

$$\sum_{i=1}^{m-1} \left|\left|X^T u_i\right|\right|_2^2 = \sum_{i=1}^{k} \left|\left|\mathcal{P}_{\text{span}(u_1,u_2,\ldots,u_{m-1})}\, x_i\right|\right|_2^2 \tag{117}$$

$$\geq \sum_{i=1}^{k} \left|\left|\mathcal{P}_{\text{span}(b_1,b_2,\ldots,b_{m-1})}\, x_i\right|\right|_2^2 \tag{118}$$

$$= \sum_{i=1}^{m-1} \left|\left|X^T b_i\right|\right|_2^2. \tag{119}$$

By (116)

$$\sum_{i=1}^{n} \left|\left|\mathcal{P}_{\text{span}(u_k)}\, x_i\right|\right|_2^2 = \left|\left|X^T u_k\right|\right|_2^2 \tag{120}$$

$$\geq \left|\left|X^T b_k\right|\right|_2^2. \tag{121}$$

Combining (119) and (121) we conclude

$$\sum_{i=1}^{k} \left|\left|\mathcal{P}_{\text{span}(u_1,u_2,\ldots,u_m)}\, x_i\right|\right|_2^2 = \sum_{i=1}^{m} \left|\left|X^T u_i\right|\right|_2^2 \tag{122}$$

$$\geq \sum_{i=1}^{m} \left|\left|X^T b_i\right|\right|_2^2 \tag{123}$$

$$\geq \sum_{i=1}^{k} \left|\left|\mathcal{P}_{\mathcal{S}}\, x_i\right|\right|_2^2. \tag{124}$$

A.3 Proof of Lemma A.1

The left singular vectors are an orthonormal basis of $\mathbb{R}^n$, so we can represent any unit-norm vector h_m that is orthogonal to $u_m, \ldots, u_{m-1}$ as

$$h_k = \sum_{i=m}^{n} \alpha_i u_i \tag{125}$$

where

$$||h_m||_2^2 = \sum_{i=m}^{n} \alpha_i^2 = 1. \tag{126}$$

Note that h_1 is just an arbitrary unit-norm vector.

Now we have

$$\left|\left|X^T h_m\right|\right|_2^2 = \sum_{i=1}^{n} \sigma_i \left(\sum_{j=m}^{n} \alpha_j u_i^T u_j \right)^2 \tag{127}$$

$$= \sum_{i=m}^{n} \sigma_i \alpha_i^2 \quad \text{because } u_1, \ldots, u_m \text{ is an orthonormal basis} \tag{128}$$

$$\leq \sigma_m \sum_{i=m}^{n} \alpha_i^2 \quad \text{because } \sigma_m \geq \sigma_{m+1} \geq \ldots \geq \sigma_n \tag{129}$$

$$= \sigma_m \quad \text{by (126).} \tag{130}$$

This establishes (113) and (115). To prove (114) and (116) we just need to show that u_m achieves the maximum

$$\left|\left|X^T u_m\right|\right|_2^2 = \sum_{i=1}^{n} \sigma_i \left(u_i^T u_m\right)^2 \tag{131}$$

$$= \sigma_m. \tag{132}$$

A.4 Proof of Proposition 2.7

The proof relies on the following concentration bounds for χ^2 or chi-square random variables, proved in Section A.5 of the appendix.

Proposition A.2 (Tail bound for chi-square random variables)**.** *Let Z be a χ^2 or chi-square random variable with m degrees of freedom, i.e.*

$$Z = \sum_{i=1}^{m} X_i^2 \tag{133}$$

where $X_1, \ldots, X_m$ are independent Gaussian random variables with zero mean and unit variance. For any $\epsilon > 0$ we have

$$P\left(Z > m\left(1+\epsilon\right)\right) \leq \exp\left(-\frac{m\epsilon^2}{8}\right), \tag{134}$$

$$P\left(Z < m\left(1-\epsilon\right)\right) \leq \exp\left(-\frac{m\epsilon^2}{2}\right). \tag{135}$$

Let $a_i := v^T A_i$ where A_i is the ith row of A and hence an n-dimensional random vector with zero mean and covariance matrix equal to the identity I. a_i is a Gaussian random variable with zero mean and variance $||v||_2^2$. This follows from the following well-known lemma (we omit the proof).

Lemma A.3 (Linear transformation of a Gaussian vector)**.** *Let $M \in \mathbb{R}m \times n$. If an n-dimensional Gaussian vector v has mean μ and covariance matrix Σ, Mv is an m-dimensional Gaussian vector with mean $M\mu$ and covariance matrix $M\Sigma M^T$.*

As a result, we have that

$$Z := \frac{||Av||_2^2}{||v||_2^2} \tag{136}$$

$$= \sum_{i=1}^{m} \frac{a_i^2}{||v||_2^2} \tag{137}$$

is a chi-square random variable with m degrees of freedom. Let us define the events

$$\mathcal{E}_1 := \left\{ ||f(v)||_2 > (1+\epsilon)\,||v||_2 \right\}, \tag{138}$$

$$\mathcal{E}_2 := \left\{ ||f(v)||_2 < (1-\epsilon)\,||v||_2 \right\}. \tag{139}$$

Inequalities (134) and (135) applied to Z imply

$$P(\mathcal{E}_1) = P\left(Z > m(1+\epsilon)^2\right) \tag{140}$$

$$\leq \exp\left(-\frac{m\epsilon^2}{8}\right), \tag{141}$$

$$P(\mathcal{E}_2) = P\left(Z < m(1-\epsilon)^2\right) \tag{142}$$

$$\leq \exp\left(-\frac{m\epsilon^2}{2}\right). \tag{143}$$

By the union bound

$$P(\mathcal{E}_1 \cup \mathcal{E}_2) \leq P(\mathcal{E}_1) + P(\mathcal{E}_2) \tag{144}$$

$$\leq 2\exp\left(-\frac{m\epsilon^2}{8}\right). \tag{145}$$

A.5 Proof of Proposition A.2

Proof of (134) The concentration bound is established by applying Markov's inequality after exponentiation, as is commonly done to prove Chernoff bounds.

Proposition A.4 (Markov's inequality)**.** *Let X be a nonnegative random variable. For any positive constant $a > 0$,*

$$\mathrm{P}(X \geq a) \leq \frac{\mathrm{E}(X)}{a}. \tag{146}$$

Markov's inequality is proved in Section A.10 below. Let $t > 0$ be an arbitrary positive number, we have

$$
\begin{aligned}
P\left(Z > a\right) &= P\left(\exp\left(tZ\right) > \exp\left(at\right)\right) && (147)\\
&\leq \exp\left(-at\right) \mathrm{E}\left(\exp\left(tZ\right)\right) \qquad \text{by Markov's inequality} && (148)\\
&\leq \exp\left(-at\right) \mathrm{E}\left(\exp\left(\sum_{i=1}^{m} tX_i^2\right)\right) && (149)\\
&\leq \exp\left(-at\right) \prod_{i=1}^{m} \mathrm{E}\left(\exp\left(tX_i^2\right)\right) \quad \text{by independence of } X_1, \ldots, X_m. && (150)
\end{aligned}
$$

Some calculus yields

$$
\begin{aligned}
\mathrm{E}\left(\exp\left(tX^2\right)\right) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp\left(-\frac{u^2}{2}\right) \exp\left(tu^2\right) \mathrm{d}\, u && (151)\\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp\left(-\frac{\left(1-2t\right)u^2}{2}\right) \mathrm{d}\, u && (152)\\
&= \frac{1}{\sqrt{2\pi\left(1-2t\right)}} \int_{-\infty}^{\infty} \exp\left(-\frac{v^2}{2}\right) \mathrm{d}\, v \qquad \text{change of variables } v = \left(2-t\right)u\\
&= \frac{1}{\sqrt{1-2t}} && (153)
\end{aligned}
$$

Combining (150) and (153),

$$
P\left(Z > a\right) \leq \frac{\exp\left(-at\right)}{\left(1-2t\right)^{\frac{m}{2}}}. \tag{154}
$$

Setting

$$
\begin{aligned}
t &:= \frac{1}{2} - \frac{1}{2\left(1+\epsilon\right)}, && (155)\\
a &:= m\left(1+\epsilon\right) && (156)
\end{aligned}
$$

we have

$$
\begin{aligned}
P\left(Z > m\left(1+\epsilon\right)\right) &\leq \left(1+\epsilon\right)^m 2\exp\left(-\frac{m\epsilon}{2}\right) && (157)\\
&= \exp\left(-\frac{m}{2}\left(\epsilon - \log\left(1+\epsilon\right)\right)\right). && (158)
\end{aligned}
$$

The function $g\left(x\right) := x - \frac{x^2}{4} - \log\left(1+x\right)$ is nonnegative between 0 and 1 (the derivative is nonnegative and $g\left(0\right) = 0$). This implies

$$
P\left(Z > m\left(1+\epsilon\right)\right) \leq \exp\left(-\frac{m\epsilon^2}{8}\right). \tag{159}
$$

Proof of (135)

A very similar argument to the one that yields (150) gives

$$P\left(Z < a'\right) = P\left(\exp\left(-t'Z\right) > \exp\left(-a't'\right)\right) \tag{160}$$

$$\leq \exp\left(a't'\right) \prod_{i=1}^{m} \mathrm{E}\left(\exp\left(-t'X_i^2\right)\right). \tag{161}$$

Setting $t' = t$ in (153), we have

$$\mathrm{E}\left(\exp\left(-t'X^2\right)\right) = \frac{1}{\sqrt{1+2t'}}. \tag{162}$$

This implies

$$P\left(Z < a'\right) \leq \frac{\exp\left(a't'\right)}{\left(1+2t'\right)^{\frac{m}{2}}}. \tag{163}$$

Setting

$$t' := -\frac{1}{2} + \frac{1}{2\left(1-\epsilon\right)}, \tag{164}$$

$$a' := m\left(1-\epsilon\right) \tag{165}$$

we have

$$P\left(Z < m\left(1-\epsilon\right)\right) \leq \left(1-\epsilon\right)^{\frac{m}{2}} \exp\left(\frac{m\epsilon}{2}\right) \tag{166}$$

$$= \exp\left(-\frac{m}{2}\left(-\epsilon - \log\left(1-\epsilon\right)\right)\right). \tag{167}$$

The function $h\left(x\right) := -x - \frac{x^2}{2} - \log\left(1-x\right)$ is nonnegative between 0 and 1 (the derivative is nonnegative and $g\left(0\right) = 0$). We conclude that

$$P\left(Z < m\left(1-\epsilon\right)\right) \leq \exp\left(-\frac{m\epsilon^2}{2}\right). \tag{168}$$

A.6 Proof of Lemma 3.2

Note that

$$q := A^T v \tag{169}$$

is a subgradient of the ℓ_1 norm at x by Proposition 3.5 in Lecture Notes 2. For any feasible vector $\tilde{x}$, $h := x - \tilde{x}$ satisfies $Ah = 0$. Since q is a subgradient at x we immediately have

$$\begin{aligned} ||\tilde{x}||_1 &= ||x + h||_1 && (170)\\ &\geq ||x||_1 + q^T h && (171)\\ &= ||x||_1 + v^T A h && (172)\\ &= ||x||_1 . && (173) \end{aligned}$$

This proves that x is a solution but not that it is the only solution. For this we need to use the strict inequality (32) and the assumption that A_T is full rank. If A_T is full rank then $h_{T^c} \neq 0$ unless $h = 0$ because otherwise h_T would be a nonzero vector in the null space of A_T. This together with (32) implies

$$||h_{T^c}||_1 > q^T h_{T^c}. \tag{174}$$

Let $\mathcal{P}_T(\cdot)$ denote a projection that sets to zero all entries of a vector except the ones indexed by T. We have

$$\begin{aligned} ||\tilde{x}||_1 &= ||x + \mathcal{P}_T(h)||_1 + ||h_{T^c}||_1 \qquad \text{because } x \text{ is supported on } T && (175)\\ &> ||x||_1 + q^T \mathcal{P}_T(h) + q^T \mathcal{P}_{T^c}(h) \qquad \text{by (174)} && (176)\\ &= ||x||_1 + q^T h && (177)\\ &= ||x||_1 . && (178) \end{aligned}$$

Since this holds for any arbitrary feasible vector $\tilde{x}$ we conclude that x must be the unique solution.

A.7 Proof of Lemma 3.3

The Lagrangian is equal to

$$\begin{aligned} \mathcal{L}(\tilde{x}, \tilde{v}) &= ||\tilde{x}||_1 + \tilde{v}^T (y - A\tilde{x}) && (179)\\ &= ||\tilde{x}||_1 - \left(A^T \tilde{v}\right)^T \tilde{x} + y^T \tilde{v}. && (180) \end{aligned}$$

To obtain the Lagrange dual function we minimize with respect to $\tilde{x}$,

$$\min_{\tilde{x}} \mathcal{L}(\tilde{x}, \tilde{v}) = \begin{cases} y^T \tilde{v} & \text{if } \left|\left|A^T \tilde{v}\right|\right|_\infty \leq 1, \\ -\infty & \text{otherwise.} \end{cases} \tag{181}$$

The dual problem consists of maximizing this Lagrange dual function.

A.8 Proof of Proposition 3.8

We construct an ϵ covering set $\mathcal{N}_\epsilon \subseteq \mathcal{S}^{s-1}$ recursively:

- We initialize $\mathcal{N}_\epsilon$ to the empty set.
- We choose a point $x \in \mathcal{S}^{s-1}$ such that $||x - y||_2 > \epsilon$ for any $y \in \mathcal{C}$. We add x to $\mathcal{N}_\epsilon$ until there are no points in $\mathcal{S}^{s-1}$ that are ϵ away from any point in $\mathcal{N}_\epsilon$.

This algorithm necessarily ends in a finite number of steps because the n-dimensional sphere is compact (otherwise we would have an infinite sequence such that no subsequence converges).

Now, let us consider the balls of radius $\epsilon/2$ centered at each of the points in $\mathcal{N}_\epsilon$. These balls do not intersect since their centers are at least ϵ apart and they are all inside the ball of radius $1 + \epsilon/2$ centered at the origin because $\mathcal{C} \subseteq \mathcal{S}^{s-1}$. This means that

$$\begin{aligned} \text{Vol}\left(\mathcal{B}^s_{1+\epsilon/2}\left(0\right)\right) &\geq \text{Vol}\left(\cup_{x \in \mathcal{N}_\epsilon} \mathcal{B}^s_{\epsilon/2}\left(x\right)\right) && (182) \\ &= \left|\mathcal{N}_\epsilon\right| \text{Vol}\left(\mathcal{B}^s_{\epsilon/2}\left(0\right)\right) && (183) \end{aligned}$$

where $\mathcal{B}^s_r\left(x\right)$ is the ball of radius r centered at x. By multivariable calculus

$$\text{Vol}\left(\mathcal{B}^s_r\left(0\right)\right) = r^s \, \text{Vol}\left(\mathcal{B}^s_1\left(0\right)\right), \tag{184}$$

so (182) implies

$$\left(1 + \epsilon/2\right)^s \geq \left|\mathcal{N}_\epsilon\right| \left(\epsilon/2\right)^s. \tag{185}$$

A.9 Proof of Lemma 3.5

By symmetry of the Gaussian probability density function, we just need to bound the probability that $u > t$. Applying Markov's inequality (see Proposition A.4) we have

$$\begin{aligned} \text{P}\left(u \geq t\right) &= \text{P}\left(\exp\left(ut\right) \geq \exp\left(t^2\right)\right) && (186) \\ &\leq \text{E}\left(\exp\left(ut - t^2\right)\right) && (187) \\ &= \exp\left(-\frac{t^2}{2}\right) \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp\left(-\frac{\left(u-t\right)^2}{2}\right) \text{d}\, u && (188) \\ &= \exp\left(-\frac{t^2}{2}\right). && (189) \end{aligned}$$

A.10 Proof of Proposition A.4

Consider the indicator variable $1_{X \geq a}$. Clearly the random variable

$$X - a\, 1_{X \geq a} \geq 0. \tag{190}$$

So in particular its expectation is non-negative (as it is the sum or integral of a non-negative quantity over the positive real line). By linearity of expectation and the fact that $1_{X \geq a}$ is a Bernoulli random variable with expectation $\mathrm{P}(X \geq a)$ we have

$$\mathrm{E}(X) \geq a\, \mathrm{E}(1_{X \geq a}) = a\, \mathrm{P}(X \geq a). \tag{191}$$

Super-resolution

1 Super-resolution of point sources

The word *super-resolution* has different meanings in different disciplines. In optics it is usually used to describe data-acquisition techniques designed to overcome the diffraction limit [8]. In image-processing and computer-vision applications the term tends to refer to the problem of obtaining a high-resolution image either from several low-resolution images [10] or by upsampling a single image while preserving its edges and hallucinating textures in a reasonable way [5]. In this lecture, super-resolution denotes the inverse problem of estimating a signal from low-pass measurements. In contrast to the previous definitions, we assume that the data-acquisition mechanism is *fixed* and we aim to *recover* rather than hallucinate the lost high-resolution information.

1.1 Spatial super-resolution

As Lord Rayleigh pointed out in his seminal 1891 paper On Pin-hole Photography, *it has long been known that the resolving power of lenses, however perfect, is limited*. Diffraction imposes an inflexible limit on the resolution of any optical system. However we are often interested in information that is only apparent beyond this limit. For instance, in microscopy, astronomy or medical imaging it may be challenging to discern cellular structures, celestial bodies or incipient tumors from the available data. This is illustrated by the image on the left of Figure 1, which shows measurements of the interior of a cell obtained by fluorescence microscopy. The limited resolution of the microscope produces aliasing artifacts that completely obscure the fine-scale details of the image. The aim of super-resolution is to uncover such fine-scale structure from coarse-scale measurements.

Let x be a high-resolution representation of an object of interest. Mathematically, a reasonable model for the data in many of the applications mentioned in the previous paragraph is

$$x_{\mathrm{LR}} := \phi * x. \tag{1}$$

The signal x is convolved with a low-pass point spread function (PSF) ϕ that depends on the sensing mechanism. This convolution smooths out the fine-scale details producing a low-resolution version of the original signal. The aim of *spatial super-resolution* is to estimate x from x_{LR}, a problem which is often also referred to as *deconvolution*. Recovering the lost fine-scale information amounts to extrapolating the spectrum of x. This becomes apparent

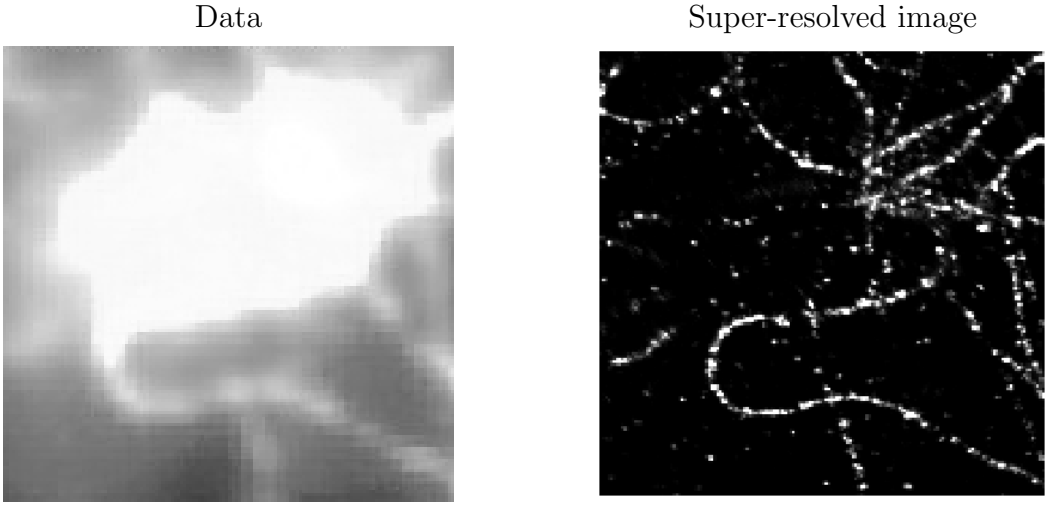


Figure 1: An example of super-resolution applied to cell imaging using fluorescence microscopy data. In both images a superposition of a sequence of frames is shown. The data on the left are contaminated by noise and heavy aliasing. Super-resolving the probes reveals the fine-scale details of the cell, as we can see on the right. The image is due to Veniamin Morgenshtern.

when we consider the measurement process in the frequency domain. If the cut-off frequency of the PSF is equal to f_c, the spectrum of x_{LR}

$$\widehat{x}_{\mathrm{LR}} = \widehat{\phi}\,\widehat{x} = \widehat{\phi}\,\Pi_{[-f_c,f_c]^d}\,\widehat{x}, \tag{2}$$

where d is the ambient dimension, $\widehat{\ }$ denotes the frequency representation of a signal and $\Pi_{[-f_c,f_c]^d}$ is an indicator function that is zero out of the set $[-f_c, f_c]^d$. The high frequency information in the signal is suppressed in the data. Super-resolution aims to extrapolate this information from the low-pass measurements.

In order to super-resolve a signal it is necessary to leverage some prior knowledge about its structure. Otherwise the problem is hopelessly ill posed; the missing spectrum can be filled in arbitrarily to produce estimates that correspond to the data. In this lecture we consider signals that are well modeled as superpositions of point sources,

$$x := \sum_{t_j \in T} c_j \delta_{t_j}, \tag{3}$$

where δ_τ is a Dirac measure at τ, T is a set of locations in the unit interval and the amplitudes a_j may be complex valued. In this case, the low-resolution data are of the form

$$x_{\mathrm{LR}}(t) := \phi * x(t) \tag{4}$$

$$= \sum_{t_j \in T} c_j \phi(t - t_j), \tag{5}$$

as illustrated at the top of Figure 2.

Point sources are used to represent celestial bodies in astronomy, neuron spikes in neuroscience or line spectra in signal processing and spectroscopy. In addition, locating pointwise fluorescent probes is a crucial step in some optical super-resolution procedures capable of handling more complicated objects. Techniques such as photoactivated localization microscopy (PALM) or stochastic optical reconstruction microscopy (STORM) are based on localizing probes that switch randomly between a fluorescent and a non-fluorescent state. To obtain an image of a certain object, multiple frames are gathered. Each frame consists of a superposition of blurred light sources that correspond to the active probes. Deblurring these sources and combining the results allows to super-resolve the object of interest. The image to the right of Figure 1 was generated in this way.

An additional assumption that we make in this lecture is that the PSF of the sensing mechanism is known. This is usually a realistic assumption in point-source super-resolution as long as the measurement process is indeed space-invariant. In such cases the PSF can be estimated by locating an isolated blurred source in the data. For general images or PSFs that are not low pass, which arise for instance due to motion blurring, it is necessary to jointly estimate the PSF and the signal; a problem known as *blind deconvolution* which we will not discuss here.

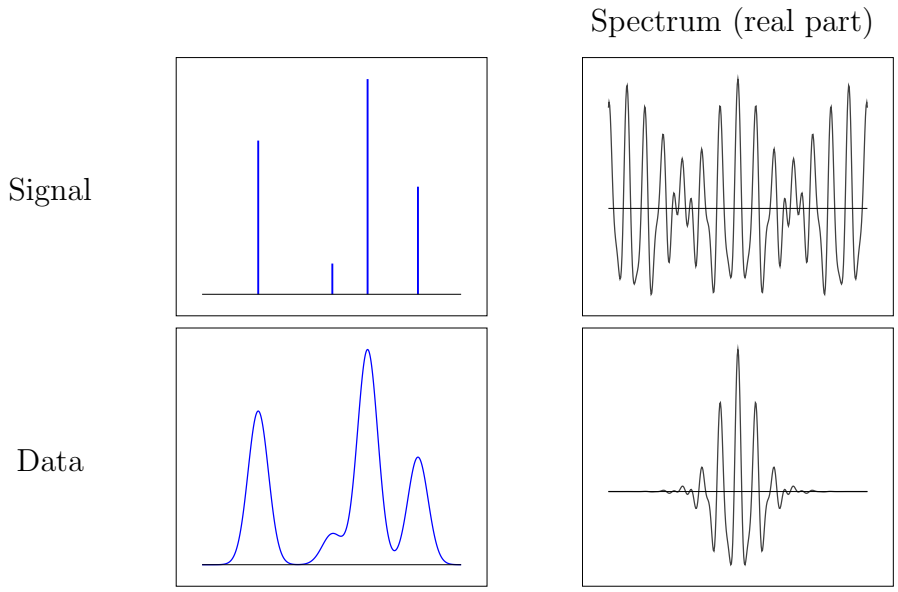
An important instance of the model given by (3) and (4) is when the signal is one dimensional and the Fourier transform of the PSF ϕ is constant over $[-f_c, f_c]$, i.e. ϕ is a periodized sinc or Dirichlet kernel. In this case, since the support of x_{LR} is restricted to the unit interval, it follows from the sampling theorem that its spectrum is completely determined by the discrete samples

$$
\begin{aligned}
y_k &:= \widehat{x}_{\mathrm{LR}}(k) && (6)\\
&= \int_0^1 \exp\left(-i2\pi kt\right) x(\mathrm{d}t) && (7)\\
&= \sum_{t_j \in T} c_j \exp\left(-i2\pi k t_j\right), \quad k \in \mathbb{Z},\ |k| \leq f_c, && (8)
\end{aligned}
$$

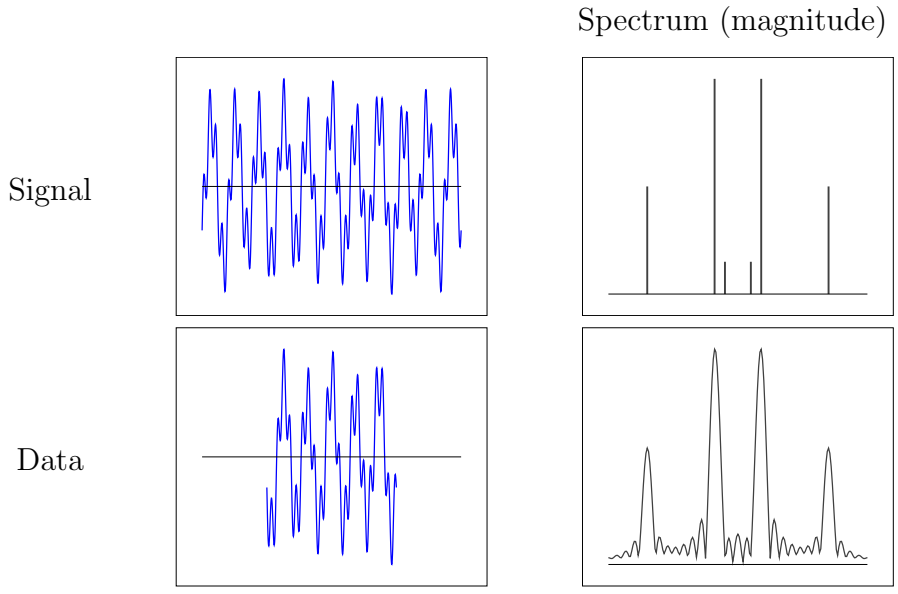
where we assume for simplicity that f_c is an integer. In a more compact form,

$$
y = \mathcal{F}_n\, x \tag{9}
$$

where $y \in \mathbb{C}^n$ and $\mathcal{F}_n$ is the linear operator that maps a measure or function to its lowest $n := 2f_c + 1$ Fourier coefficients.



Spatial super-resolution



Spectral super-resolution

Figure 2: Schematic illustration of spatial and spectral super-resolution.

1.2 Spectral super-resolution

A basic problem in signal processing is to estimate the spectrum of a multisinusoidal signal

$$g(t) := \sum_{f_j \in T} c_j \exp(-i2\pi f_j t) \tag{10}$$

from a finite number of samples. The corresponding spectrum of the signal is of the form

$$\widehat{g} = \sum_{f_j \in T} c_j \delta_{f_j}. \tag{11}$$

Spectra that are composed of a superposition of Dirac measures are called *line spectra*. Estimating line spectra is important for direction-of-arrival (DOA) problems in radar where the aim is to determine the direction from which a propagating wave arrives at an array of sensors.

Let us assume that the spectrum of g is restricted to the interval $[-1/2, 1/2]$. This means that its cut-off frequency is $f = 1/2$. Regular samples taken at the rate dictated by the Nyquist-Shannon theorem $2f = 1$

$$g(k) := \sum_{f_j \in T} c_j \exp(-i2\pi k f_j), \qquad \infty \leq k \leq \infty, \tag{12}$$

suffice to recover $\widehat{g}$. Indeed, these samples $g(k)$ are the Fourier coefficients of $\widehat{g}$!

Unfortunately, in most practical situations we only have access to a finite number of samples. Truncating the measurements in the time domain is equivalent to convolving the spectrum of the signal with a periodized sinc. This induces aliasing in the frequency domain as depicted in the lower half of Figure 2. *Spectral super-resolution* is the problem of estimating the line spectra of g from such measurements. This exactly equivalent to the spatial super-resolution in 1D where the PSF is a sinc. Indeed, $\widehat{g}$ is a superposition of Dirac measures just like x and the measurements are a contiguous subset of the Fourier coefficients of the measure.

1.3 Deconvolution in reflection seismography

Characterizing underground geological structure, such as the strata portrayed in Figure 3, from surface measurements is an important problem in geophysics. Reflection seismology is a technique to achieve this by sending pulses into the ground and analyzing the reflected signals. Figure 4 provides a cartoon description of the problem. The reflections at the interface between two strata are governed by a *reflection coefficient* that corresponds to the difference between the impedance of the two layers. Estimating the reflection coefficients and their positions from the reflected pulse allows to determine the composition of the different strata.

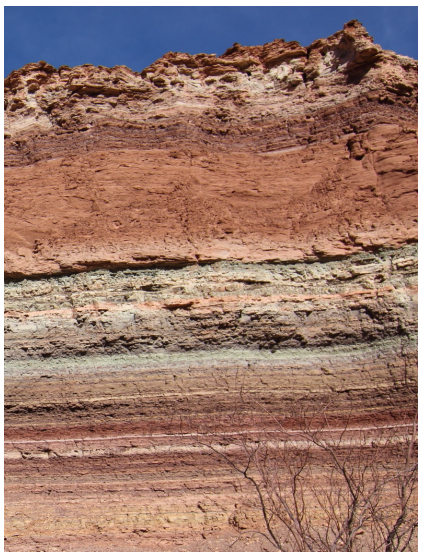


Figure 3: Geological strata in Salta, Argentina.

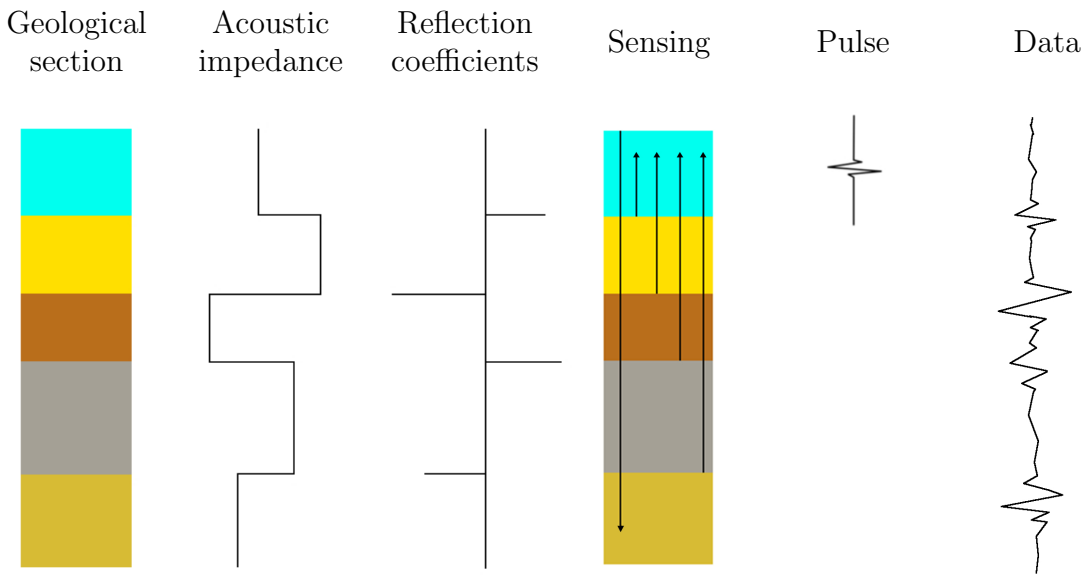


Figure 4: Cartoon description of reflection seismology.

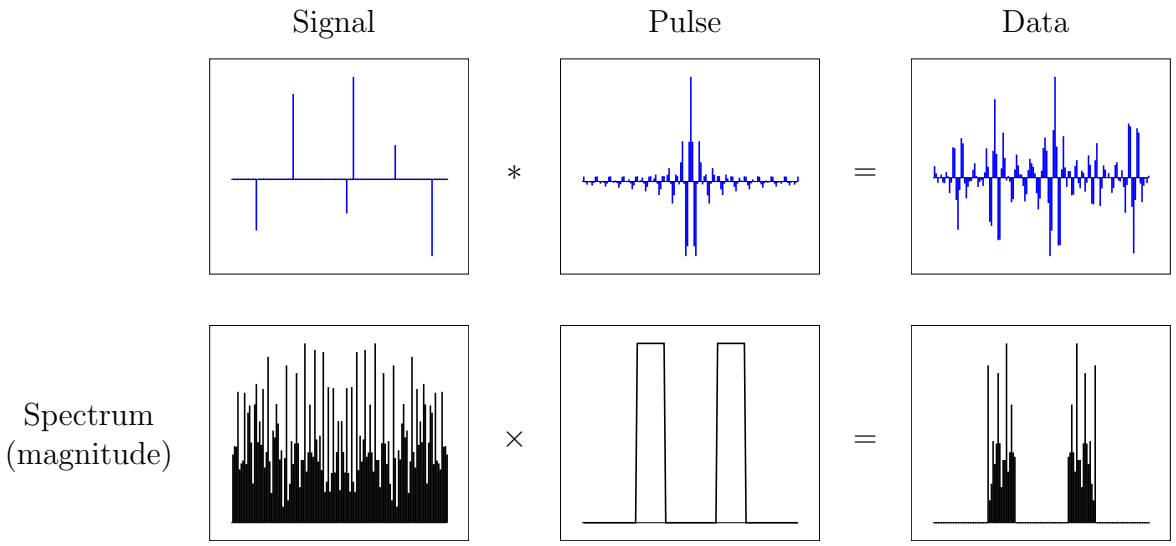


Figure 5: Simplified model for a seismography problem where the aim is to estimate reflection coefficients from bandpass measurements.

A simplified mathematical model of the measurement process is that the data are equal to the convolution of the pulse ϕ and the reflection coefficients x,

$$y = \phi * x \tag{13}$$

where the reflection coefficients can be modeled as a superposition of Dirac measures,

$$x := \sum_{t_j \in T} c_j \delta_{t_j}. \tag{14}$$

The spectrum of pulses used in reflection seismology is often approximately bandpass; it is mostly concentrated in a certain band of frequencies. If we model the pulse as being exactly bandpass, then the spectrum of the data corresponds to a contiguous subset of Fourier coefficients of x, or two contiguous subsets if we consider real bandpass filters as illustrated in Figure 5. The problem is consequently very similar to the simplified models for spatial and spectral super-resolution introduced in the previous sections.

2 Conditioning of the super-resolution problem

In this section we analyze the conditioning of the super-resolution problem when the signal is equal to a superposition of Dirac measures and the data are equal to the first n low-pass Fourier coefficients

$$y = \mathcal{F}_n \, x. \tag{15}$$

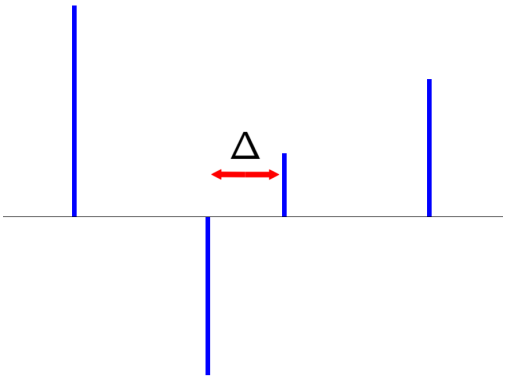


Figure 6: Illustration of the minimum separation of a signal.

As we discussed in the previous lecture, a fundamental insight in the theoretical analysis of compressed sensing is that random measurements conserve the energy of sparse signals with high probability. More specifically, the randomized operator obeys the restricted-isometry property (RIP). This ensures that the inverse problem of sparse recovery from random measurements is well posed at least in principle.

In contrast, sparsity is not enough to make super-resolution a well posed problem. The crucial difference between super-resolution and compressed sensing is the correlation structure of the measurement process. The sensing operator $\mathcal{F}_c$ defined by (19) can be interpreted as an infinite matrix with n rows and infinite columns parametrized by $t \in [0, 1]$. Each of the columns is a sinusoidal atom of the form $\exp i2\pi kt$ for $-f_c \leq k \leq f_c$. If the difference between t_1 and t_2 is small, the corresponding columns

$$(\mathcal{F}_n)_{t_1} = \begin{bmatrix} \exp -i2\pi f_c t_1 \\ \exp -i2\pi \left(f_c - 1\right) t_1 \\ \cdots \\ \exp i2\pi f_c t_1 \end{bmatrix} \approx (\mathcal{F}_n)_{t_2} = \begin{bmatrix} \exp -i2\pi f_c t_2 \\ \exp -i2\pi \left(f_c - 1\right) t_2 \\ \cdots \\ \exp i2\pi f_c t_2 \end{bmatrix} \tag{16}$$

are extremely correlated. As a result, the measurement process does not even satisfy the RIP with sparsity level equal to two! In this case, we cannot recover a 1-sparse signal reliably in the presence of noise. However, if the difference between t_1 and t_2 is large then the atoms are not too correlated. In fact, as we shall demonstrate numerically in a moment, submatrices corresponding to columns that are spread out are well conditioned. This suggests restricting the class of signals of interest to signals that have a support that is not too clustered. To this end, we define the *minimum separation* of the support T of a signal. The concept is illustrated in Figure 6.

Definition 2.1 (Minimum separation)**.** *For a set of points $T \subseteq [0, 1]$, the minimum separation (or minimum distance) is defined as the closest distance between any two elements from*

T,

$$\Delta(T) = \inf_{(t,t') \in T \,:\, t \neq t'} |t - t'|. \tag{17}$$

In order to determine how the linear operator $\mathcal{F}_n$ acts upon signals with supports that satisfy a certain minimum separation, we compute the singular values of the submatrix formed by the corresponding columns. Figure 7 shows the results. For minimum separations above $0.5/f_c$ the submatrices are very well conditioned, but as soon as the minimum separation falls below $0.5/f_c$ some of the singular values of the operator plummet. Below $0.35/f_c$ a fourth of the singular values are extremely small. Below $0.2/f_c$ the magnitude of half of them is negligeable. The effect becomes more pronounced as the cardinality s of the support of the signal increases.

Note that signals in the span of the singular vectors that correspond to very small singular values in Figure 8 are essentially mapped to zero by the low-resolution operator. These signals are consequently almost in the null space of the measurements: they are impossible to estimate under very small perturbations to the data *even if we know the support beforehand*. As a result, a fundamental limit for signal recovery in terms of the minimum separation is $1/f_c$, the inverse of the cut-off frequency (it also corresponds to the width of the main lobe of the point-spread function ϕ, when the PSF is a sinc). To see why, consider two signals x_1 and x_2 with a minimum separation just below $1/f_c$ such their difference $d := x_1 - x_2$ has a minimum separation under $0.5/f_c$. From Figure 7 we can choose d so that

$$\mathcal{F}_n \, d = \mathcal{F}_n \, x_1 - \mathcal{F}_n \, x_2 \approx 0. \tag{18}$$

Even under very low levels of noise it will be impossible to distinguish x_1 and s_2 from low-resolution data. Figure 8 illustrates this: the measurements corresponding to two signals with disjoint supports and a minimum distance of $0.9\lambda_c$ for $f_c = 10^3$ are indeed almost indistinguishable. The phenomenon can be characterized theoretically in an asymptotic setting using Slepian's prolate-spheroidal sequences [12] (see also Section 3.2 in [2]). More recently, [9] (see Theorem 1.3) provides a non-asymptotic analysis.

The take-home lesson is that we can only hope to achieve stable super-resolution of signals with a minimum separation above $0.5/f_c$ if the signal consists of a large number of point sources. For signals with a small number of point sources, robust recovery may be possible below that limit. However, even for very small supports, the problem is always hopelessly ill posed for a small enough value of the minimum distance.

3 Linear methods

In this section we describe some simple linear techniques to tackle the super-resolution problem.

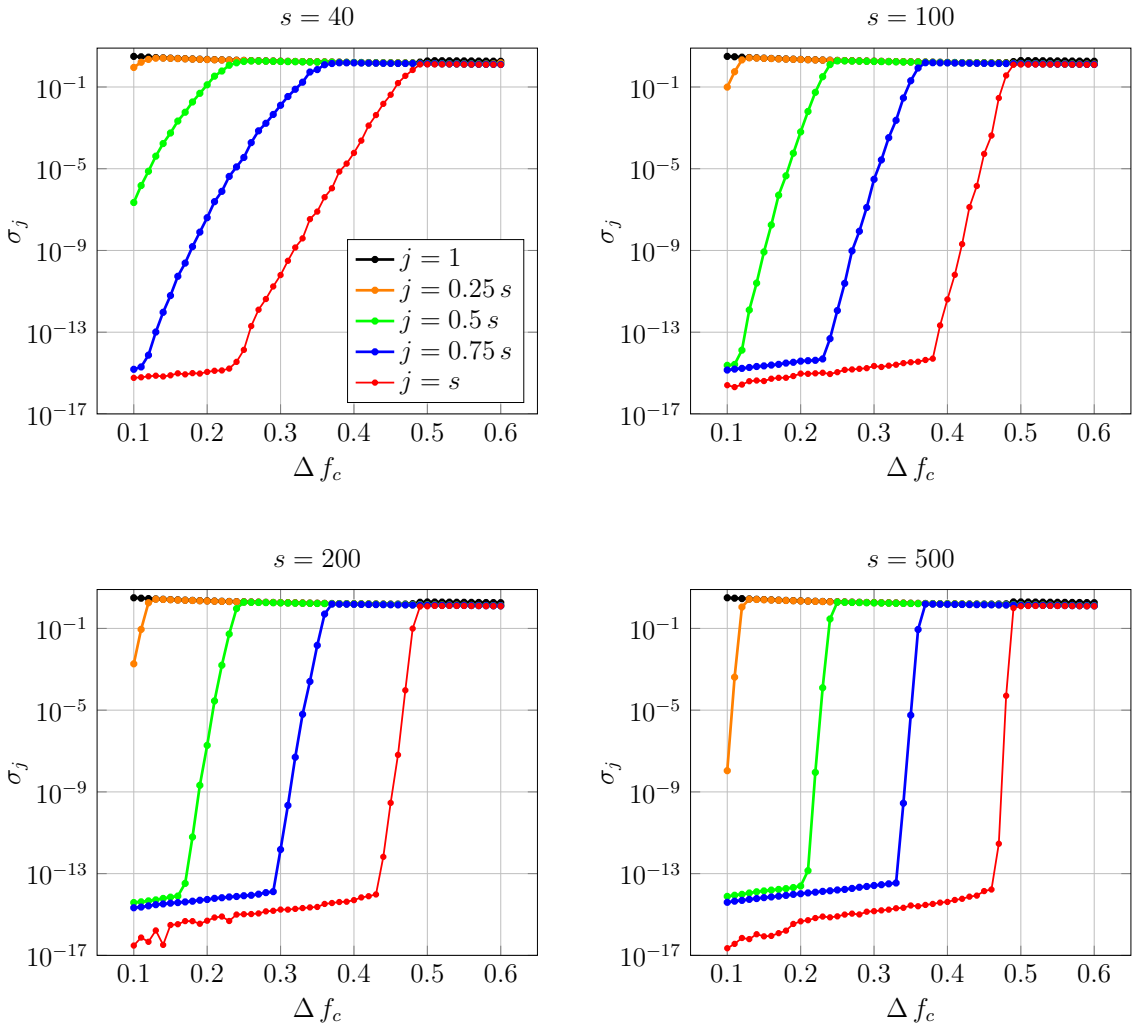


Figure 7: Singular values of the linear operator $\mathcal{F}_n$ when restricted to act upon signals of sparsity s that have a minimum separation of Δ.

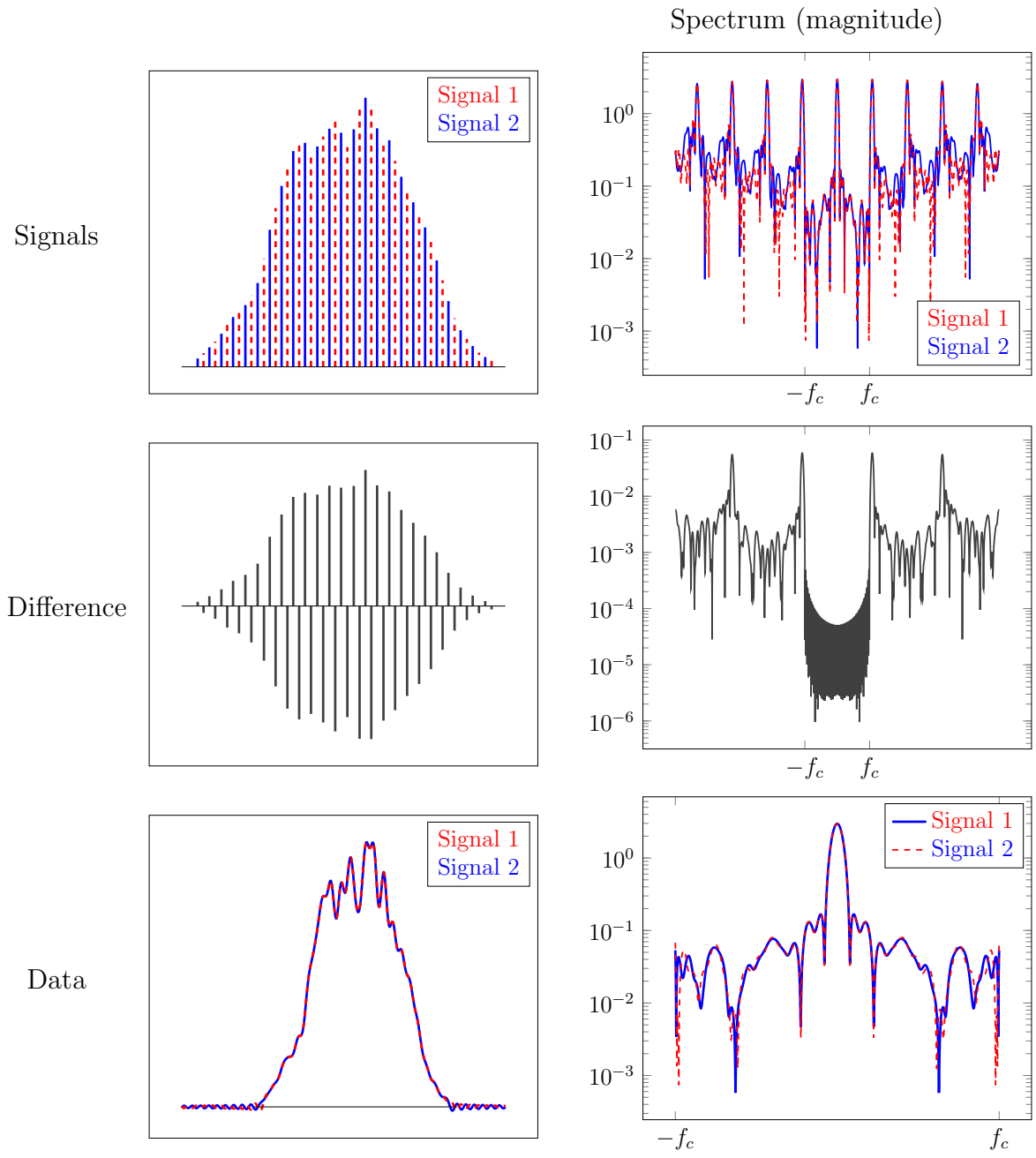


Figure 8: Two signals with disjoint supports that satisfy the minimum-separation condition for $\Delta(T) = 0.9f_c$ when $f_c = 10^3$ (top left) and their spectrum (top right). Their difference (center left) has a spectrum that is concentrated away for the low-pass band between $-f_c$ and f_c (center right). As a result, it is very difficult to distinguish the data corresponding to the two signals (bottom).

3.1 The periodogram

As explained in Section 1.2 in the spectral super-resolution problem the data (12) correspond to a contiguous subset of the Fourier coefficients of the signal

$$y = \mathcal{F}_n \, x. \tag{19}$$

If x corresponds to a superposition of line spectra, this is equivalent to convolving the sparse spectrum with a periodized sinc function. The periodogram recovers this representation by applying the adjoint of the measurement operator $\mathcal{F}_n$ to the data. This is essentially equivalent to projecting the data onto the signal space.

$$\begin{aligned} P\left(t\right) &= \mathcal{F}_n^* \, y && (20)\\ &= \sum_{k=-f_c}^{f_c} y\left(k\right) \exp\left(i2\pi kt\right) && (21)\\ &= \sum_{k=-f_c}^{f_c} \left(\sum_{t_j \in T} c_j \exp\left(i2\pi k t_j\right) \right) \exp\left(i2\pi kt\right) && (22)\\ &= \sum_{t_j \in T} c_j \sum_{k=-f_c}^{f_c} \exp\left(i2\pi k \left(t - t_j\right)\right) && (23)\\ &= \sum_{t_j \in T} c_j D_{f_c}\left(t - t_j\right), && (24) \end{aligned}$$

where D_{f_c} is the periodized sinc or Dirichlet kernel

$$D_{f_c}\left(t\right) := \sum_{k=-f_c}^{f_c} \exp\left(i2\pi kt\right) = \begin{cases} 1 & \text{if } t = 0 \\ \frac{\sin((2f_c+1)\pi t)}{(2f_c+1)\sin(\pi t)} & \text{otherwise .} \end{cases} \tag{25}$$

Just to clarify, in most applications that use the periodogram t would index the frequency domain, *not* the time domain, but we keep this notation for the sake of consistency. Also, f_c no longer has a physical meaning (beyond determining the number of time-domain samples) in contrast to spatial super-resolution where it represents the cut-off frequency of the sensing mechanism.

Computing the periodogram does not solve the super-resolution problem, it just allows to visualize the aliased spectrum corresponding to the data samples available in spectral super-resolution. If the line spectra of the signal are far apart, a straightforward way to estimate their location is to locate local maxima of P. The problem with this approach is that the side lobes corresponding to large blurred spikes may mask the presence of smaller spikes. As a result, the periodogram is not very useful if the spikes are not far enough from each other

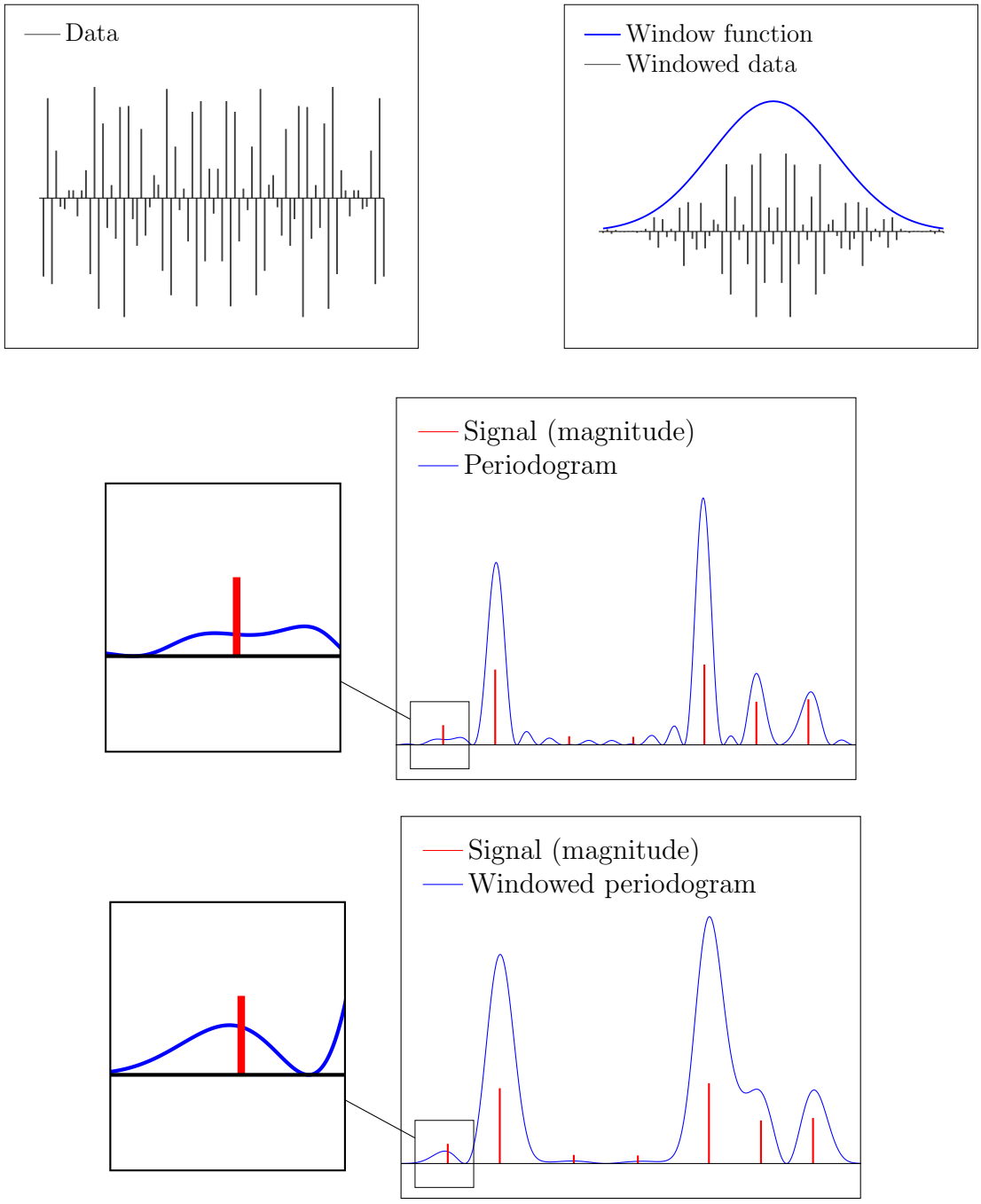


Figure 9: Example of data in spectral super-resolution generated according to (6) before (top left) and after applying a window function (top right). No noise is added to the data. Below we can see the periodogram (center) and windowed periodogram (bottom) computed from the data. The scaling of the periodograms is set so that both the large and small peaks can be seen on the plot.

or if their amplitudes differ substantially, *even if no noise is present in the data*. The image at the center of Figure 9 illustrates this: detecting some of the lower-amplitude spikes from the periodogram is impossible.

In order to alleviate this problem one can apply a window function $\widehat{w} \in \mathbb{C}^n$ to the data before computing the periodogram,

$$y_{\widehat{w}} = y \cdot \widehat{w}, \tag{26}$$

where $\cdot$ denotes pointwise multiplication. The windowed periodogram

$$P_{\widehat{w}}(f) = \mathcal{F}_n^* \, y_{\widehat{w}} \tag{27}$$

$$= \sum_{t_j \in T} c_j w \left(t - t_j \right), \tag{28}$$

where w denotes the inverse Fourier transform of the window function. Ideally, w should be as *spiky* as possible to make it easier to locate the support of the signal from the windowed periodogram. However, this is challenging due to the constraint that $\widehat{w}$ has finite support and hence w is a low-pass function.

In the image on the top right of Figure 9 we apply a Gaussian window to the data. To be more precise, we set $\widehat{w}$ to be a truncated Gaussian, so that w is equal to the convolution between a periodized sinc and a periodized Gaussian. The resulting periodogram, shown at the center of Figure 9, has much less *spectral leakage* from the largest signal components, due to the fact that the Gaussian window has lower side lobes than the periodized sinc. However, the latter is spikier at the origin, which allows to better distinguish neighboring spikes with similar amplitudes. In general, designing an adequate window implies finding a good tradeoff between the width of the main lobe and the height of the side lobes. We refer the reader to [6] for a detailed account of design considerations and types of window function. As discussed in Lecture Notes 4, windowing is also important when computing short-time Fourier transforms.

Figure 10 compares the result of applying the periodogram and the windowed periodogram to noisy data from two signals with different minimum separations. The signal-to-noise ratio is 20 dB (equivalently the ratio between the ℓ_2-norm of the signal and the noise is equal to 10). When the minimum separation is large the windowed periodogram provides a relatively good estimate of the number of point sources, although their locations do not exactly coincide with the local maxima for some of the spikes. Without windowing, the periodogram provides sharper localization but suffers from spurious spikes produced by the side lobes of the sinc function. When the minimum separation is small, it is impossible to super-resolve many of the spikes, which is not surprising: the problem becomes very ill-conditioned as discussed in Section 2.

To conclude, the periodogram is a useful method to obtain a general idea of the spectral structure of the signal and can provide insight as to the number of sources and their approximate location, especially if the data is processed with a suitable window function and the

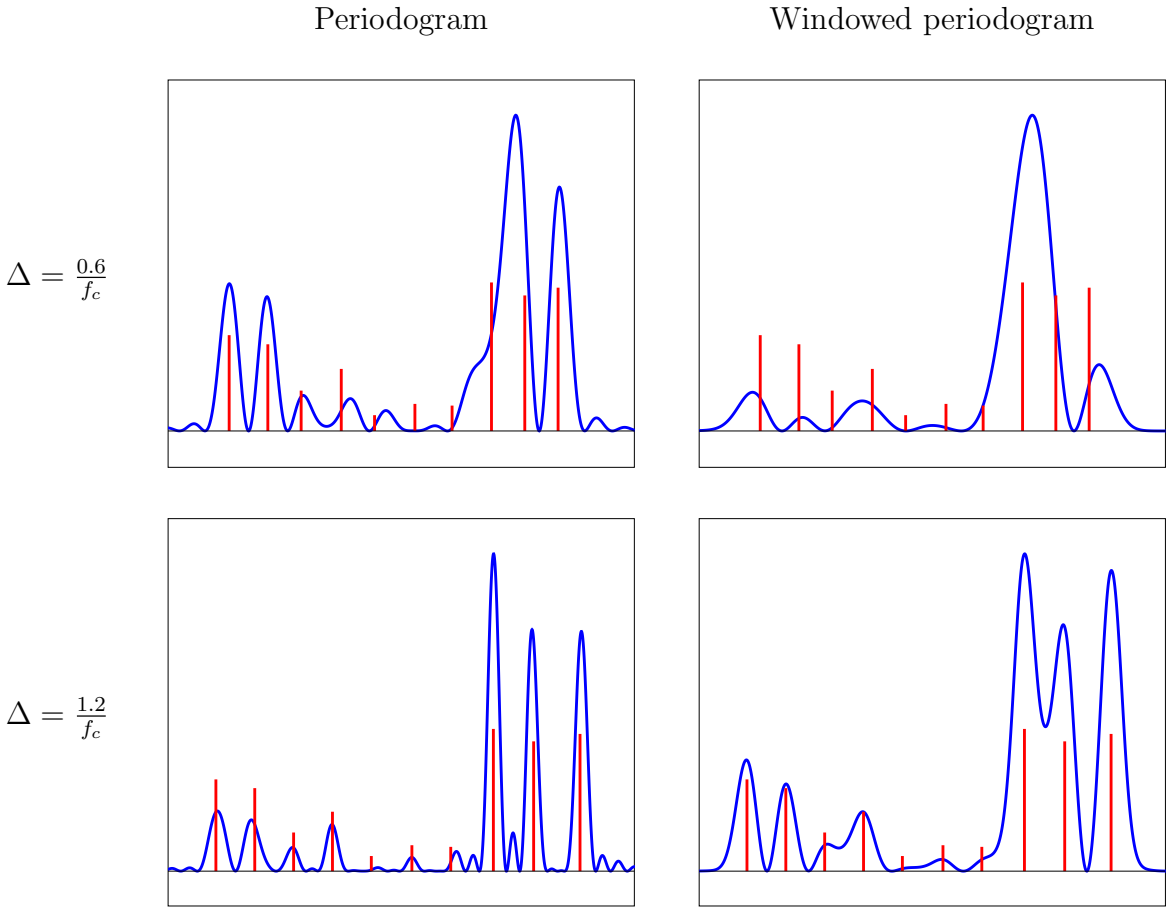


Figure 10: Periodogram (left) and windowed periodogram (right) for two signals with identical amplitudes but different minimum separations.

support of the signal is not too clustered together. However, the method does not provide a precise estimate of the support even in the absence of noise. This is the aim of the techniques described in the following sections.

3.2 Local fitting

If we have low-resolution measurements of the form (4) and the signal consists of a single blurred source then

$$x_{\mathrm{LR}}(t) := c_1 \phi(t - t_1). \tag{29}$$

In this case, it seems reasonable to estimate the location of the source by fitting the shifted PSF to the data, for instance, by minimizing a least-squares cost function. For a single source, performing a least-squares fit amounts to finding the shift that maximizes the inner product between the data and the PSF ϕ shifted by $\tilde{t}$, which is denoted by $\phi_{\tilde{t}}$. Let us assume that x_{LR} and $\phi_{\tilde{t}}$ are in $\mathcal{L}_2$ and that $||\phi_{\tilde{t}}||_{\mathcal{L}_2} = 1$. For a fixed $\tilde{t}$ the best coefficient is obtained by projecting x_{LR} onto $\phi_{\tilde{t}}$

$$\langle x_{\mathrm{LR}}, \phi_{\tilde{t}} \rangle = \arg\min_{\tilde{t}, \alpha \in \mathbb{C}} ||x_{\mathrm{LR}} - \alpha \, \phi_{\tilde{t}}||_2 . \tag{30}$$

As a result, choosing the first shift is equivalent to maximizing the inner product

$$\begin{aligned} t_{\mathrm{est}} &= \arg\min_{\tilde{t}} \min_{\alpha \in \mathbb{C}} ||x_{\mathrm{LR}} - \alpha \, \phi_{\tilde{t}}||_2 \\ &= \arg\min_{\tilde{t}} ||x_{\mathrm{LR}} - \langle x_{\mathrm{LR}}, \phi_{\tilde{t}} \rangle \ \phi_{\tilde{t}}||_2 \\ &= \arg\min_{\tilde{t}} -2 \left| \langle x_{\mathrm{LR}}, \phi_{\tilde{t}} \rangle \right|^2 \\ &= \arg\max_{\tilde{t}} \left| \langle x_{\mathrm{LR}}, \phi_{\tilde{t}} \rangle \right| . \end{aligned} \tag{31}$$

This procedure is known as *matched filtering* in the signal-processing literature.

When more than one point source is present, the overlap between neighboring blurred sources may be negligible if the sources are far enough with respect to the decay of the PSF. In this case, one can just fit the PSF locally to each of them. The approach is essentially equivalent to applying matching pursuit (Algorithm 3.2 in Lecture Notes 4) with a dictionary of shifted copies of ϕ. This method is quite popular in fluorescence microscopy. However, if there is interference between the blurred sources, then the problem becomes much more challenging. As a result, in fluorescence microscopy the data is usually measured at a rate that ensures a certain separation between the active fluorophores.

4 Parametric methods for spectral super-resolution

In this section we review parametric methods for spectral super-resolution based on Prony's method. These algorithms take the number of line spectra of the original signal as input and return an estimate of their location. As in our description of the periodogram, we use the variable t to denote frequency for the sake of consistency with other sections.

4.1 Prony's method

Prony's method solves the spectral super-resolution problem by encoding the position of s sources $T = \{t_1, \ldots, t_s\}$ as the zeros of a trigonometric polynomial of order s. In the absence of noise, it turns out that we can always find such a polynomial and achieve perfect spectral super-resolution by locating its roots.

We consider a signal x

$$x := \sum_{t_j \in T} c_j \delta_{t_j}, \tag{32}$$

and data that correspond to the n first Fourier coefficients of x

$$y_k := \widehat{x}\,(k)\,, \qquad 0 \leq k \leq n-1. \tag{33}$$

If we set $n := 2f_c + 1$, this model is essentially equivalent to (6).

The following lemma shows that for *any* x of the form (32) there exists a nonzero polynomial of order s which is exactly zero on the support of the x.

Lemma 4.1 (Existence of Prony polynomial)**.** *Assume that x is as in* (3) *and $|T| = s$, then*

$$P_{\text{prony}}(t) := \prod_{j=1}^{s} \left(1 - \exp\left(i2\pi(t - t_j)\right)\right) \tag{34}$$

$$= 1 + \sum_{l=1}^{s} v_l \exp\left(i2\pi l t\right), \quad v_0 := 1, \tag{35}$$

is a nonzero polynomial of order s with roots located exactly on T.

Proof. Since $v_0 = 1$ the polynomial is nonzero. By construction $P_{\text{prony}}(t_j) = 0$ for every $t_j \in T$. □

If we are able to compute such a polynomial, then finding its roots immediately reveals the support of x. The problem is how to compute P_{prony} from the available data y. Prony's

method consists of setting up a system of linear equations that only depends on y such that the solution is equal to the coefficients of P_{prony}.

Since P_{prony} is zero wherever x is nonzero, the inner product of x and P_{prony} is equal to zero,

$$\begin{aligned}
\langle P_{\text{prony}}, x \rangle &= \int_0^1 P_{\text{prony}}(t)\, \overline{x}(\mathrm{d}t) && (36)\\
&= \sum_{j=0}^{s} \overline{c_j}\, \exp\left(-i2\pi l t_j\right) \left(1 - \exp\left(i2\pi(t_j - t_j)\right)\right) && (37)\\
&= 0. && (38)
\end{aligned}$$

By Parseval's Theorem this inner product equals the inner product between the Fourier coefficients of x and P_{prony} which implies

$$\langle v, \widehat{x} \rangle = \langle P_{\text{prony}}, x \rangle = 0. \tag{39}$$

We can use this identity to obtain an equation tying the coefficient vector v to the data.

$$\begin{aligned}
0 = \langle v, \widehat{x} \rangle &= \sum_{k=-\infty}^{\infty} v_k\, \overline{\widehat{x}\,(k)} && (40)\\
&= \sum_{l=0}^{s} v_l\, \overline{\widehat{x}\,(k)} && (41)\\
&= \sum_{l=0}^{s} v_l\, \overline{y\,(k)} \qquad \text{if } s+1 \leq n. && (42)
\end{aligned}$$

In order to estimate v we need at least $s-1$ other equations. Consider taking the inner product between the coefficients of P_{prony} and $\widehat{x}$ shifted by k', which we denote by

$$\widehat{x}_{k'} := \widehat{x}\,(k + k')\,. \tag{43}$$

By basic Fourier identities, $\widehat{x}_{k'}$ is the Fourier representation of $\exp(2\pi k' t)\, x$. This measure has exactly the same support as x. As a result, we can repeat the same argument as above to conclude that

$$\begin{aligned}
0 &= \langle P_{\text{prony}}, \exp(2\pi k' t)\, x \rangle && (44)\\
&= \langle v, \widehat{x}_{k'} \rangle && (45)\\
&= \sum_{l=0}^{s} v_k\, \overline{y\,(k + k')} \qquad \text{if } s + k' \leq n - 1 && (46)\\
&= 0. && (47)
\end{aligned}$$

If we select k' to be between 0 and $s-1$ we obtain a system of s equations. This system only involves the data y and the unknown coefficients v as long as $n \geq 2s$.

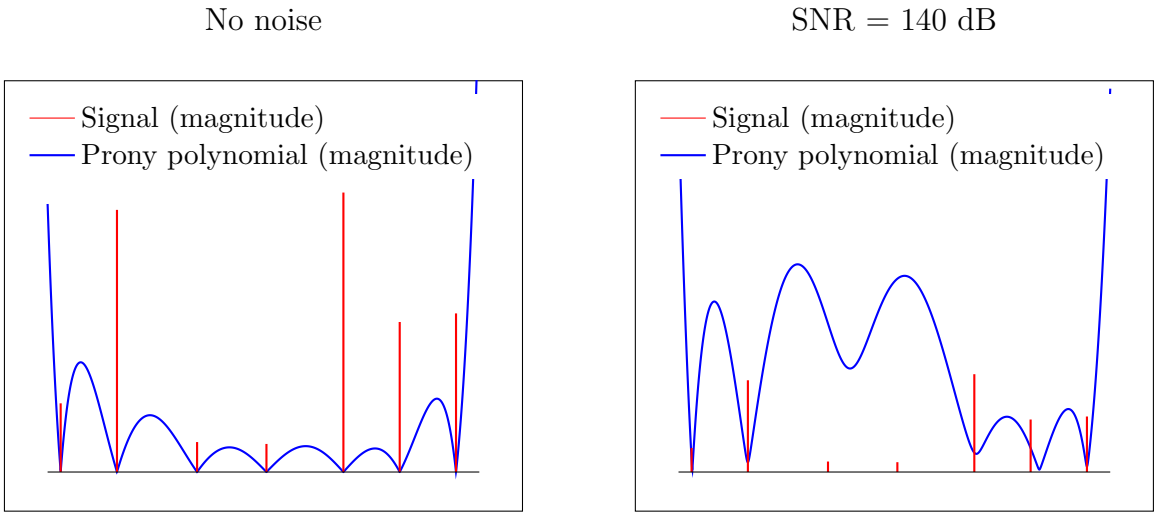


Figure 11: Prony polynomial applied on noiseless data (left). The image on the right shows the effect of adding a very small quantity of noise to the data. The roots of the polynomial no longer coincide with the support of the original signal. Note that the vertical axis is scaled differently in the two images.

Prony's method consists of solving this system and then decoding the support of the signal by rooting the corresponding polynomial. For simplicity we assume that $n = 2s$.

Algorithm 4.2 (Prony's method)**.** *The input to the algorithm is the cardinality of the support s and the data y, which are assumed to be of the form* (33)*.*

1. *Form the system of equations*

$$\begin{bmatrix} y_1 & y_2 & \cdots & y_s \\ y_2 & y_3 & \cdots & y_{s+1} \\ \cdots & \cdots & \cdots & \cdots \\ y_s & y_{s+1} & \cdots & y_{n-1} \end{bmatrix} \begin{bmatrix} \tilde{v}_1 \\ \tilde{v}_2 \\ \cdots \\ \tilde{v}_s \end{bmatrix} = - \begin{bmatrix} y_0 \\ y_1 \\ \cdots \\ y_{s-1} \end{bmatrix}. \tag{48}$$

2. *Solve the system to obtain $v_1, \ldots, v_s$ and set $v_0 = 1$.*
3. *Root the polynomial corresponding to $v_0, \ldots, v_s$ to obtain its s roots $z_1, \ldots, z_s$.*
4. *For every root on the unit circle $z_j = \exp(i2\pi\tau)$ include τ in the estimated support.*

This procedure is guaranteed to achieve exact recovery of the original signal.

Lemma 4.3. *In the absence of noise, the output of Prony's method is equal to the support of the original signal.*

Proof. The coefficients of the polynomial (35) are a feasible solution for the system of equations (48). In fact, they are the unique solution. To show this we compute the factorization

$$\begin{bmatrix} y_1 & y_2 & \cdots & y_s \\ y_2 & y_3 & \cdots & y_{s+1} \\ \cdots & \cdots & \cdots & \cdots \\ y_s & y_{s+1} & \cdots & y_{n-1} \end{bmatrix} = \tag{49}$$

$$\begin{bmatrix} e^{-i2\pi t_1} & e^{-i2\pi t_2} & \cdots & e^{-i2\pi t_s} \\ e^{-i2\pi 2t_1} & e^{-i2\pi 2t_2} & \cdots & e^{-i2\pi 2t_s} \\ \cdots & \cdots & \cdots & \cdots \\ e^{-i2\pi st_1} & e^{-i2\pi st_2} & \cdots & e^{-i2\pi st_s} \end{bmatrix} \begin{bmatrix} c_1 & 0 & \cdots & 0 \\ 0 & c_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & c_s \end{bmatrix} \begin{bmatrix} 1 & e^{-i2\pi t_1} & \cdots & e^{-i2\pi(s-1)t_1} \\ 1 & e^{-i2\pi t_2} & \cdots & e^{-i2\pi(s-1)t_2} \\ \cdots & \cdots & \cdots & \cdots \\ 1 & e^{-i2\pi t_s} & \cdots & e^{-i2\pi(s-1)t_s} \end{bmatrix}. \tag{50}$$

The diagonal matrix is full rank as long as all the coefficients c_j are nonzero, whereas the two remaining matrices are full rank by the following lemma, proved in Section A.1 of the appendix (set $z_j := \exp(-i2\pi t_j)$).

Lemma 4.4 (Vandermonde matrix)**.** *For any distinct set of s nonzero complex numbers $z_1, z_2, \ldots, z_s$ and any positive integers m_1, m_2, s such that $m_2 - m_1 + 1 \geq s$ the Vandermonde matrix*

$$\begin{bmatrix} z_1^{m_1} & z_2^{m_1} & \cdots & z_s^{m_1} \\ z_1^{m_1+1} & z_2^{m_1+1} & \cdots & z_s^{m_1+1} \\ z_1^{m_1+2} & z_2^{m_1+2} & \cdots & z_s^{m_1+2} \\ & & \cdots & \\ z_1^{m_2} & z_2^{m_2} & \cdots & z_s^{m_2} \end{bmatrix} \tag{51}$$

is full rank.

As a result, the matrix in (48) is full rank, so the system of equations has a unique solution equal to (35). This completes the proof, as rooting (35) obviously yields the support of the signal. □

Unfortunately, Prony's method as presented above cannot be applied to real data even if the signal-to-noise ratio is exceptionally high. The image on the left of Figure 11 shows how the Prony polynomial allows to super-resolve the support to very high accuracy from noiseless

data. However, on the right we see the result of applying the method to data that have a very small amount of noise (the ratio between the ℓ_2 norm of the noise and the noiseless data is around 10^{-8}!). The roots of the Prony polynomial are perturbed away from the points of the unit circle that correspond to the support, so that it is no longer possible to accurately locate the support of the signal. It is consequently necessary to adapt the method to deal with noisy data if there is to be any hope of applying it in any realistic scenario. The following sections discuss such extensions.

4.2 Subspace methods

In order to motivate subspace-based methods to tackle noisy data, we first describe an alternative interpretation of Prony's method.

Solving the system (48) is essentially a way to find a nonzero vector in the null space of $Y(s+1)^T$, where $Y(m)$ is defined for any integer m as the Hankel matrix

$$Y(m) := \begin{bmatrix} y_0 & y_1 & \cdots & y_{n-m} \\ y_1 & y_2 & \cdots & y_{n-m+1} \\ \cdots & \cdots & \cdots & \cdots \\ y_{m-1} & y_m & \cdots & y_{n-1} \end{bmatrix}. \tag{52}$$

The vector in the null space of $Y(s+1)^T$ corresponds to the coefficients of the Prony polynomial, which we can root to find the support of the signal.

Recall that in the absence of noise the data are of the form

$$\begin{aligned} y_k &:= \widehat{x}(k) && (53) \\ &= \sum_{t_j \in T} c_j e^{-i2\pi k t_j}, \qquad 0 \le k \le n-1. && (54) \end{aligned}$$

Consider the decomposition

$$\begin{aligned} Y(m) &= \begin{bmatrix} a_{0:m-1}(t_1) & a_{0:m-1}(t_2) & \cdots & a_{0:m-1}(t_s) \end{bmatrix} \begin{bmatrix} c_1 & 0 & \cdots & 0 \\ 0 & c_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & c_s \end{bmatrix} \begin{bmatrix} a_{0:n-m}(t_1)^T \\ a_{0:n-m}(t_2)^T \\ \cdots \\ a_{0:n-m}(t_s)^T \end{bmatrix} \\ &= A_{0:m-1}(T)\, C\, A_{0:m}(T)^T, && (55) \end{aligned}$$

where for $k > 0$ we define

$$a_{0:k}(t) := \begin{bmatrix} 1 \\ e^{-i2\pi t} \\ e^{-i2\pi 2t} \\ \cdots \\ e^{-i2\pi kt} \end{bmatrix}, \qquad A_{0:k}(T) := \begin{bmatrix} a_{0:k}(t_1) & a_{0:k}(t_2) & \cdots & a_{0:k}(t_s) \end{bmatrix}. \tag{56}$$

This decomposition suggests an alternative way of estimating the support T from $Y(m)$: finding sinusoidal atoms $a_{0:m-1}(t)$ that are in the column space of $Y(m+1)$. Lemma 4.5 below proves that the only atoms of this form that belong to the column space of $Y(m)$ are precisely $a_{0:m-1}(t_1)$, $a_{0:m-1}(t_2)$, ..., $a_{0:m-1}(t_s)$.

In order to make this procedure robust to noise, in practice we check what atoms are *close* to the column space of $Y(m)$. To quantify this we compute the orthogonal complement $\mathcal{N}$ of the column space of $Y(m)$ and construct the *pseudospectrum*

$$P_{\mathcal{N}}(t) = \log \frac{1}{|\mathcal{P}_{\mathcal{N}}(a_{0:m-1}(t))|^2}, \tag{57}$$

where $\mathcal{P}_{\mathcal{N}}$ denotes a projection onto $\mathcal{N}$. If an atom is almost orthogonal to $\mathcal{N}$ then $P_{\mathcal{N}}$ will have a very large value at that point.

In order to find the subspace $\mathcal{N}$, one possibility is computing the null space of the empirical covariance matrix

$$\Sigma(m) = \frac{1}{n-m+1} YY^* \tag{58}$$

$$= \frac{1}{n-m+1} \sum_{j=0}^{n-m} \begin{bmatrix} y_j \\ y_{j+1} \\ \cdots \\ y_{j+m-1} \end{bmatrix} \begin{bmatrix} \overline{y}_j & \overline{y}_{j+1} & \cdots & \overline{y}_{j+m-1} \end{bmatrix}. \tag{59}$$

The following lemma shows that the maxima of the pseudospectrum reveal the support of the signal in the noiseless case.

Lemma 4.5. *Let $\mathcal{N}$ be the null space of the empirical covariance matrix $\Sigma(m)$ for $m \geq s$. If the data are of the form* (53)*,*

$$P_{\mathcal{N}}(t_j) = \infty, \qquad \text{for } t_j \in T, \tag{60}$$

$$P_{\mathcal{N}}(t) < \infty, \qquad \text{for } t \notin T. \tag{61}$$

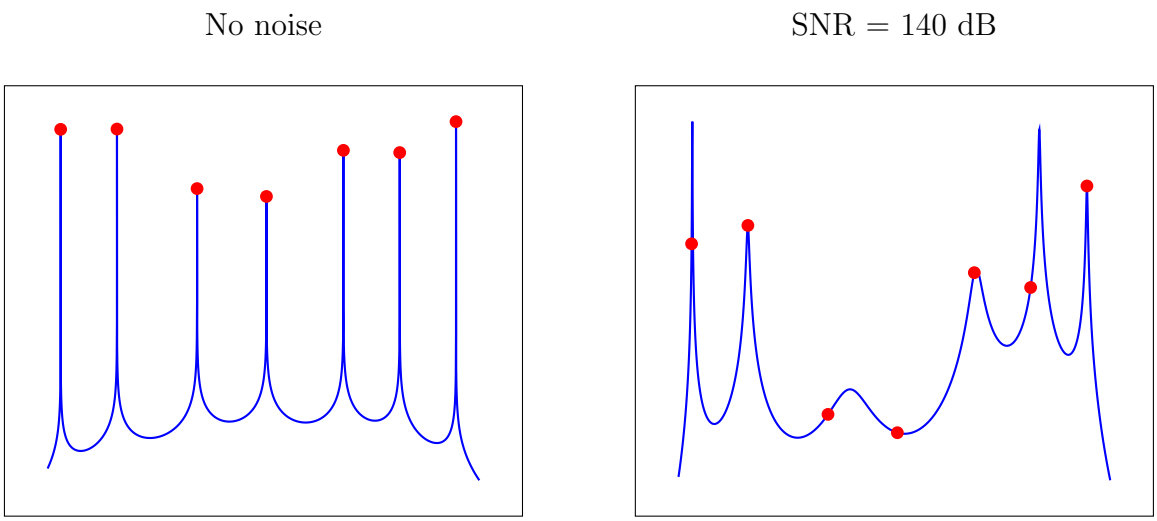


Figure 12: Pseudospectrum corresponding to the data used to construct the Prony polynomials in Figure 11. The true location of the support is marked with red circles.

Proof. By (55) the atoms $a_{0:m-1}(t_1)$, ..., $a_{0:m-1}(t_s)$ span the column space of $Y(m)$ and $\Sigma(m)$. As a result they are orthogonal to the null space $\mathcal{N}$ of the empirical covariance matrix, which proves (60).

We prove (61) by contradiction. The atoms $a_{0:m-1}(t_1)$, ..., $a_{0:m-1}(t_s)$ span the orthogonal complement to $\mathcal{N}$. As a result, if $a_{0:m-1}(t)$ is orthogonal to $\mathcal{N}$ for some t then $a_{0:m-1}(t)$ is in the span of $a_{0:m-1}(t_1)$, ..., $a_{0:m-1}(t_s)$. This would imply that $A^T_{0:m-1}(T \cup \{t\})$ is not full rank, which can only hold if $t \in T$ by Lemma 4.4. □

Figure 12 shows the pseudospectrum corresponding to the data used to construct the Prony polynomials in Figure 11 when $n = 2s - 1$. In the noiseless case, the pseudospectrum allows to locate the support T perfectly. Unfortunately, the locations of the local maxima are severely perturbed by even a very small amount of noise.

In order to obtain an estimate that is robust to noise, we need to use more data to estimate the support of the signal. The definition of the empirical covariance matrix $\Sigma(m)$ in (58) provides a way to do this, we can just average over more data. If we fix m and increase n, the column space of $\Sigma(m)$ remains the same, but the averaging process may cancel out the noise to some extent. This is the principle underlying the multiple-signal classification (MUSIC) [1,11].

Algorithm 4.6 (Multiple-signal classification (MUSIC))**.** *The input is the cardinality of the support s, the data y, which are assumed to be generated as in* (66) *and the value of the parameter $m \geq s$.*

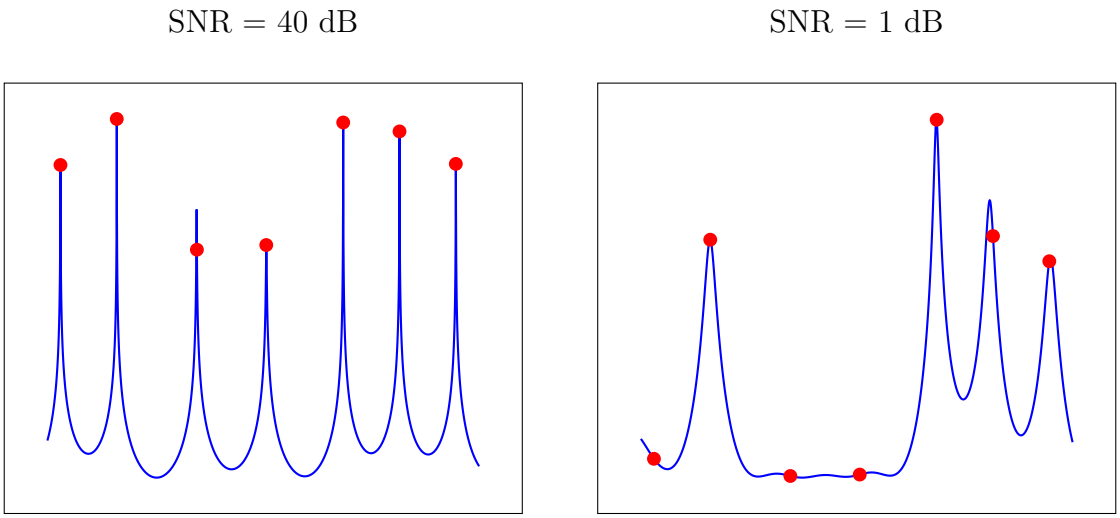


Figure 13: Pseudospectrum constructed by the MUSIC algorithm with $n = 81$ and $m = 30$ for the same signal used in Figure 11 and different noise levels. The true location of the support is marked with red circles.

1. *Build the empirical covariance matrix* $\Sigma(m)$ *defined in* (58).
2. *Compute the eigendecomposition of* $\Sigma(m)$ *to select the subspace* $\mathcal{N}$ *corresponding to the* $m - s$ *smallest eigenvalues.*
3. *Output an estimate of* s *estimated positions for the support by locating the* s *highest peaks of the pseudospectrum*

$$P_{\mathcal{N}}(t) = \log \frac{1}{\left|\mathcal{P}_{\mathcal{N}}\left(a_{0:m-1}\right)\right|^2}, \tag{62}$$

By Lemma 4.5, in the absence of noise MUSIC allows to estimate the support of the signal perfectly. When additive Gaussian noise is present in the data, MUSIC is much more robust than Prony's method. Figure 13 shows the result of applying MUSIC algorithm with $n = 81$ and $m = 30$ to the same data used in Figure 11 and different noise levels. The method is able to locate the support of the signal at a noise level of 40 dB. At 1 dB the pseudospectrum does not detect the smaller spikes (the true magnitudes are shown in Figure 11), but the estimate for the rest is still rather accurate.

In order to provide a theoretical justification of why MUSIC is stable we study the method in an asymptotic regime where the two following assumptions on the signal and the noise are met:

- **Assumption 1:** The original signal,

$$x = \sum_{t_j \in T} c_j \delta_{t_j} = \sum_{t_j \in T} |c_j| \exp(i\phi_j) \delta_{t_j}, \tag{63}$$

follows a probabilistic model where the phases ϕ_j are independent and uniformly distributed in the interval $[0, 2\pi]$, whereas the amplitudes are arbitrary and deterministic. Note that this implies that the expectation of x is equal to zero and that the covariance matrix equals

$$\mathrm{E}[cc^*] = D_c := \begin{bmatrix} |c_1|^2 & 0 & \cdots & 0 \\ 0 & |c_2|^2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & |c_s|^2 \end{bmatrix}. \tag{64}$$

- **Assumption 2:** The measurements are corrupted by white Gaussian noise with zero mean and standard deviation σ, which is independent from the signal. At time k we measure

$$y_k := \int_0^1 \exp(-i2\pi kt)\, x(\mathrm{d}t) + z_k \tag{65}$$

$$= \sum_{t_j \in T} c_j \exp(-i2\pi k t_j) + z_k, \tag{66}$$

where $z \sim N(0, \sigma^2 I)$ is a zero-mean Gaussian random vector with covariance matrix $\sigma^2 I$.

Under these assumptions, a sequence of m measurements $y_0, y_1, \ldots, y_{m-1}$ is a random vector of the form

$$\tilde{y} = A_{0:m-1}(T)\ c + z, \tag{67}$$

The following proposition, proved in Section A.2 of the appendix, characterizes the covariance matrix of $\tilde{y}$.

Proposition 4.7. *Let $\tilde{y}$ be an m dimensional vector of data satisfying Assumptions 1 and 2. The eigendecomposition of the covariance matrix of $\tilde{y}$ is equal to*

$$\mathrm{E}[\tilde{y}\tilde{y}^*] = \begin{bmatrix} U_{\mathcal{S}} & U_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} \Lambda + \sigma^2 I_s & 0 \\ 0 & \sigma^2 I_{n-s} \end{bmatrix} \begin{bmatrix} U_{\mathcal{S}}^* \\ U_{\mathcal{N}}^* \end{bmatrix}. \tag{68}$$

The eigenvectors are divided into two unitary matrices.

- *$U_{\mathcal{S}} \in \mathbb{C}^{m \times s}$ contains an orthonormal basis of the* signal subspace *which corresponds to the span of $a_{0:m-1}(t_1)$, …, $a_{0:m-1}(t_s)$.*

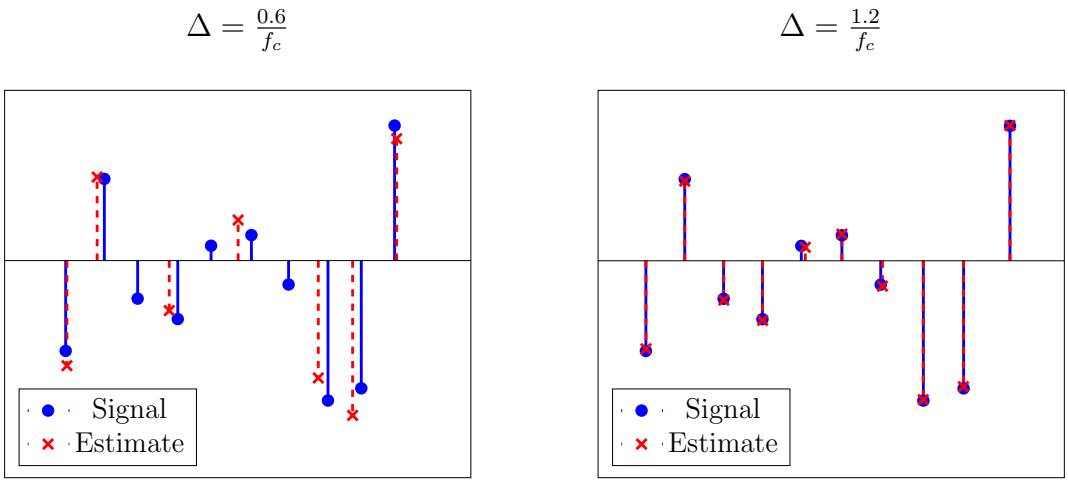


Figure 14: Result of applying MUSIC to data with an SNR of 20 dB for two different minimum separations. In both cases, $f_c = 40$ (so $n = 81$) and $m = 40$.

- $U_{\mathcal{N}} \in \mathbb{C}^{m \times (m-s)}$ *is a unitary matrix spanning the* noise subspace*, which is the orthogonal complement of the signal subspace.*

In addition, all eigenvalues are positive: $\sigma^2 I_k \in \mathbb{C}^{k \times k}$ *is a diagonal matrix with diagonal entries equal to* σ^2 *and* Λ *is another diagonal matrix with positive entries.*

This proposition provides a rather intuitive interpretation of the MUSIC algorithm. The eigendecomposition of the covariance matrix of the data allows to estimate a signal subspace and a noise subspace. As a result, the term *subspace methods* is often used to describe MUSIC and related algorithms. Computing the pseudospectrum from these subspaces allows us to locate the support of the signal.

In practice, we approximate the covariance matrix using the empirical covariance matrix $\Sigma(m)$ defined in (58). Asymptotically, if we fix s and m and let $n \to \infty$, $\Sigma(m)$ converges to the true covariance matrix (see Section 4.9.1 in [13]). However that this does *not* necessarily imply that MUSIC will allow to find the support! To ensure that we can actually identify the noise subspace correctly, the eigenvalues in Λ must all be large with respect to the variance of the noise σ^2. In the case of signals that have a small separation (with respect to n), some of these eigenvalues may be small due to the correlation between the atoms $a_{0:m-1}(t_1)$, $\ldots$, $a_{0:m-1}(t_s)$. This is not surprising because at small separations the inverse problem is ill posed, as explained in Section 2. Figure 14 shows the result of applying MUSIC to data with the same noise level for two different minimum separations. When the minimum separation is too small the method fails to detect some of the smaller spikes.

Figure 15 compares the performance of MUSIC for different noise levels and different values of the parameter m. On the left column, we see the decay of the eigenvalues of the empirical

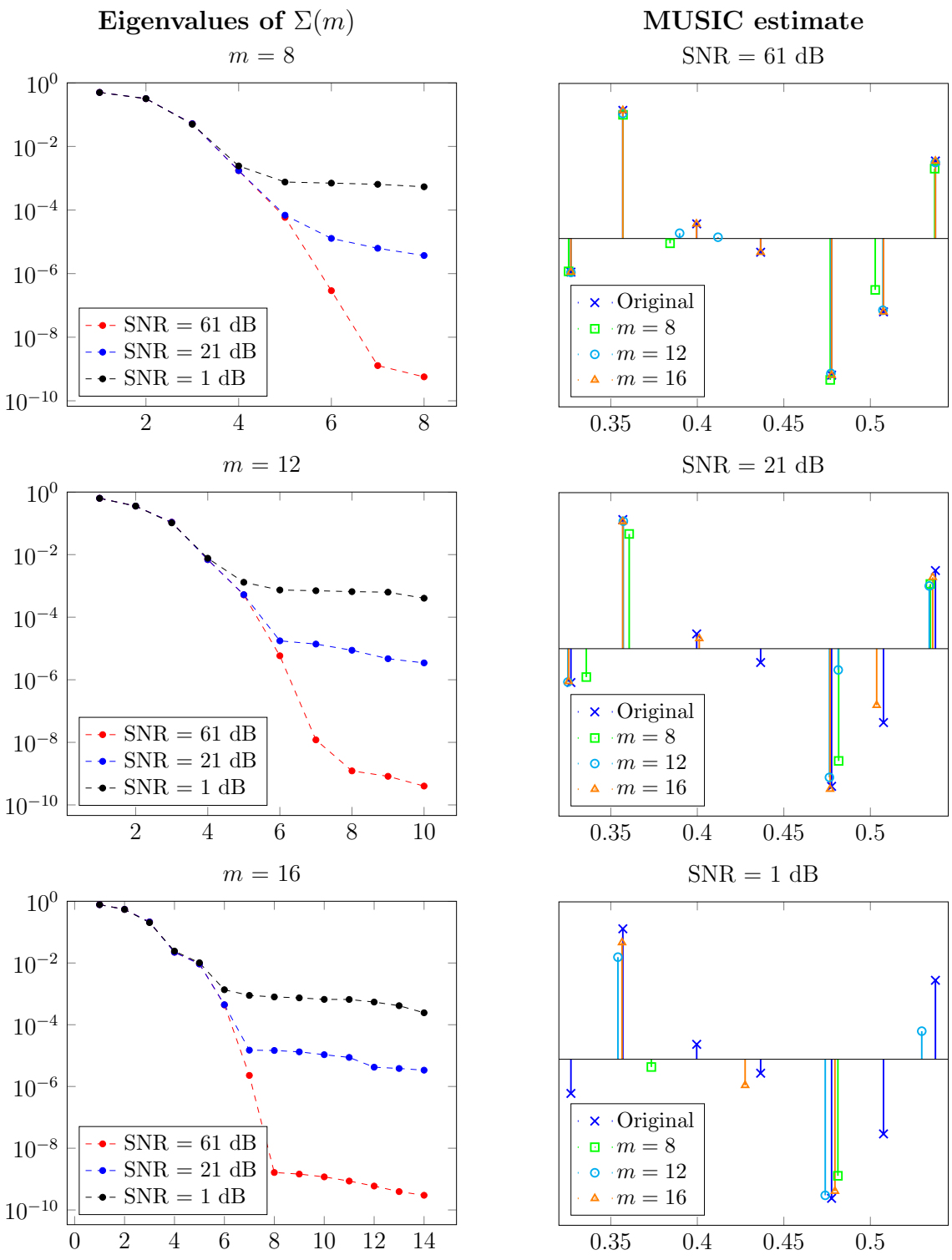


Figure 15: Eigenvalues of the empirical covariance matrix $\Sigma(m)$ used by MUSIC (left) and corresponding estimates (right) for different values of the parameter m and of the SNR. The cardinality of the true support is $s = 7$.

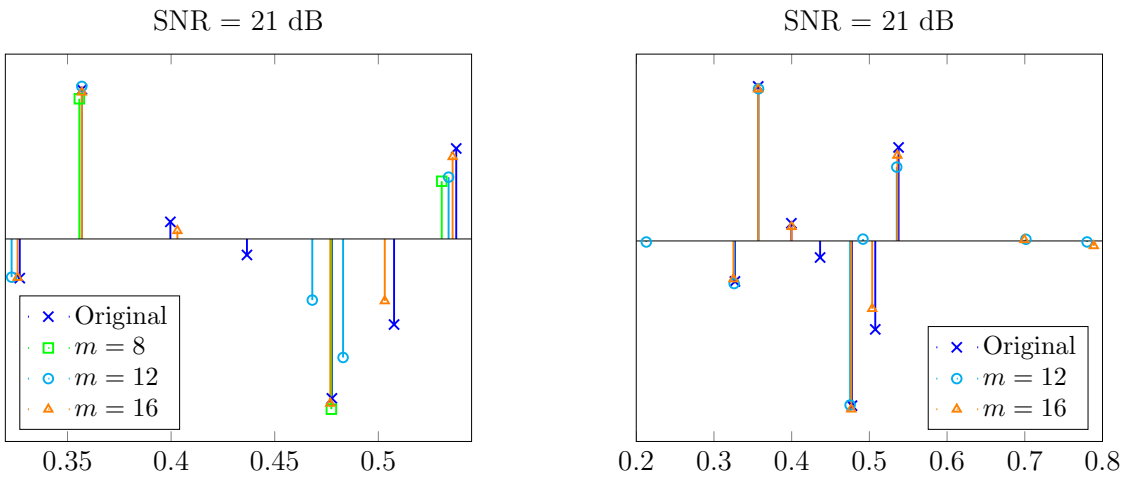


Figure 16: Line-spectra estimates obtained by Root MUSIC when the estimated number of sources is equal to $s - 1$ (left) and $s + 1$ (right) for the same data as in Figure 15.

covariance matrix. At high signal-to-noise ratios (SNR) there is a clear transition between the eigenvalues corresponding to the signal subspace (in this case $s = 7$) and the noise subspace, but this is no longer necessarily the case when the noise is increased. On the right column, we see the performance of the algorithm for different values of the SNR and the parameter m. At relatively high SNRs MUSIC is an effective algorithm as long as the assumptions on the signal (random phases), noise (Gaussian) and measurement model (equispaced time samples) are satisfied. In Figure 16 we show the result of running the algorithm for the wrong value of the parameter s. If the value is not too different to s and the SNR not too low, the method is still capable of approximately locating the support.

4.3 Matrix-pencil methods

In this section we describe an alternative approach to perform spectral super-resolution. In the previous section, we saw that MUSIC exploits the fact that in the absence of noise the matrix

$$Y_0 = \begin{bmatrix} y_0 & y_1 & \cdots & y_{n-m} \\ y_1 & y_2 & \cdots & y_{n-m+1} \\ \cdots & \cdots & \cdots & \cdots \\ y_{m-1} & y_m & \cdots & y_{n-1} \end{bmatrix} \tag{69}$$

has the rank-s factorization

$$Y_0 = A_{0:m-1}(T)\; C\; A_{0:n-m}(T)^T \tag{70}$$

$$= \sum_{t_j \in T} c_j\, a_{0:m-1}(t_j)\; a_{0:n-m}(t_j)^T \tag{71}$$

by computing the eigendecomposition of the empirical covariance matrix. Recall that

$$A_{m_1:m_2}(T) := \begin{bmatrix} a_{m_1:m_2}(t_1) & a_{m_1:m_2}(t_2) & \cdots & a_{m_1:m_2}(t_s) \end{bmatrix}. \tag{72}$$

Matrix-pencil methods make use of this factorization in a different way.

Let us begin by defining a matrix pencil and its rank-reducing values.

Definition 4.8 (Matrix pencil)**.** *The matrix pencil of two matrices M_1 and M_2 is the matrix-valued function*

$$L_{M_1,M_2}(\mu) := M_2 - \mu M_1 \tag{73}$$

where $\mu \in \mathbb{C}$.

The set of rank-reducing values $\mathcal{R}$ of a matrix pencil satisfy

$$\operatorname{rank}(L_{M_1,M_2}(\mu)) = \operatorname{rank}(L_{M_1,M_2}(\mu_j)) + 1 \qquad \text{for all } \mu_j \in \mathcal{R} \text{ and } \mu \notin \mathcal{R}. \tag{74}$$

Now consider the matrix

$$Y_1 = \begin{bmatrix} y_1 & y_2 & \cdots & y_{n-m+1} \\ y_2 & y_3 & \cdots & y_{n-m+2} \\ \cdots & \cdots & \cdots & \cdots \\ y_m & y_{m+1} & \cdots & y_n \end{bmatrix} \tag{75}$$

which can be written as

$$Y_1 = A_{1:m}(T)\; C\; A_{0:n-m}(T)^T \tag{76}$$

$$= \sum_{t_j \in T} c_j\, a_{1:m}(t_j)\; a_{0:n-m}(t_j)^T. \tag{77}$$

The insight behind matrix-pencil methods for spectral super-resolution is that $\exp(i2\pi t_1)$, $\exp(i2\pi t_2)$, $\ldots$, $\exp(i2\pi t_s)$ are all rank-reducing values of the matrix pencil L_{Y_0,Y_1}.

Lemma 4.9. *$\exp(i2\pi\tau)$ is a rank-reducing value of L_{Y_0,Y_1} if and only if $\tau \in T$.*

This lemma, proved in Section A.3 of the appendix implies that we can estimate the support of the signal by finding the rank-reducing numbers of the matrix pencil L_{Y_0,Y_1}. The following lemma, proved in Section A.4 of the appendix, shows how to compute them.

Lemma 4.10. *Let $Y_0 = U_0 \, \Sigma_0 \, V_0^*$ be the singular-value decomposition of Y_0. The s eigenvalues of the matrix*

$$M = V_0 \, \Sigma_0^{-1} \, U_0^* \, Y_1 \tag{78}$$

are equal to $\exp(i2\pi t_j)$ for $1 \leq j \leq s$.

The matrix-pencil algorithm [7] estimates the rank-reducing values from the noisy data, hoping that the noise will not disrupt the eigenvalues of M too much.

Algorithm 4.11 (Matrix-pencil method)**.** *The input is the cardinality of the support s, the data y, which are assumed to be generated as in* (66) *and the value of the parameter m.*

1. *Build the matrices Y_0 and Y_1.*
2. *Compute the singular-value decomposition of $Y_0 = U_0 \, \Sigma_0 \, V_0^*$.*
3. *Compute the s largest eigenvalues $\lambda_1, \, \lambda_2, \, \ldots, \, \lambda_s$ of the matrix*

$$M = V_0 \, \Sigma_0^{-1} \, U_0^* \, Y_1. \tag{79}$$

4. *Output τ such that $\lambda_j = \exp(i2\pi\tau)$ for $1 \leq j \leq s$.*

For more information we refer the interested reader to Chapter 4 of [13], where it is reported that matrix-pencil algorithms exhibit a similar performance to subspace methods. For theoretical guarantees under a minimum-separation condition see [9].

5 Super-resolution via convex programming

In this section we describe how to perform super-resolution by penalizing a sparsity-inducing norm. This is reminiscent of compressed sensing, which consists of estimating sparse signals from randomized measurements via ℓ_1-norm minimization. As we have explained in the previous sections, in super-resolution problem the signal is of the form

$$x := \sum_{t_j \in T} c_j \delta_{t_j}, \tag{80}$$

and the measurements are given by

$$y = \mathcal{F}_n \, x. \tag{81}$$

Since the signal can be supported anywhere in the unit interval, we cannot use the ℓ_1 norm to induce sparsity, as it is only defined for finite-dimensional vectors. In order to promote

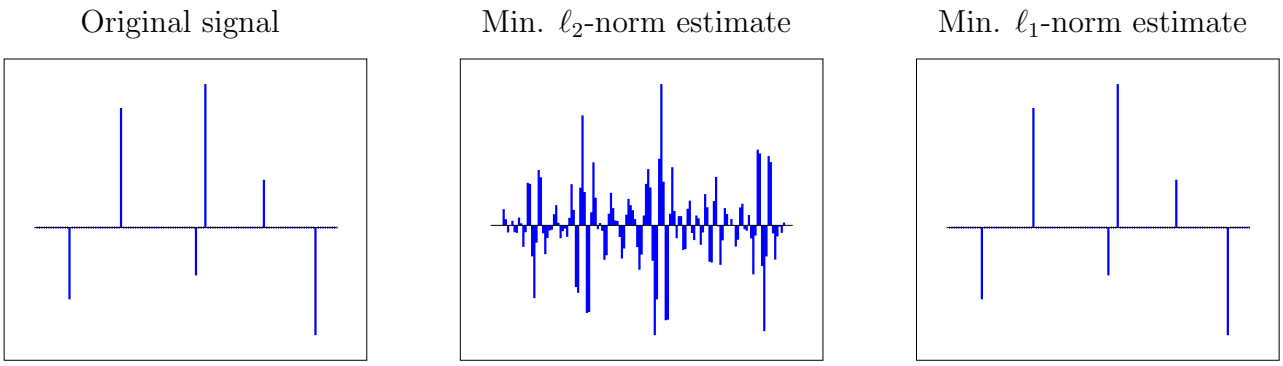


Figure 17: Results of applying ℓ_2 and ℓ_1-norm minimization to the inverse problem described in Figure 5. ℓ_1-norm minimization achieves exact recovery.

sparsity within a continuous interval, we resort to a continuous counterpart of the ℓ_1 norm: the total-variation (TV) norm. The total-variation norm of a measure x is defined as

$$||x||_{\text{TV}} = \sup_{||f||_\infty \leq 1, f \in C(\mathbb{T})} \int_{\mathbb{T}} \overline{f(t)} x(\mathrm{d}t). \tag{82}$$

For a superposition of Dirac deltas $\sum_j a_j \delta_{t_j}$, this norm is equal to the ℓ_1 norm of the coefficients, i.e. $||x||_{\text{TV}} = \sum_j |a_j|$. The term total-variation norm is somewhat unfortunate because *total variation* may also refer to the ℓ_1 norm of the discontinuities of a piecewise-constant function, as we have seen in previous lectures.

Super-resolution via TV-norm minimization requires solving the optimization problem

$$\begin{aligned} &\text{minimize} && ||\tilde{x}||_{\text{TV}} && (83)\\ &\text{subject to} && \mathcal{F}_n \tilde{x} = y, && (84) \end{aligned}$$

where the minimization is carried out over the set of all finite complex measures $\tilde{x}$ supported on $[0, 1]$. This infinite-dimensional convex problem can be solved by recasting it into a finite-dimensional problem involving matrices (more specifically a semidefinite program), as we will see later on in the course.

If the support of the signal is restricted to lie on a grid, problem 83 is equivalent to ℓ_1-norm minimization, exactly as in compressed sensing. Figure 17 shows the result of applying this algorithm to the reflection-seismography inverse problem described in Figure 5, where ℓ_1-norm minimization achieves exact recovery. In fact, it was geophysicists working in this area who first suggested using ℓ_1-norm minimization to tackle underdetermined linear inverse problems. Figure 18 shows results on real seismography data from a paper in 1979 [14]. These results spurred the interest of statisticians like David Donoho, who would eventually spearhead the use of ℓ_1-norm minimization in other areas of signal processing and compressed sensing.

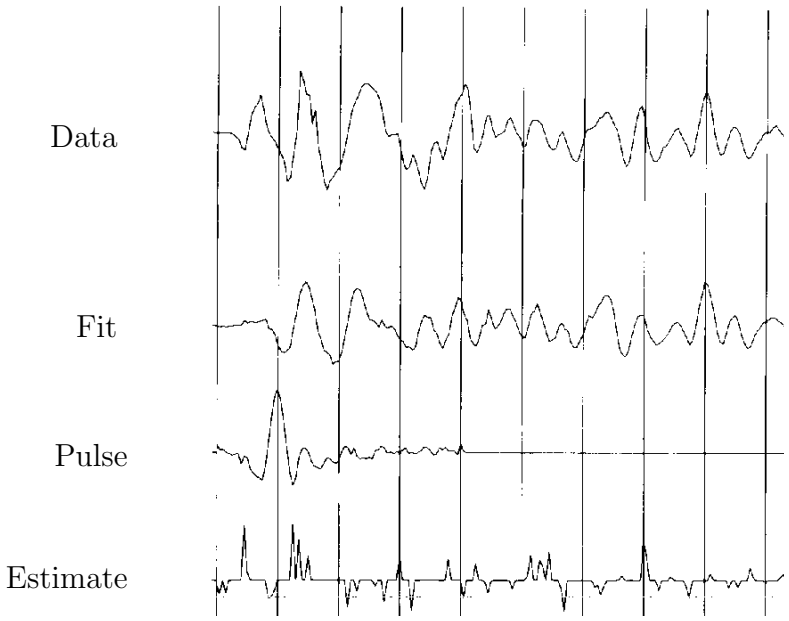


Figure 18: Result of applying ℓ_1-norm minimization to real seismography data from a paper in 1979 [14].

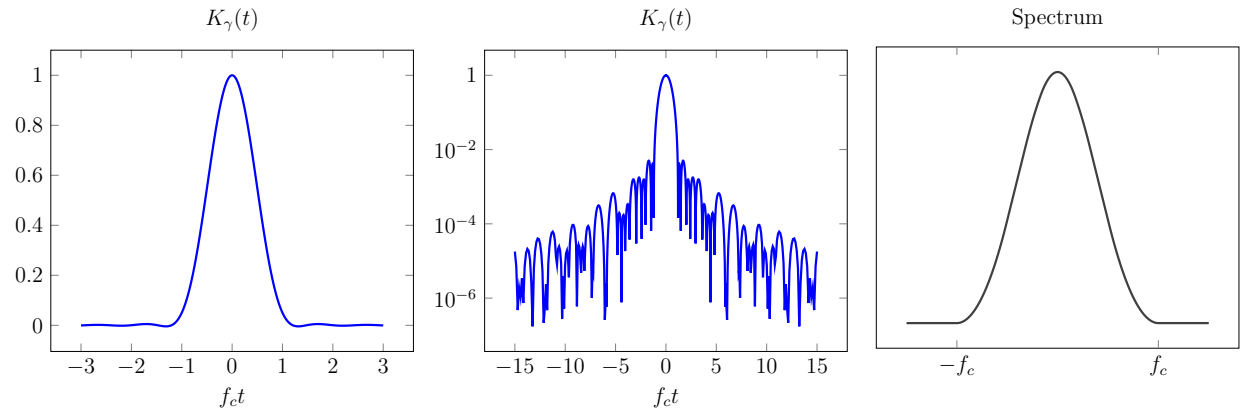


Figure 19: Interpolating kernel K used in the proof of Theorem 5.1 (left), along with its asymptotic decay (center) and its low-pass spectrum (right).

5.1 Exact recovery

The following theorem establishes that TV-norm minimization succeeds for signals that have a minimum separation of $1.26/f_c$, very near the limit of $1/f_c$ at which the problem becomes ill posed.

Theorem 5.1 (Total-variation minimization)**.** *Let T be the support of a signal x. If the minimum separation obeys*

$$\Delta(T) \geq 1.26\,/f_c \tag{85}$$

then x is the unique solution to (83)*.*

The proof of this theorem is based on the construction of a dual certificate for the TV norm. To simplify matters, while still conveying the main ideas, let us assume that the signal lies on a grid and we are minimizing the ℓ_1 norm. In that case, as we saw in the lecture on compressed sensing, a dual certificate is a vector v such that $q = \mathcal{F}_n^* v$ interpolates the sign pattern of the signal on its support and has magnitude smaller than one on its off-support. In compressed sensing q is a superposition of random vectors, whereas in super-resolution q is a superposition of low-pass trigonometric polynomials (the rows of the linear operator $\mathcal{F}_n$). This difference is crucial: if the support of the signal is cluttered together and the sign of its coefficients varies rapidly, it may not be possible to achieve the interpolation with a bounded polynomial.

We will sketch the proof of Theorem 5.1, which can be found in [2, 4]. The construction of dual certificates for compressed sensing typically relies on concentration arguments from probability theory. However there is nothing random about Theorem 5.1. We resort to a different approach: interpolating the sign of the signal using a low-pass kernel K to construct

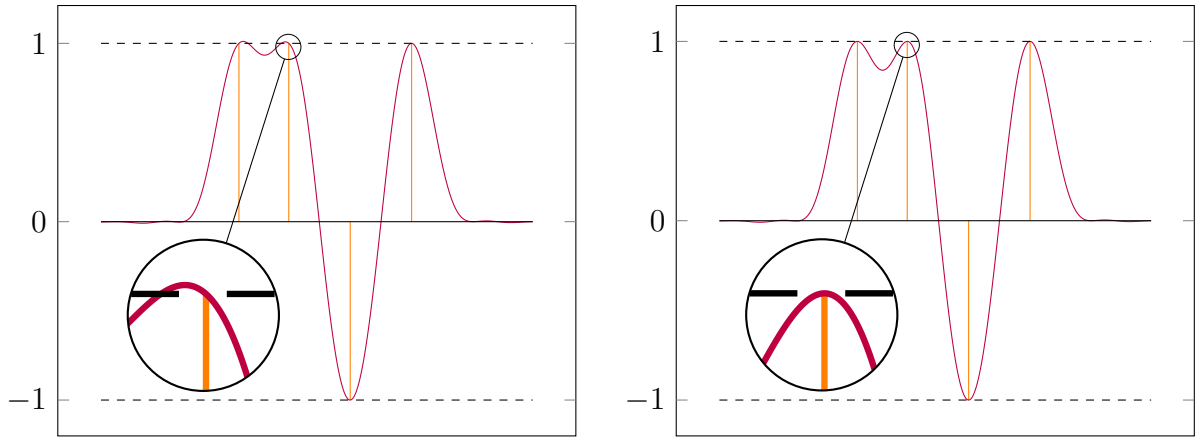


Figure 20: Dual-polynomial if we only use K to interpolate the sign pattern (left), as opposed to also incorporating its derivative into the construction and forcing the derivative of the polynomial to be zero at the interpolation points (right).

q directly. Since any shift of a low-pass kernel is low pass, this means that $q = \mathcal{F}_n^* v$ for some $v \in \mathbb{C}^n$. Figure 19 shows the kernel and its spectrum.

Interpolating the sign pattern with an adequately selected kernel is not sufficient to produce a valid construction. The reason is that the magnitude of the polynomial tends to exceed one near the elements of the support T. This can be avoided, however, by forcing the derivative of the kernel to be zero at those points. As a result, the magnitude of the polynomial has a local extremum at the interpolation point and its magnitude remains below one for a sufficiently large minimum separation, as illustrated by Figure 20. In order to enforce the extra constraint on the derivative of q, we need more degrees of freedom in the construction. For this purpose, we incorporate the derivative of the kernel, so that the q is of the form

$$q(t) = \sum_{t_j \in T} \alpha_j K_\gamma(t - t_j) + \beta_j K_\gamma^{'}(t - t_j), \tag{86}$$

where $\alpha, \beta \in \mathbb{C}^{|T|}$ are coefficient sequences satisfying

$$q(t_k) = \sum_{t_j \in T} \alpha_j K_\gamma\left(t_k - t_j\right) + \beta_j K_\gamma^{'}\left(t_k - t_j\right) = v_j, \quad t_k \in T, \tag{87}$$

$$q_R^{'}(t_k) + i q_I^{'}(t_k) = \sum_{t_j \in T} \alpha_j K_\gamma^{'}\left(t_k - t_j\right) + \beta_j K_\gamma^{''}\left(t_k - t_j\right) = 0, \quad t_k \in T, \tag{88}$$

and q_R and q_I are the real and imaginary parts of q. Since the derivative of a low-pass function is still low pass, q is in the row space of the linear operator $\mathcal{F}_c$. All is left is to show that the system of equations has a solution and that the magnitude of q is indeed bounded on the complement of T. We refer the interested reader to [2, 4] for the gory details.

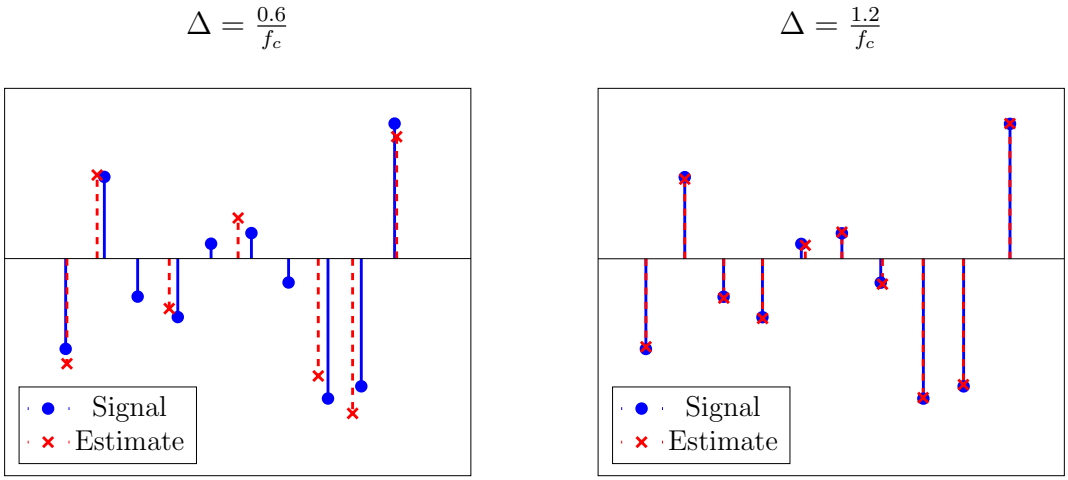


Figure 21: Result of applying TV-norm minimization to super-resolve data with an SNR of 20 dB for two different minimum separations. In both cases, $f_c = 40$.

5.2 Super-resolution from noisy data

Consider the additive-noise model

$$y = \mathcal{F}_n x + z \tag{89}$$

where z is a noise term. In order to account for the perturbation, we can adapt Problem (83) by using an inequality constraint to quantify the uncertainty,

$$\min_{\tilde{x}} ||\tilde{x}||_{\text{TV}} \quad \text{subject to} \quad ||\mathcal{F}_n \tilde{x} - y||_2^2 \leq \delta, \tag{90}$$

where δ is an estimate of the noise level. Alternatively, we could also consider a Lagrangian formulation of the form

$$\min_{\tilde{x}} ||\tilde{x}||_{\text{TV}} + \gamma \, ||\mathcal{F}_n \tilde{x} - y||_2^2 \,, \tag{91}$$

where the regularization parameter $\gamma > 0$ governs the tradeoff between data fidelity and the sparsity of the estimate. Figure 21 shows the result of applying TV-norm minimization to super-resolve data with an SNR of 20 dB for two different minimum separations. For a theoretical analysis of the performance of Problem 90 see [3].

References

For more information on spectral super-resolution using the periodogram, subspace methods and algorithms based on matrix pencil we refer the reader to [13] and references therein.

[1] G. Bienvenu. Influence of the spatial coherence of the background noise on high resolution passive methods. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 306 – 309, 1979.

[2] E. J. Candès and C. Fernandez-Granda. Towards a mathematical theory of super-resolution. *Communications on Pure and Applied Mathematics*, 67(6):906–956.

[3] E. J. Candès and C. Fernandez-Granda. Super-resolution from noisy data. *Journal of Fourier Analysis and Applications*, 19(6):1229–1254, 2013.

[4] C. Fernandez-Granda. Super-resolution of point sources via convex programming. *Information and Inference*, 2016.

[5] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002.

[6] F. Harris. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1):51 – 83, 1978.

[7] Y. Hua and T. K. Sarkar. Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 38(5):814–824, 1990.

[8] J. Lindberg. Mathematical concepts of optical superresolution. *Journal of Optics*, 14(8):083001, 2012.

[9] A. Moitra. Super-resolution, extremal functions and the condition number of Vandermonde matrices. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, 2015.

[10] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine*, 20(3):21 – 36, 2003.

[11] R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3):276 – 280, 1986.

[12] D. Slepian. Prolate spheroidal wave functions, Fourier analysis, and uncertainty. V - The discrete case. *Bell System Technical Journal*, 57:1371–1430, 1978.

[13] P. Stoica and R. L. Moses. *Spectral Analysis of Signals*. Prentice Hall, 2005.

[14] H. L. Taylor, S. C. Banks, and J. F. McCoy. Deconvolution with the ℓ_1 norm. *Geophysics*, 44(1):39–52, 1979.

A Proofs

A.1 Proof of Lemma 4.4

Let us define

$$Z := \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_s \\ z_1^2 & z_2^2 & \cdots & z_s^2 \\ & & \cdots & \\ z_1^{s-1} & z_2^{s-1} & \cdots & z_s^{s-1} \end{bmatrix}. \tag{92}$$

The determinant of the first s rows of our matrix of interest is equal to

$$|Z| \prod_{1 \le i \le s} z_i = \prod_{1 \le j < k \le s} (z_j - z_k) \prod_{1 \le i \le s} z_i \neq 0 \tag{93}$$

This implies that the first s rows are linearly independent and consequently that the whole matrix is full rank.

A.2 Proof of Proposition 4.7

Due to the assumptions,

$$\begin{aligned} \mathrm{E}\left[yy^*\right] &= \mathrm{E}\left[A_{0:m-1} cc^* A_{0:m-1}^* + A_{0:m-1} cz^* + zc^* A_{0:m-1}^* + zz^*\right] && (94) \\ &= A_{0:m-1} \mathrm{E}\left[cc^*\right] A_{0:m-1}^* + A_{0:m-1} \mathrm{E}\left[c\right] \mathrm{E}\left[z^*\right] + \mathrm{E}\left[z\right] \mathrm{E}\left[c^*\right] A_{0:m-1}^* + \mathrm{E}\left[zz^*\right] && (95) \\ &= A_{0:m-1} D_c A_{0:m-1}^* + \sigma^2 I. && (96) \end{aligned}$$

We begin by writing the full eigendecomposition of $A_{0:m-1} D_c A_{0:m-1}^*$, which is a positive semidefinite symmetric matrix of rank s. By the spectral theorem, this matrix has a singular value decomposition of the form

$$A_{0:m-1} D_c A_{0:m-1}^* = \begin{bmatrix} U_{\mathcal{S}} & U_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{\mathcal{S}}^* \\ U_{\mathcal{N}}^* \end{bmatrix}, \tag{97}$$

where $U_{\mathcal{S}}$ and $U_{\mathcal{N}}$ are as defined in the statement of the proposition. Λ is a diagonal matrix with positive entries, because $A_{0:m-1} D_c A_{0:m-1}^*$ is positive semidefinite. Indeed if we define

$\sqrt{D_c}$, a diagonal matrix with $|c|$ in the diagonal, we have that for any nonzero vector u

$$u^* A_{0:m-1} D_c A_{0:m-1}^* u = u^* A_{0:m-1} \sqrt{D_c} \sqrt{D_c} A_{0:m-1}^* u \tag{98}$$

$$= \left|\left| \sqrt{D_c} A_{0:m-1}^* u \right|\right|_2^2 \tag{99}$$

$$> 0, \tag{100}$$

where the inequality is strict because $\sqrt{D_c} A_{0:m-1}^*$ is full rank.

To complete the proof, we decompose the identity matrix using $U_{\mathcal{S}}$ and $U_{\mathcal{N}}$ to obtain

$$\mathrm{E}\left[yy^*\right] = A_{0:m-1} D_c A_{0:m-1}^* + \sigma^2 I \tag{101}$$

$$= \begin{bmatrix} U_{\mathcal{S}} & U_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{\mathcal{S}}^* \\ U_{\mathcal{N}}^* \end{bmatrix} + \begin{bmatrix} U_{\mathcal{S}} & U_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} \sigma^2 I_s & 0 \\ 0 & \sigma^2 I_{n-s} \end{bmatrix} \begin{bmatrix} U_{\mathcal{S}}^* \\ U_{\mathcal{N}}^* \end{bmatrix}. \tag{102}$$

A.3 Proof of Lemma 4.9

For any $\tau \in T$

$$a_{1:m}(\tau) = \exp(i2\pi\tau)\, a_{0:m-1}(\tau) \tag{103}$$

so

$$L_{Y_0,Y_1}(\exp(i2\pi\tau)) = Y_1 - \exp(i2\pi\tau)\, Y_0 \tag{104}$$

$$= \sum_{t_j \in T} c_j \left(a_{1:m}(t_j) - \exp(i2\pi\tau)\, a_{0:m-1}(t_j)\right) a_{0:n-m}(t_j)^T \tag{105}$$

$$= \sum_{t_j \in T/\{\tau\}} c_j \left(a_{1:m}(t_j) - \exp(i2\pi\tau)\, a_{0:m-1}(t_j)\right) a_{0:n-m}(t_j)^T, \tag{106}$$

which has rank $s-1$.

If $\tau \notin T$ the column space of $L_{Y_0,Y_1}(\exp(i2\pi\tau))$ is spanned by $\exp(i2\pi t_1)$, ..., $\exp(i2\pi t_s)$ and $\exp(i2\pi\tau)$. By Lemma 4.4 these vectors are linearly independent, so the matrix pencil has rank s.

A.4 Proof of Lemma 4.10

From (103) we have

$$A_{0:m-1}(T) = A_{0:m-1}(T)\, \Phi, \tag{107}$$

where

$$\Phi := \begin{bmatrix} \exp(i2\pi t_1) & 0 & \cdots & 0 \\ 0 & \exp(i2\pi t_2) & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \exp(i2\pi t_s) \end{bmatrix}. \tag{108}$$

Y_0 can be decomposed as

$$Y_0 = A_{0:m-1}(T)\, C A_{0:n-m}(T)^T. \tag{109}$$

Let $U \Sigma V^*$ be the singular value decomposition of $C\, A_{0:n-m}(T)^T$, where $U \in \mathbb{C}^{s\times s}$, $\Sigma \in \mathbb{C}^{s\times s}$ and $V \in \mathbb{C}^{n-m+1\times s}$. If the coefficients $c_1, \ldots, c_s$ are nonzero then C is full rank. $A_{0:n-m}(T)^T$ is full rank by Lemma 4.4. This implies that $V \Sigma^{-1} U^*$ is a right inverse of $C\, A_{0:n-m}(T)^T$,

$$C\, A_{0:n-m}(T)^T\, V \Sigma^{-1} U^* = I. \tag{110}$$

We now use these facts to decompose Y_1,

$$\begin{aligned} Y_1 &= A_{1:m}(T)\; C\, A_{0:n-m}(T)^T && (111)\\ &= A_{0:m-1}(T)\; \Phi\, C\, A_{0:n-m}(T)^T \qquad \text{by (107)} && (112)\\ &= A_{0:m-1}(T)\; C\, A_{0:n-m}(T)^T\, V \Sigma^{-1} U^* \Phi\, C\, A_{0:n-m}(T)^T \qquad \text{by (110)} && (113)\\ &= Y_0\, V \Sigma^{-1} U^*\, \Phi\, U \Sigma V^*. && (114) \end{aligned}$$

The row space of Y_0 is the same as the row space of $CA_{0:n-m}(T)^T$ because $A_{0:m-1}(T)^T$ is full rank by Lemma 4.4. This implies that

$$\begin{aligned} V_0 \Sigma_0^{-1} U_0^*\, Y_0 V &= V_0 V_0^* V && (115)\\ &= V. && (116) \end{aligned}$$

As a consequence

$$\begin{aligned} V_0 \Sigma_0^{-1} U_0^*\; Y_1 &= V_0 \Sigma_0^{-1} U_0^* Y_0\, V \Sigma^{-1} U^*\, \Phi\, U \Sigma V^* && (117)\\ &= V \Sigma^{-1} U^*\, \Phi\, U \Sigma V^* && (118)\\ &= P^{-1} \begin{bmatrix} \Phi & 0 \\ 0 & 0 \end{bmatrix} P, && (119) \end{aligned}$$

where

$$P := \begin{bmatrix} U & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} V^* \\ V_\perp^* \end{bmatrix} \tag{120}$$

and $V_\perp$ is any orthogonal matrix whose column space is the orthogonal complement to the column space of V. This establishes that the nonzero eigenvalues of the matrix $V_0 \Sigma_0^{-1} U_0^*\, Y_1$ correspond to the diagonal entries of Φ.

Sparse regression

1 Linear regression

In statistics, the problem of regression is that of learning a function that allows to estimate a certain quantity of interest, the *response* or *dependent variable*, from several observed variables, known as *covariates*, *features* or *independent variables*. For example, we might be interested in estimating the price of a house based on its extension, the number of rooms, the year it was built, etc. The function that models the relation between the response and the predictors is learnt from training data and can then be used to predict the response for new examples.

In linear regression, we assume that the response is well modeled as a linear combination of the predictors. The model is parametrized by an intercept $\beta_0 \in \mathbb{R}$ and a vector of weights $\beta \in \mathbb{R}^p$, where p is the number of predictors. Let us assume that we have n data points consisting of a value of the response y_i and the corresponding values of the predictors X_{i1}, X_{i2}, ..., X_{ip}, $1 \leq i \leq n$. The linear model is given by

$$y_i \approx \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}, \quad 1 \leq i \leq n, \tag{1}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} \approx \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} \beta_0 + \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ & & \cdots & \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_p \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} \beta_0 + X\beta. \tag{2}$$

We have already encountered linear models that arise in inverse problems such as compressed sensing or super-resolution. A major difference between statistics and those applications is that in statistics the ultimate aim is to predict y accurately for new values of the predictors, *not* to estimate β. The role of β is merely to quantify the linear relation between the response and the predictors. In contrast, when solving an inverse problems the main objective is to determine β, which has a physical interpretation: an image of a 2D slice of a body in MRI, the spectrum of multisinusoidal signal in spectral super-resolution, reflection coefficients of strata in seismography, etc.

1.1 Least-squares estimation

To calibrate the linear regression model, we estimate the weight vector from the training data. By far the most popular method to compute the estimate is to minimize the ℓ_2 norm

of the fitting error on the training set. In more detail, the weight estimate β_{ls} is the solution of the least-squares problem

$$\text{minimize} \quad \left|\left| y - X\tilde{\beta} \right|\right|_2 . \tag{3}$$

The least-squares cost function is convenient from a computational view, since it is convex and can be minimized efficiently (in fact, as we will see in a moment it has a closed-form solution). In addition, as detailed in Proposition 1.3 below, it has a reasonable probabilistic interpretation.

The following proposition, proved in Section 1.1, shows that the solution to the least-squares problem has a closed form solution.

Proposition 1.1 (Least-squares solution)**.** *For $p \geq n$, if X is full rank then the solution to the least-squares problem* (3) *is*

$$\beta_{\text{ls}} := \left(X^T X\right)^{-1} X^T y. \tag{4}$$

A corollary to this result provides a geometric interpretation for the least-squares estimate of y: it is obtained by projecting the response onto the column space of the matrix formed by the predictors.

Corollary 1.2. *For $p \geq n$, if X is full rank then $X\beta_{\text{ls}}$ is the projection of y onto the column space of X.*

Proof. Let $X = U\Sigma V^T$ be the singular-value decomposition of X. Since X is full rank and $p \geq n$ we have $U^T U = I$, $V^T V = I$ and Σ is a square invertible matrix, which implies

$$\begin{aligned} X\beta_{\text{ls}} &= X\left(X^T X\right)^{-1} X^T y & (5)\\ &= U\Sigma V^T \left(V\Sigma U^T U \Sigma V^T\right) V\Sigma U^T y & (6)\\ &= UU^T y. & (7) \end{aligned}$$

□

Proposition 1.3 (Least-squares solution as maximum-likelihood estimate)**.** *If we model y as*

$$y = X\beta + z \tag{8}$$

where $X \in \mathbb{R}^{n\times p}$, $p \geq n$, $\beta \in \mathbb{R}^p$ and $y \in \mathbb{R}^n$ are fixed and the entries of $z \in \mathbb{R}^n$ are iid Gaussian random variables with mean zero and the same variance, then the maximum-likelihood estimate of β given y is equal to β_{ls}.

The proposition is proved in Section A.2 of the appendix.

1.2 Preprocessing

The careful reader might notice that we have not explained how to fit the intercept β_0. Before fitting a linear regression model, we typically perform the following preprocessing steps.

1. Centering each predictor column X_i subtracting its mean

$$\mu_j := \sum_{i=1}^{n} X_{ij} \tag{9}$$

   so that each column of X has mean zero.

2. Normalizing each predictor column X_j by dividing by

$$\sigma_j := \sqrt{\sum_{i=1}^{n} (X_{ij} - \mu_j)^2} \tag{10}$$

   so that each column of X has ℓ_2 norm equal to one. The objective is to make the estimate invariant to the units used to measure each predictor.

3. Centering the response vector y by subtracting its mean

$$\mu_y := \sum_{i=1}^{n} y_i. \tag{11}$$

By the following lemma, the intercept of the linear model once we have centered the data is equal to zero.

Lemma 1.4 (Intercept)**.** *If the mean of y and of each of the columns of X is zero, then the intercept in the least-squares solution is also zero.*

The lemma is proved in Section A.3 of the appendix.

Once have solved the least-squares problem using the centered and normalized data to obtain β_{ls}, we can use the model to estimate the response corresponding to a new vector of predictors $x \in \mathbb{R}^p$ by computing

$$f(x) := \mu_y + \sum_{i=1}^{p} \beta_{\text{ls},i} \, \frac{x_i - \mu_i}{\sigma_i}. \tag{12}$$

1.3 Overfitting

Imagine that a friend tells you:

I found a cool way to predict the temperature in New York: It's just a linear combination of the temperature in every other state. I fit the model on data from the last month and a half and it's perfect!

Your friend is not lying, but the problem is that she is using a number of data points to fit the linear model that is roughly the same as the number of parameters. If $n = p$ we can find a β such that $y = X\beta$ exactly, even if y and X have nothing to do with each other! This is called overfitting and is usually caused by using a model that is too flexible with respect to the number of data that are available.

To evaluate whether a model suffers from overfitting we separate the data into a training set and a test set. The training set is used to fit the model and the test set is used to evaluate the error. A model that overfits the training set will have a very low error when evaluated on the training examples, but will not generalize well to the test examples.

Figure 1 shows the result of evaluating the training error and the test error of a linear model with $p = 50$ parameters fitted from n training examples. The training and test data are generated by fixing a vector of weights β and then computing

$$y_{\text{train}} = X_{\text{train}}\,\beta + z_{\text{train}}, \tag{13}$$

$$y_{\text{test}} = X_{\text{test}}\,\beta, \tag{14}$$

where the entries of X_{train}, X_{test}, z_{train} and β are sampled independently at random from a Gaussian distribution with zero mean and unit variance. The training and test errors are defined as

$$\text{error}_{\text{train}} = \frac{||X_{\text{train}}\,\beta_{\text{ls}} - y_{\text{train}}||_2}{||y_{\text{train}}||_2}, \tag{15}$$

$$\text{error}_{\text{test}} = \frac{||X_{\text{test}}\,\beta_{\text{ls}} - y_{\text{test}}||_2}{||y_{\text{test}}||_2}. \tag{16}$$

Note that even the true β does not achieve zero training error because of the presence of the noise, but the test error is actually zero if we manage to estimate β exactly.

The training error of the linear model grows with n. This makes sense as the model has to fit more data using the same number of parameters. When n is close to p (50), the fitted model is much better than the true model at replicating the training data (the error of the true model is shown in green). This is a sign of overfitting: the model is adapting to the noise and not learning the true linear structure. Indeed, in that regime the test error is extremely high. At larger n, the training error rises to the level achieved by the true linear model and the test error decreases, indicating that we are learning the underlying model.

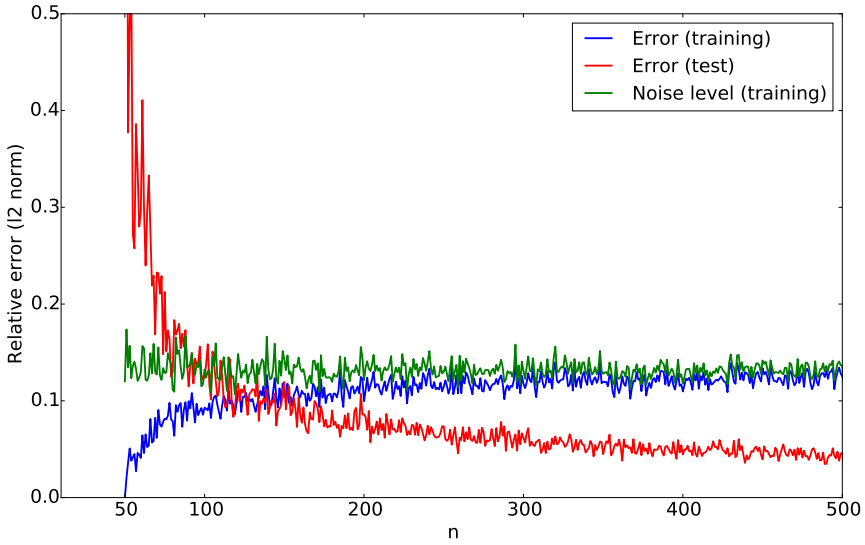


Figure 1: Relative ℓ_2-norm error in estimating the response achieved using least-squares regression for different values of n (the number of training data). The training error is plotted in blue, whereas the test error is plotted in red. The green line indicates the training error of the true model used to generate the data.

1.4 Theoretical analysis of linear regression

In this section we analyze the solution of the least-squares regression fit to understand its dependence with the number of training examples. The following theorem, proved in Section A.4 of the appendix, characterizes the error in estimating the weights under the assumption that the data are indeed generated by a linear model with additive noise.

Theorem 1.5. *Assume the data y are generated according to a linear model with additive noise,*

$$y = X\beta + z, \tag{17}$$

where $X \in \mathbb{R}^{n \times p}$ and $\beta \in \mathbb{R}^p$, and that the entries of $z \in \mathbb{R}^n$ are drawn independently at random from a Gaussian distribution with zero mean and variance σ_z^2. The least-squares estimate

$$\beta_{\text{ls}} := \arg\min_{\tilde{\beta}} \left|\left| y - X\tilde{\beta} \right|\right|_2 \tag{18}$$

satisfies

$$\frac{p\,\sigma_z^2\,(1-\epsilon)}{\sigma_{\max}^2} \leq ||\beta - \beta_{\text{ls}}||_2^2 \leq \frac{p\,\sigma_z^2\,(1+\epsilon)}{\sigma_{\min}^2} \tag{19}$$

with probability $1 - 2\exp\left(-\frac{p\epsilon^2}{8}\right)$. $\sigma_{\min}$ *and* $\sigma_{\max}$ *denote the smallest and largest singular value of* X *respectively.*

The bounds in the theorem are in terms of the singular values of the predictor matrix $X \in \mathbb{R}^{n \times p}$. To provide some intuition as to the scaling of these singular values when we fix p and increase n, let us assume that the entries of X are drawn independently at random from a standard Gaussian distribution. Then by Proposition 3.4 in Lecture Notes 5 both $\sigma_{\min}$ and $\sigma_{\max}$ are close to $\sqrt{n}$ with high probability as long as $n > C\,p$ for some constant C. This implies that if the variance of the noise z equals one,

$$||\beta - \beta_{\text{ls}}||_2 \approx \sqrt{\frac{p}{n}}. \tag{20}$$

If each of the entries of β has constant magnitude the ℓ_2 norm of β is approximately equal to $\sqrt{p}$. In this case, the theoretical analysis predicts that the normalized error

$$\frac{||\beta - \beta_{\text{ls}}||_2}{||\beta||_2} \approx \frac{1}{\sqrt{n}}. \tag{21}$$

Figure 2 shows the result of a simulation where the entries of X, β and z are all generated by sampling independently from standard Gaussian random variables. The relative error scales precisely as $1/\sqrt{n}$.

For a fixed number of predictors, this analysis indicates that the least-squares solution converges to the true weights as the number of data n grows. In statistics jargon, the estimator is *consistent*. The result suggests two scenarios in which the least-squares estimator may not yield a good estimate: when p is of the same order as n and when some of the predictors are highly correlated, as some of the singular values of X may be very small.

1.5 Global warming

In this section we describe the application of linear regression to climate data. In particular, we analyze temperature data taken in a weather station in Oxford over 150 years.[1] Our objective is not to perform prediction, but rather to determine whether temperatures have risen or decreased during the last 150 years in Oxford.

In order to separate the temperature into different components that account for seasonal effects we use a simple linear with three predictors and an intercept

$$y \approx \beta_0 + \beta_1 \cos\left(\frac{2\pi t}{12}\right) + \beta_2 \cos\left(\frac{2\pi t}{12}\right) + \beta_3\, t \tag{22}$$

[1]The data is available at `http://www.metoffice.gov.uk/pub/data/weather/uk/climate/stationdata/oxforddata.txt`.

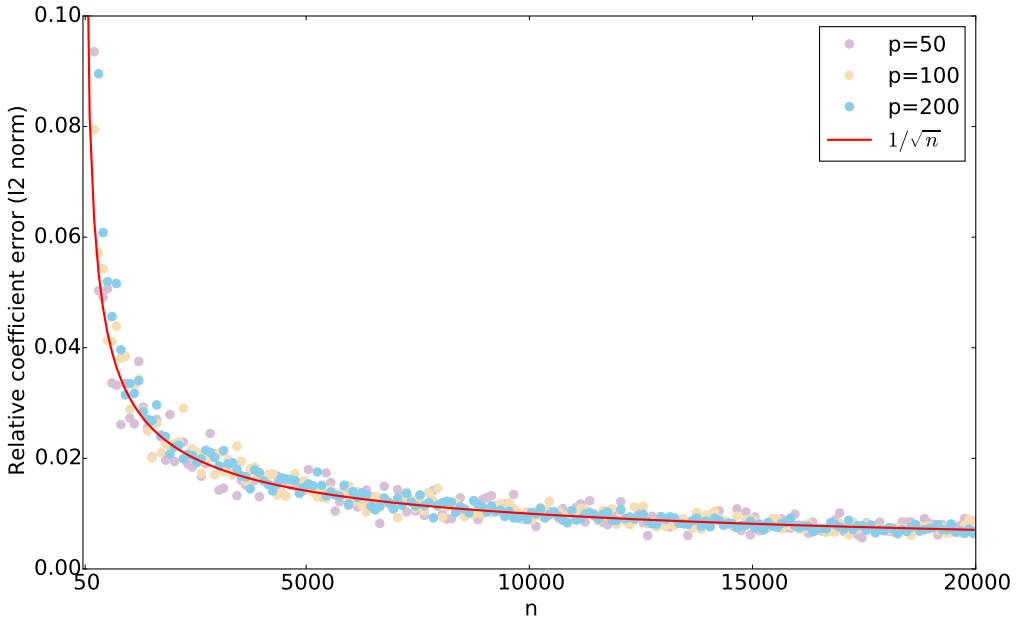


Figure 2: Relative ℓ_2-norm error of the least-squares estimate as n grows. The entries of X, β and z are all generated by sampling independently from standard Gaussian random variables. The simulation is consistent with 21.

where t denotes the time in months. The corresponding matrix of predictors is

$$X := \begin{bmatrix} 1 & \cos\left(\frac{2\pi t_1}{12}\right) & \sin\left(\frac{2\pi t_1}{12}\right) & t_1 \\ 1 & \cos\left(\frac{2\pi t_2}{12}\right) & \sin\left(\frac{2\pi t_2}{12}\right) & t_2 \\ \dots & \dots & \dots & \dots \\ 1 & \cos\left(\frac{2\pi t_n}{12}\right) & \sin\left(\frac{2\pi t_n}{12}\right) & t_n \end{bmatrix}. \tag{23}$$

The intercept β_0 represents the mean temperature, β_1 and β_2 account for periodic yearly fluctuations and β_3 is the overall trend. If β_3 is positive then the model indicates that temperatures are increasing, if it is negative then it indicates that temperatures are decreasing.

The results of fitting the linear model are shown in Figures 3 and 4. The fitted model indicates that both the maximum and minimum temperatures have an increasing trend of about 0.8 degrees Celsius (around 1.4 degrees Fahrenheit).

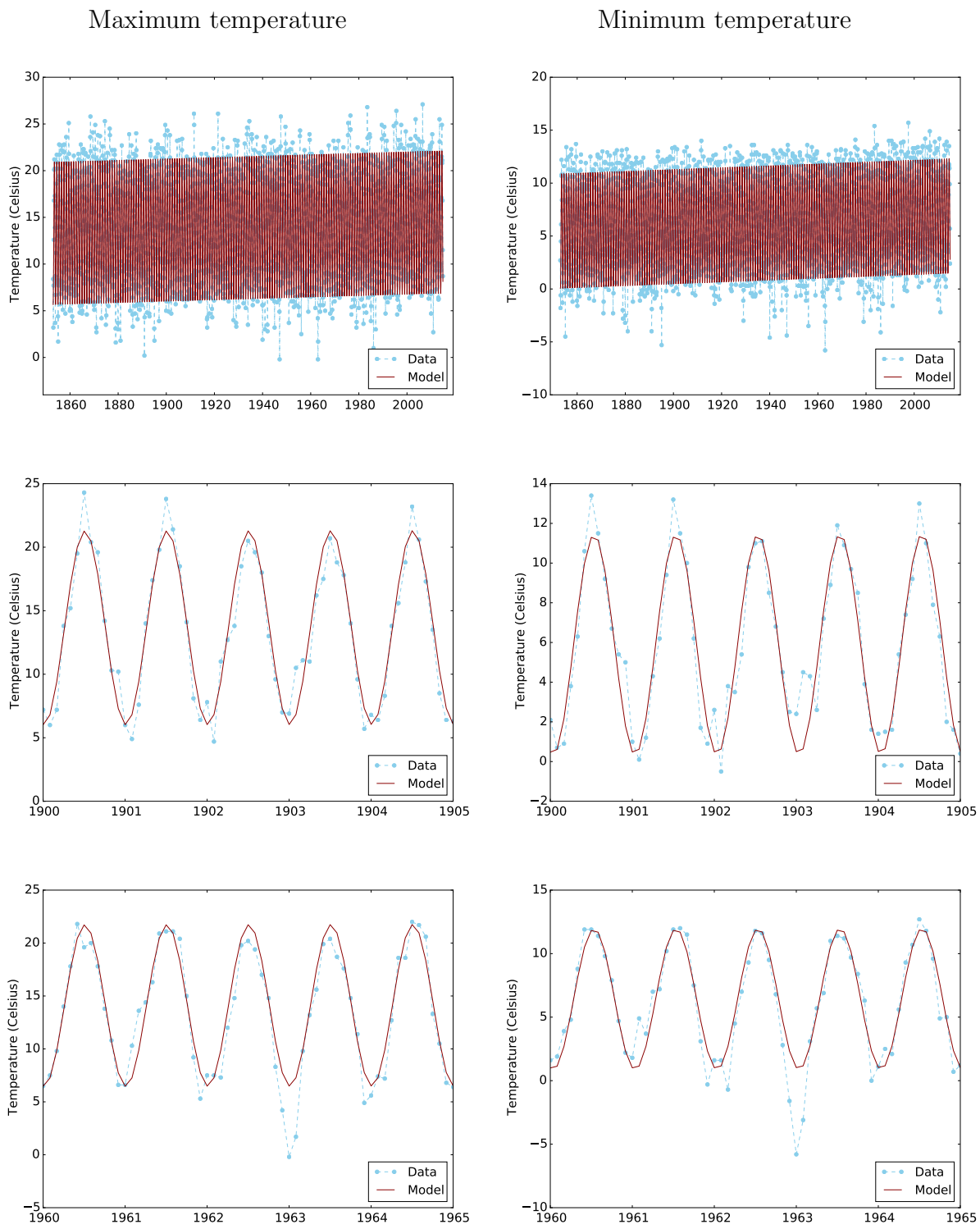
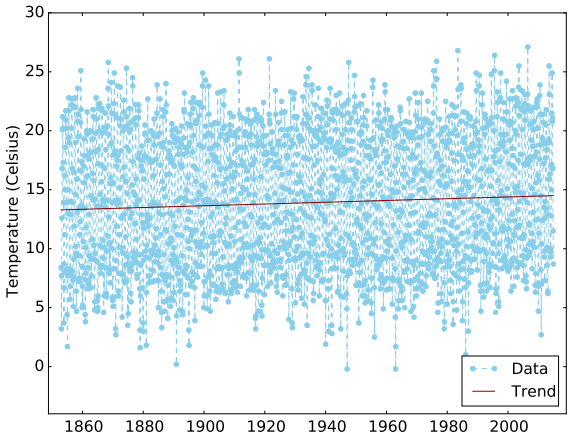


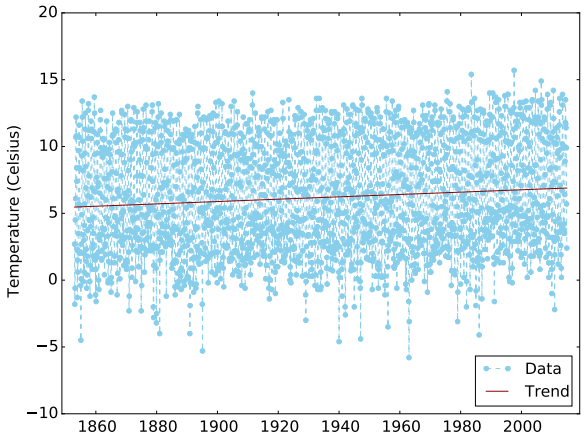
Figure 3: Temperature data together with the linear model described by (22) for both maximum and minimum temperatures.

Maximum temperature



+ 0.75 °C / 100 years

Minimum temperature



+ 0.88 °C / 100 years

Figure 4: Temperature trend obtained by fitting the model described by (22) for both maximum and minimum temperatures.

1.6 Logistic regression

The problem of classification in statistics and machine learning is very related to regression. The only difference is that in classification the response is binary: it is equal to either 0 or 1. Of course, the labels 0 and 1 are arbitrary, they represent two distinct classes into which we aim to classify our data. For example, the predictors might be pictures and the two classes *cats* and *dogs*.

In linear regression, we use a linear model to predict the response from the predictors. In logistic regression, we use a linear model to predict *how likely* it is for the response to equal 0 or 1. This requires mapping the output of the linear model to the $[0, 1]$ interval, which is achieved by applying the logistic function or sigmoid,

$$g(t) := \frac{1}{1 + \exp -t}, \tag{24}$$

to a linear combination of the predictors. In more detail, we model the probability that $y_i = 1$ by

$$\mathrm{P}(y_i = 1 | X_{i1}, X_{i2}, \ldots, X_{ip}) \approx g\left(\beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}\right) \tag{25}$$

$$= \frac{1}{1 + \exp\left(-\beta_0 - \sum_{j=1}^{p} \beta_j X_{ij}\right)}. \tag{26}$$

In Proposition 1.3 we established that least-squares fitting computes a maximum-likelihood estimate in the case of linear models with additive Gaussian noise. The following proposition derives a cost function to calibrate a logistic-regression model by maximizing the likelihood under the assumption that the response is a Bernouilli random variable parametrized by the linear model.

Proposition 1.6 (Logistic-regression cost function)**.** *If we assume that the response values $y_1, y_2, \ldots, y_n$ in the training data are independent samples from Bernouilli random variables $\check{y}_1, \check{y}_2, \ldots, \check{y}_n$ such that*

$$\mathrm{P}\left(\check{y}_i = 1 | X_{i1}, X_{i2}, \ldots, X_{ip}\right) := g\left(\beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}\right), \tag{27}$$

$$\mathrm{P}\left(\check{y}_i = 0 | X_{i1}, X_{i2}, \ldots, X_{ip}\right) := 1 - g\left(\beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}\right), \tag{28}$$

then the maximum-likelihood estimate of the intercept and the weights β_0 and β_0 are obtained by maximizing the function

$$\log \mathcal{L}\left(\tilde{\beta}_0, \tilde{\beta}\right) := \sum_{i=1}^{n} y_i \log g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right) + (1 - y_i) \log\left(1 - g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right)\right). \tag{29}$$

Proof. Due to the independence assumption, the joint probability mass function (pmf) of the random vector $\check{y}$ equals

$$p_{\check{y}}(y) := \prod_{i=1}^{n} g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right)^{y_i} \left(1 - g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right)\right)^{1-y_i}. \tag{30}$$

The likelihood is defined as the joint pmf parametrized by the weight vectors,

$$\mathcal{L}\left(\tilde{\beta}_0, \tilde{\beta}\right) := \prod_{i=1}^{n} g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right)^{y_i} \left(1 - g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right)\right)^{1-y_i}. \tag{31}$$

Taking the logarithm of this nonnegative function completes the proof. □

The log-likelihood function is strictly concave, so the logistic-regression estimate is well defined. Although the cost function is derived by assuming that the data follow a certain probabilistic model, logistic regression is widely deployed in situations where the probabilistic assumptions do not hold. The model will achieve high prediction accuracy on any dataset where the predictors are linearly separable, as long as sufficiently data is available.

2 Sparse regression

2.1 Model selection

In Section 1.4 we establish that linear regression allows to learn a linear model when the number of available examples n is large with respect to the number of predictors p. However, in many modern applications, the number of predictors can be extremely large. An example is computational genomics, where the predictors may correspond to gene-expression measurements from thousands of genes, whereas n is the number of patients which might only be in the hundreds.

It is obviously impossible to fit a linear model when $p > n$, or even when $p \approx n$ without overfitting (depending on the noise level), but it may still be possible to fit a sparse linear model that only depends on a subset of $s < p$ predictors. Selecting the relevant predictors to include in the model is called *model selection* in statistics.

2.2 Best subset selection and forward stepwise regression

A possible way to select a small number of relevant predictors from a training set is to fix the order of the sparse model $s < p$ and then evaluate the least-squares fit of all possible s-sparse models in order to select the one that provides the best fit. This is called the *best-subset selection* method. Unfortunately it is computationally intractable even for small values of s and p. For instance, there are more than 10^{13} possible models if $s = 10$ and $p = 100$.

An alternative to an exhaustive evaluation of all possible sparse models is to select the predictors greedily in the spirit of signal-processing methods such as orthogonal matching pursuit. In *forward stepwise regression* we select the predictor that is most correlated with the response and then project the rest of predictors onto its orthogonal complement. Iterating this procedure allows to learn an s-sparse model in s steps.

Algorithm 2.1 (Forward stepwise regression)**.** *Given a matrix of predictors $X \in \mathbb{R}^{n \times p}$ and a response $y \in \mathbb{R}^n$, we initialize the residual and the subset of relevant predictors $\mathcal{S}$ by setting,*

$$j_0 := \arg\max_j |\langle y, X_j \rangle| \tag{32}$$

$$\mathcal{S}_0 := \{j_0\} \tag{33}$$

$$\beta_{\text{ls}} := \arg\min_{\tilde{\beta}} \left|\left| y - X_{\mathcal{S}_0} \tilde{\beta} \right|\right|_2 \tag{34}$$

$$r^{(0)} := y - X_{\mathcal{S}_0} \beta_{\text{ls}}. \tag{35}$$

Then for $k = 2, 3, \ldots, s$ we compute

$$j_k := \arg\max_{j \notin \mathcal{S}_{j-1}} \left| \left\langle y, \mathcal{P}_{\mathrm{col}\left(X_{\mathcal{S}_{j-1}}\right)^{\perp}} X_j \right\rangle \right| \tag{36}$$

$$\mathcal{S}_j := \mathcal{S}_{j-1} \cup \{j_k\} \tag{37}$$

$$\beta_{\mathrm{ls}} := \arg\min_{\tilde{\beta}} \left\| y - X_{\mathcal{S}_j} \tilde{\beta} \right\|_2 \tag{38}$$

$$r^{(k)} := r^{(k-1)} - X_{\mathcal{S}_j} \beta_{\mathrm{ls}}. \tag{39}$$

The algorithm is very similar to orthogonal matching pursuit. The only difference is the orthogonalization step in which we project the remaining predictors onto the orthogonal complement of the span of the predictors that have been selected already.

2.3 The lasso

Fitting a sparse model that uses a subset of the available predictors is equivalent to learning a weight vector β that only contains a small number of nonzeros. As we saw in the case of sparse signal representations and underdetermined inverse problems, penalizing the ℓ_1 norm is an efficient way of promoting sparsity. In statistics, ℓ_1-norm regularized least squares is known as the *lasso*,

$$\text{minimize} \quad \frac{1}{2n} \left\| y - \tilde{\beta}_0 - X\tilde{\beta} \right\|_2^2 + \lambda \left\| \tilde{\beta} \right\|_1 \tag{40}$$

$$\tag{41}$$

$\lambda > 0$ is a regularization parameter that controls the tradeoff between the fit to the data and the ℓ_1 norm of the weights. Figure 5 shows the values of the coefficients obtained by the lasso for a linear regression problem with 50 predictors where the response only depends on 3 of them. If λ is very large, all coefficients are equal to zero. If λ is very small, then the lasso estimate is equal to the least-squares estimate. For values in between, the lasso yields a sparse model containing the coefficients corresponding to the relevant predictors.

A different formulation for the lasso, which is the one that appeared in the original paper [4], incorporates a constraint on the ℓ_1 norm of the weight vector, instead of an additive term.

$$\text{minimize} \quad \left\| y - X\tilde{\beta} - \tilde{\beta}_0 \right\|_2^2 \tag{42}$$

$$\text{subject to} \quad \left\| \tilde{\beta} \right\|_1 \leq \tau \tag{43}$$

The two formulations are equivalent, but the relation between λ and τ depends on X and y.

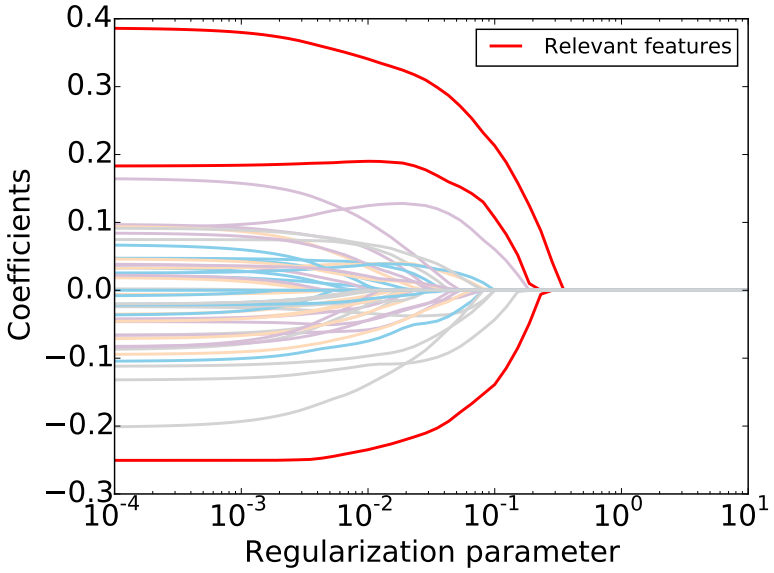


Figure 5: Magnitude of the lasso coefficients for different value of the regularization parameter. The number of predictors is 50, but the response only depends on 3 of them, which are marked in red.

To compare the performance of the lasso, forward stepwise regression and least-squares regression on a dataset that follows a sparse linear model we generate simulated data by computing

$$y_{\text{train}} = X_{\text{train}}\,\beta + z_{\text{train}}, \tag{44}$$

$$y_{\text{test}} = X_{\text{test}}\,\beta, \tag{45}$$

where the entries of X_{train}, X_{test} and z_{train} and β are sampled independently at random from a Gaussian distribution with zero mean and unit variance. 10 entries of β are also sampled iid from a standard normal, but the rest are set to zero. As a result the response only depends on 10 predictors out of a total of 50. The training and test errors were computed as in (15) and (16).

Figure 6 shows the results for different values of n (to be clear we compute the estimates once at each value of n). As expected, the least-squares estimator overfits the training data and performs very badly on the test set for small values of n. In contrast, the lasso and forward stepwise regression do not overfit and achieve much smaller errors on the test set, even when n is equal to p.

In our implementation of forward stepwise regression we set the number of predictors in the sparse model to the true number. On real data we would need to also estimate the order of the model. The greedy method performs very well in some instances because it manages to

select the correct sparse model. However, in other cases, mistakes in selecting the relevant predictors produce high fit errors, even on the training set. In contrast, the lasso achieves accurate fits for every value of n.

2.4 Theoretical analysis of the lasso

Assume that we have data that follow a sparse linear model of the form

$$y = X\beta + z \tag{46}$$

where the weight vector β is sparse, so that the response y only depends on $s < p$ predictors. By Theorem 1.5, if the noise has variance σ^2, least-squares regression yields an estimate of the weight vector that satisfies

$$||\beta - \beta_{\text{ls}}||_2 \approx \sigma_z \sqrt{\frac{p}{n}}. \tag{47}$$

as long as the matrix of predictors X is well conditioned and has entries with constant amplitudes. When p is close to n this indicates that least-squares regression does not yield a good estimate.

In order to characterize the error achieved by the lasso, we introduce the restricted-eigenvalue property (REP), which is similar to the restricted-isometry property (RIP) that we studied in Lecture Notes 5.

Definition 2.2 (Restricted-eigenvalue property)**.** *A matrix $M \in \mathbb{R}^{n \times p}$ satisfies the restricted-eigenvalue property with parameter s if there exists $\gamma > 0$ such that for any $v \in \mathbb{R}^p$ if*

$$||v_{T^c}||_1 \leq ||v_T||_1 \tag{48}$$

for any subset T with cardinality s then

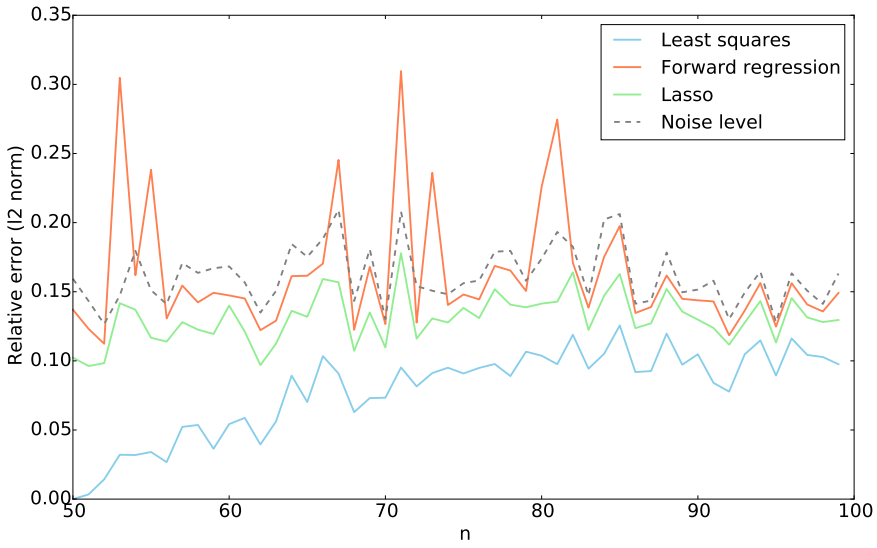
$$\frac{1}{n} ||M\ v||_2^2 \geq \gamma\, ||v||_2^2 . \tag{49}$$

Just like the RIP, the REP states that the matrix preserves the norm of a certain class of vectors. In this case, the vectors are not necessarily sparse, but rather have ℓ_1-norm concentrated on a sparse subset of the entries, which can be interpreted as a robustified notion of sparsity. The property may hold even if $p > n$, i.e. when we have more predictors than examples. The following theorem provides guarantees for the lasso under the REP.

Theorem 2.3. *Assume that the data y are generated according to a linear model with additive noise,*

$$y = X\beta + z, \tag{50}$$

Training



Test

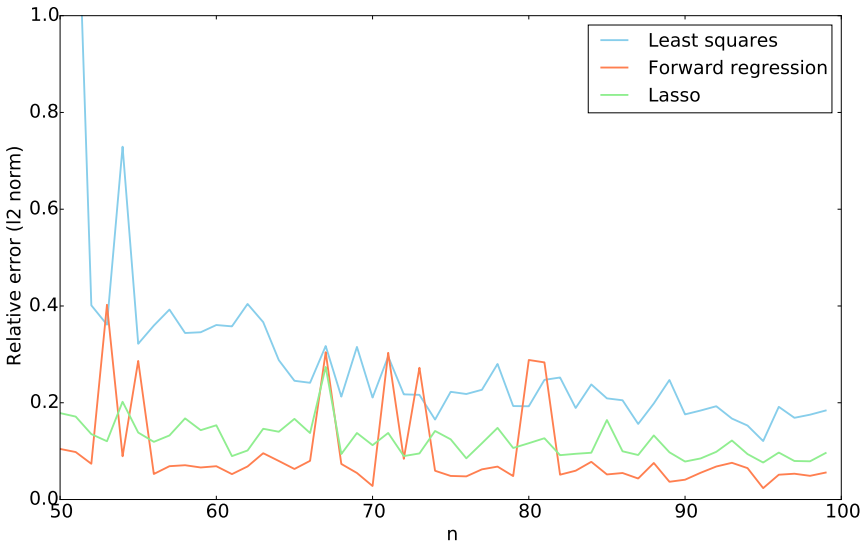


Figure 6: Comparison of the training and test error of the lasso, forward stepwise regression and least-squares regression for simulated data where the number of predictors is equal to 50 but only 10 are used to generate the response.

where $X \in \mathbb{R}^{n \times p}$ and $\beta \in \mathbb{R}^p$, and that the entries of $z \in \mathbb{R}^n$ are drawn independently at random from a Gaussian distribution with zero mean and variance σ_z^2. If β has s nonzero entries and X satisfies the restricted-eigenvalue property, the solution β_{lasso} to

$$\text{minimize} \quad \left|\left| y - X\tilde{\beta} \right|\right|_2^2 \tag{51}$$

$$\text{subject to} \quad \left|\left| \tilde{\beta} \right|\right|_1 \leq \tau \tag{52}$$

if we set $\tau := ||\beta||_1$ satisfies

$$||\beta - \beta_{\text{lasso}}||_2 \leq \frac{\sigma_z \sqrt{32\ \alpha\ s \log p}}{\gamma\ n} \max_i ||X_i||_2 \tag{53}$$

with probability $1 - 2\exp\left(-\left(\alpha - 1\right)\log p\right)$ for any $\alpha > 2$.

The result establishes that the lasso achieves an error that scales as $\sigma_z\sqrt{s/n}$, which is the same rate achieved by least squares if the true sparse model is known!

In this section we have focused on the estimation of the weight vector. This is important for model selection, as the sparsity pattern and the amplitude of the weights reveals the sparse model used to predict the response. However, in statistics the main aim is often to predict the response. For this purpose, in principle, conditions such as the REP should not be necessary. For results on the prediction error of the lasso we refer the interested reader to Chapter 11 of [3]. In Section 3 we discuss the performance of the lasso and related methods when the predictor matrix does not satisfy the REP.

2.5 Sparse logistic regression

An advantage of the lasso over greedy methods to learn sparse linear models is that it can be easily applied to logistic regression. All we need to do is add an ℓ_1-norm regularization term to the cost function derived in Section 1.6. In detail, to learn a sparse logistic-regression model we minimize the function

$$-\sum_{i=1}^{n} y_i \log g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right) - (1 - y_i) \log\left(1 - g\left(\tilde{\beta}_0 + \sum_{j=1}^{p} \tilde{\beta}_j X_{ij}\right)\right) + \lambda \left|\left| \tilde{\beta} \right|\right|_1.$$

This version of the lasso can be used to obtain a sparse logistic model for prediction of binary responses. To illustrate this, we consider a medical dataset[2]. The response indicates whether 271 patients suffer from arrhythmia or not. The predictors contain information about each patient, such as age, sex, height and weight, as well as features extracted from

[2]The data can be found at `https://archive.ics.uci.edu/ml/datasets/Arrhythmia`

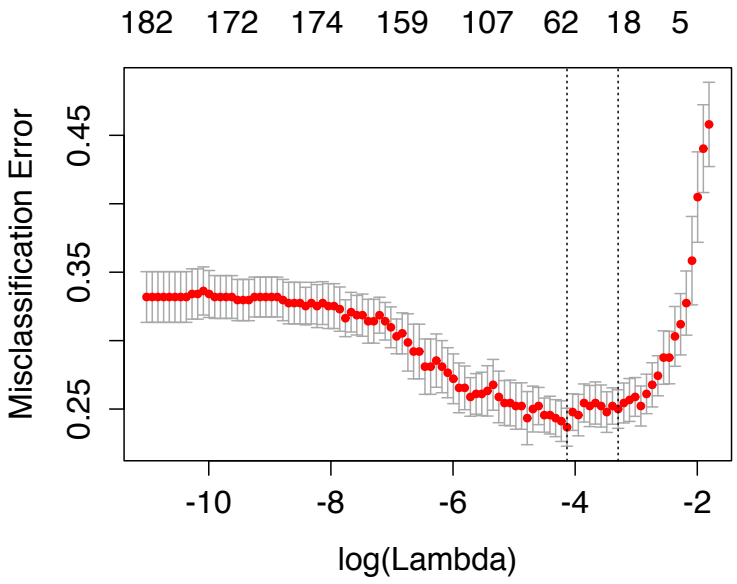


Figure 7: Distribution of misclassification errors achieved after repeatedly fitting the model using a random subset of 90% of the examples for different values of the regularization parameter λ. The number of predictors included in the sparse model is indicated above the graph.

electrocardiogram recordings. The total number of predictors is 182. We use the glmnet package in R [2] to fit the sparse logistic regression model.

Figure 7 shows the distribution of the test error achieved after repeatedly fitting the model using a random subset of 90% of the examples; a procedure known as *cross-validation* in statistics. The number of predictors included in the sparse model is indicated above the graph. The best results are obtained by a model containing 62 predictors, but a model containing 18 achieves very similar accuracy (both are marked with a dotted line).

3 Correlated predictors

In many situations, some of the predictors in a dataset may be highly correlated. As a result, the predictor matrix is ill conditioned, which is problematic for least squares regression, and also for the lasso. In this section, we discuss this issue and show how it can be tackled through regularization of the regression cost function.

3.1 Ridge regression

When the data in a linear regression problem is of the form $y = X\beta + z$, we can write the error of the least-squares estimator in terms of the singular-value decomposition of $X = U\Sigma V^T$,

$$||\beta - \beta_{\text{ls}}||_2 = \sqrt{\sum_{j=1}^{p} \left(\frac{U_j^T z}{\sigma_j} \right)^2}, \tag{54}$$

see (102) in Section 1.5. If a subset of the predictors are highly correlated, then some of the singular values will have very small values, which results in noise amplification. *Ridge regression* is an estimation technique that controls noise amplification by introducing an ℓ_2-norm penalty on the weight vector,

$$\text{minimize} \quad \left|\left| y - X\tilde{\beta} \right|\right|_2^2 + \lambda \left|\left| \tilde{\beta} \right|\right|_2^2, \tag{55}$$

where $\lambda > 0$ is a regularization parameter that controls the weight of the regularization term as in the lasso. In inverse problems, ℓ_2-norm regularization is often known as Tikhonov regularization.

The following proposition shows that, under the assumption that the data indeed follow a linear model, the error of the ridge-regression estimator can be decomposed into a term that depends on the signal and a term that depends on the noise.

Proposition 3.1 (Ridge-regression error)**.** *If $y = X\beta + z$ and $X \in \mathbb{R}^{n\times p}$, $n \geq p$, is full rank, then the solution of Problem* (55) *can be written as*

$$\beta_{\text{ridge}} = V \begin{bmatrix} \frac{\sigma_1^2}{\sigma_1^2+\lambda} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2^2}{\sigma_2^2+\lambda} & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \frac{\sigma_p^2}{\sigma_p^2+\lambda} \end{bmatrix} V^T\beta + V \begin{bmatrix} \frac{\sigma_1}{\sigma_1^2+\lambda} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2}{\sigma_2^2+\lambda} & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \frac{\sigma_p}{\sigma_p^2+\lambda} \end{bmatrix} U^T z. \tag{56}$$

We defer the proof to Section A.6 in the appendix.

Increasing the value of the regularization parameter λ allows to control the noise term when some of the predictors are highly correlated. However, this also increases the error term that depends on the original signal; if $z = 0$ then we don't recover the true weight vector β unless $\lambda = 0$. Calibrating the regularization parameter allows to adapt to the conditioning of the predictor matrix and the noise level in order to achieve a good tradeoff between both terms.

3.2 The elastic net

Figure 8 shows the coefficients of the lasso and ridge regression learnt from a dataset where the response follows a sparse regression model. The model includes 50 predictors but only 12 are used to generate the response. These 12 predictors are divided into two groups of 6 that are highly correlated[3].

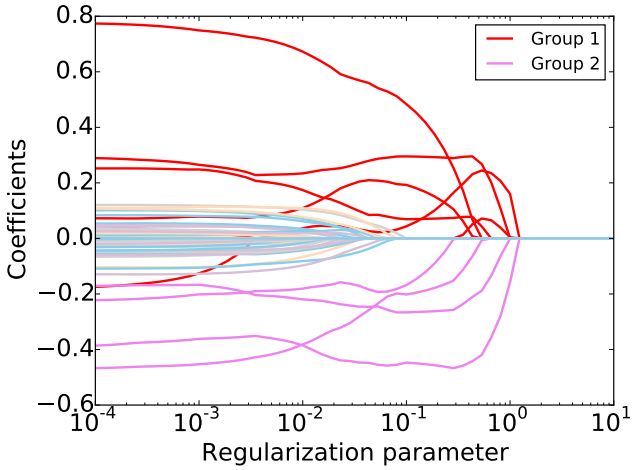
The model obtained using ridge regression assigns similar weights to the correlated variables. This is a desirably property since all of these predictors are equally predictive of the response value. However, the learnt model is not sparse for any value of the regularization parameter, which means that it selects all of the irrelevant predictors. In contrast, the lasso produces a sparse model, but the coefficients of the relevant predictors are very erratic. In fact, in the regime where the coefficients are sparse not all the relevant predictors are included in the model (two from the second group are missing).

The following lemma, proved in gives some intuition as to why the coefficient path for the lasso tends to be erratic when some predictors are highly correlated. When two predictors are exactly the same, then the lasso chooses arbitrarily between the two, instead of including both in the model with similar weights.

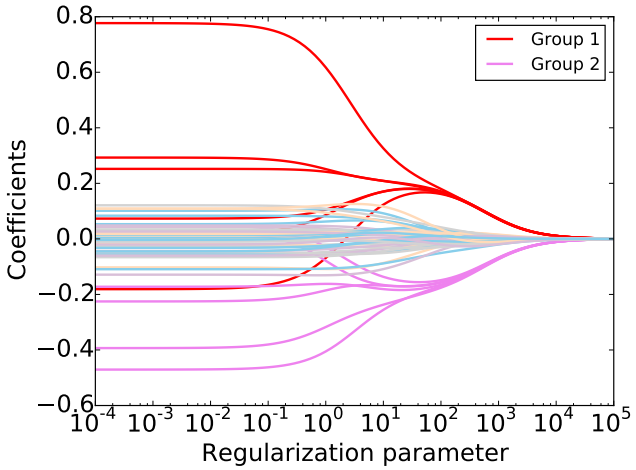
Lemma 3.2. *If two columns of the predictor matrix X are identical $X_i = X_j$, $i \neq j$, and*

[3]In more detail, the predictors in each group are sampled from a Gaussian distribution with zero mean and unit variance such that the covariance between each pair of predictors is equal to 0.95.

Lasso



Ridge regression



Elastic net ($\alpha = 0.5$)

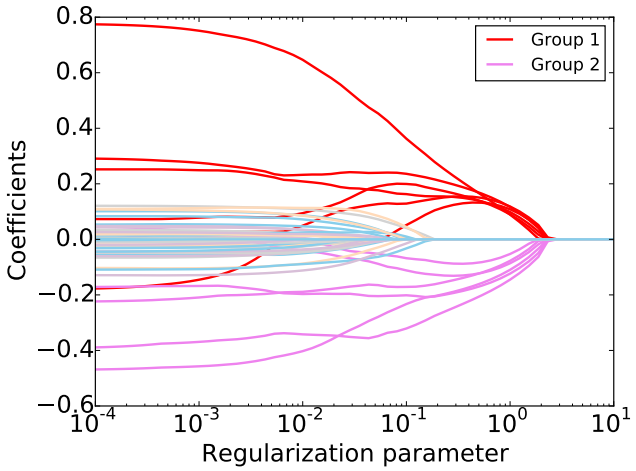


Figure 8: Coefficients of the lasso, ridge-regression and elastic-net estimate for a linear regression problem where the response only depends on two groups of 6 predictors each out of a total of 50 predictors. The predictors in each group are highly correlated.

β_{lasso} *is a solution of the lasso, then*

$$\beta(\alpha)_i := \alpha\,\beta_{\text{lasso},i} + (1-\alpha)\;\beta_{\text{lasso},j}, \tag{57}$$

$$\beta(\alpha)_j := (1-\alpha)\;\beta_{\text{lasso},i} + \alpha\,\beta_{\text{lasso},j}, \tag{58}$$

$$\beta(\alpha)_k := \beta_{\text{lasso},k}, \quad k \notin \{i,j\}, \tag{59}$$

is also a solution for any $0 < \alpha < 1$.

The following lemma, which we have borrowed from [6], provides some intuition as to why strictly convex regularization functions such as ridge regression tend to weigh highly correlated predictors in a similar way. This does not contradict the previous result because the lasso is not strictly convex. The result is proved in Section A.8 of the appendix.

Lemma 3.3 (Identical predictors)**.** *Let us consider a regularized least squares problem of the form*

$$\text{minimize} \quad \frac{1}{2n}\left|\left|y - X\tilde{\beta}\right|\right|_2^2 + \lambda\,\mathcal{R}\left(\tilde{\beta}\right) \tag{60}$$

where $\mathcal{R}$ *is an arbitrary regularizer, which is strictly convex and invariant to the ordering of its argument. If two columns of the predictor matrix* X *are identical* $X_i = X_j$ *then the corresponding coefficients in the solution* $\beta_{\mathcal{R}}$ *of the optimization problem are also identical:* $\beta_{\mathcal{R},i} = \beta_{\mathcal{R},j}$.

For sparse regression problems where some predictors are highly correlated ridge regression weighs correlated predictors similarly, as opposed to the lasso, but does not yield a sparse model. The *elastic net* combines the lasso and ridge-regression cost functions introducing an extra regularization parameter α,

$$\text{minimize} \quad \left|\left|y - X\tilde{\beta}\right|\right|_2^2 + \lambda\left(\frac{1-\alpha}{2}\left|\left|\tilde{\beta}\right|\right|_2^2 + \frac{\alpha}{2}\left|\left|\tilde{\beta}\right|\right|_1\right). \tag{61}$$

For $\alpha = 0$ the elastic net is equivalent to ridge regression, whereas for $\alpha = 1$ it's equivalent to the lasso. For intermediate values of α the cost function yields sparse linear models where the coefficients corresponding to highly correlated predictors have similar amplitudes, as shown in Figure 8.

Figure 9 plots the training and test error achieved by least squares, ridge regression, the lasso and the elastic net on a dataset where the response only depends on two groups of highly correlated predictors. The total number of predictors in the dataset is $p = 50$ and the number of training examples is $n = 100$. Least squares overfits the data, yielding the best error for the training set. Ridge regression has a significantly lower test error, but does not achieve the performance of the lasso because it does not yield a sparse model as can be seen in Figure 8. The elastic net achieves the lowest test error.

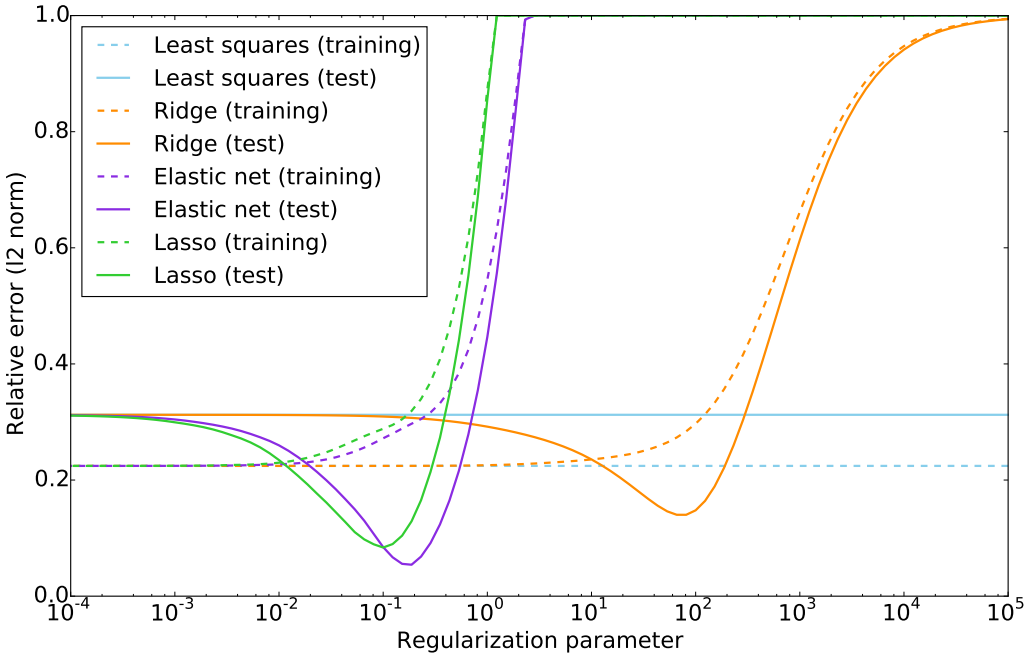


Figure 9: Training and test error achieved by least squares, ridge regression, the lasso and the elastic net on a dataset where the response only depends on two groups of highly correlated predictors. The coefficient paths are shown in Figure 8.

4 Group sparsity

4.1 Introduction

Group sparsity is a generalization of sparsity that allows us to design models that incorporate prior information about the data. If the entries of a vector are partitioned into several groups, then the vector is group sparse if only the entries corresponding to a small number of groups are nonzero, no matter how many entries are zero or nonzero *within the groups*. We have already exploited group sparsity in the context of denoising in Lecture Notes 4. There we used block thresholding to enforce the prior assumption that the STFT coefficients in a speech signal tend to have significant amplitude in contiguous areas. In this section, we will focus on the application of group-sparsity assumptions to regression models, where the groups are used to encode information about the structure of the predictors.

We consider a linear regression model where the predictors are partitioned into k groups $\mathcal{G}_1$, $\mathcal{G}_2$, $\ldots$, $\mathcal{G}_k$,

$$y \approx \beta_0 + X\beta = \beta_0 + \begin{bmatrix} X_{\mathcal{G}_1} & X_{\mathcal{G}_2} & \cdots & X_{\mathcal{G}_m} \end{bmatrix} \begin{bmatrix} \beta_{\mathcal{G}_1} \\ \beta_{\mathcal{G}_2} \\ \cdots \\ \beta_{\mathcal{G}_k} \end{bmatrix}. \tag{62}$$

A group-sparse regression model is a model in which only the predictors corresponding to a small number of groups are used to predict the response. For such models, the coefficient vector β has a group-sparse structure, since the entries corresponding to the rest of the groups are equal to zero. For example, if the predictors include the temperature, air pressure and other weather conditions at several locations, it might make sense to assume that the response will only depend on the predictors associated to a small number of locations. This implies that the β should be group sparse, where each group contains the predictors associated to a particular location.

4.2 Multi-task learning

Multi-task learning is a problem in machine learning and statistics which consists of learning models for several learning problems simultaneously, exploiting common structure. In the case of regression, an important example is when we want to estimate several responses Y_1, Y_2, $\ldots$, $Y_k \in \mathbb{R}^n$ that depend on the same predictors X_1, X_2, $\ldots$, $X_p \in \mathbb{R}^n$. To learn a linear

model for this problem we need to fit a matrix of coefficients $B \in \mathbb{R}^{p \times k}$,

$$Y = \begin{bmatrix} Y_1 & Y_2 & \dots & Y_k \end{bmatrix} \approx B_0 + XB \tag{63}$$

$$= B_0 + X \begin{bmatrix} B_1 & B_2 & \cdots & B_k \end{bmatrix}. \tag{64}$$

If we estimate B by solving a least-squares problem, then this is exactly equivalent to learning k linear regression separately, one for each response. However, a reasonable assumption in many cases is that the different responses depend on the *same predictors*, albeit with *with different coefficients*. This corresponds exactly to a group-sparse assumption on the coefficient matrix B: B should have a small number of nonzero rows.

4.3 Mixed ℓ_1/ℓ_2 norm

In order to promote group-sparse structure, a popular approach is to penalize the ℓ_1/ℓ_2 norm, which corresponds to the sum of the ℓ_2-norms of the entries in the different groups. The intuition is that minimizing the ℓ_1 norm induces sparsity, so minimizing the ℓ_1 norm of the ℓ_2-norms of the groups should induce sparsity at the group level.

Definition 4.1 (ℓ_1/ℓ_2 norm)**.** *The ℓ_1/ℓ_2 norm of a vector β with entries divided into k groups $\mathcal{G}_1$, $\mathcal{G}_2$, ..., $\mathcal{G}_k$ is defined as*

$$||\beta||_{1,2} := \sum_{i=1}^{k} ||\beta_{\mathcal{G}_i}||_2 \,. \tag{65}$$

In the case of multitask learning, where the groups correspond to the rows of a matrix, the ℓ_1/ℓ_2 norm of the matrix corresponds to the sum of the ℓ_2 norms of the rows,

$$||B||_{1,2} := \sum_{i=1}^{k} ||B_{:i}||_2 \,, \tag{66}$$

where $B_{:i}$ denotes the ith row of B.

Let us give a simple example that shows why penalizing the ℓ_1/ℓ_2 norm induces group sparsity. Let us define two groups $\mathcal{G}_1 := \{1, 2\}$ and $\mathcal{G}_1 := \{3\}$. The corresponding ℓ_1/ℓ_2 norm is given by

$$\left|\left|\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}\right|\right|_{1,2} = \sqrt{\beta_1^2 + \beta_2^2} + |\beta_3| \tag{67}$$

Our aim is to fit a regression model. Let us imagine that most of the response can be explained by the first predictor, so that

$$y \approx X \begin{bmatrix} \beta_1 \\ 0 \\ 0 \end{bmatrix}, \tag{68}$$

but the fit can be improved in two ways, by setting either β_2 or β_3 to a certain value α. The question is which of the two options is *cheaper* in terms of ℓ_1/ℓ_2 norm, since this is the option that will be chosen if we use the ℓ_1/ℓ_2 norm to regularize the fit. The answer is that modifying β_2 has much less impact on the ℓ_1/ℓ_2 norm

$$\left\| \begin{bmatrix} \beta_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha \\ 0 \end{bmatrix} \right\|_{1,2} = \sqrt{\beta_1^2 + \alpha^2}, \tag{69}$$

$$\left\| \begin{bmatrix} \beta_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \alpha \end{bmatrix} \right\|_{1,2} = \sqrt{\beta_1^2 + \alpha^2 + 2\left|\beta_1\right|\alpha}, \tag{70}$$

especially if α is small or β_1 is large. The ℓ_1/ℓ_2 norm induces a group-sparse structure by making it less costly to include entries that belong to groups which already have nonzero entries, with respect to groups where all the entries are equal to zero.

The following lemma derives the subgradient of the ℓ_1/ℓ_2 norm. We defer the proof to Section A.9 in the appendix.

Lemma 4.2 (Subgradient of the ℓ_1/ℓ_2 norm)**.** *A vector $g \in \mathbb{R}^p$ is a subgradient of the ℓ_1/ℓ_2 norm at $\beta \in \mathbb{R}^p$ if and only if*

$$g_{\mathcal{G}_i} = \frac{\beta_{\mathcal{G}_i}}{||\beta_{\mathcal{G}_i}||_2} \quad \text{for } \beta_{\mathcal{G}_i} \neq 0, \tag{71}$$

$$||g_{\mathcal{G}_i}||_2 \leq 1 \quad \text{for } \beta_{\mathcal{G}_i} = 0. \tag{72}$$

4.4 Group and multitask lasso

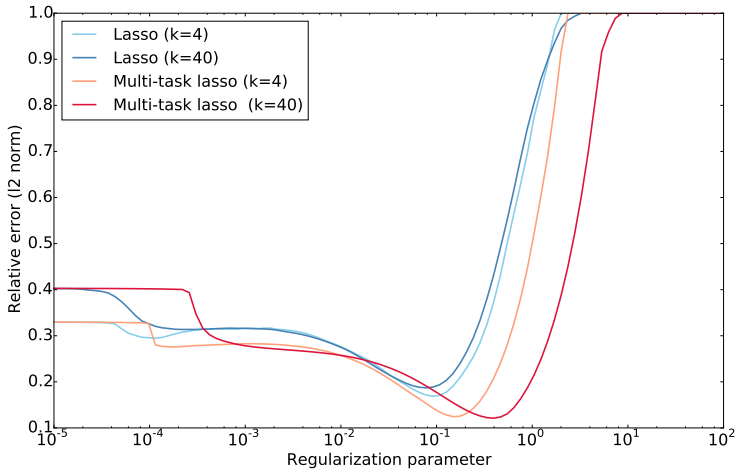
The group lasso [5] combines a least-squares term with an ℓ_1/ℓ_2-norm regularization term to fit a group-sparse linear regression model,

$$\text{minimize} \quad \left|\left|Y - \tilde{\beta}_0 - X\tilde{\beta}\right|\right|_{\text{F}}^2 + \lambda \left|\left|\tilde{\beta}\right|\right|_{1,2}, \tag{73}$$

where $\lambda > 0$ is a regularization parameter. Applying the exact same idea to the multi-task regression problem yields the multi-task lasso

$$\text{minimize} \quad \left|\left|Y - \tilde{B}_0 - X\tilde{B}\right|\right|_{\text{F}}^2 + \lambda \left|\left|\tilde{B}\right|\right|_{1,2}, \tag{74}$$

$s = 4$



$s = 30$

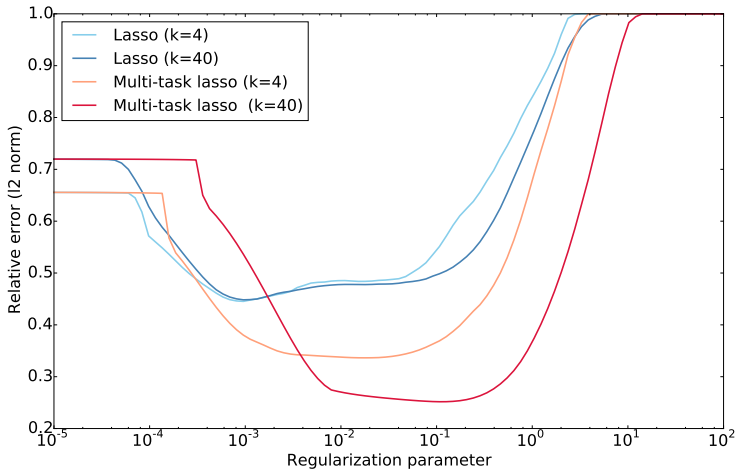


Figure 10: Errors achieved by the lasso and the multitask lasso on a multitask regression problem where the same s predictors (out of a total of $p = 100$ predictors) are used to produce the response in k sparse linear regression models. The training data is equal to $n = 50$.

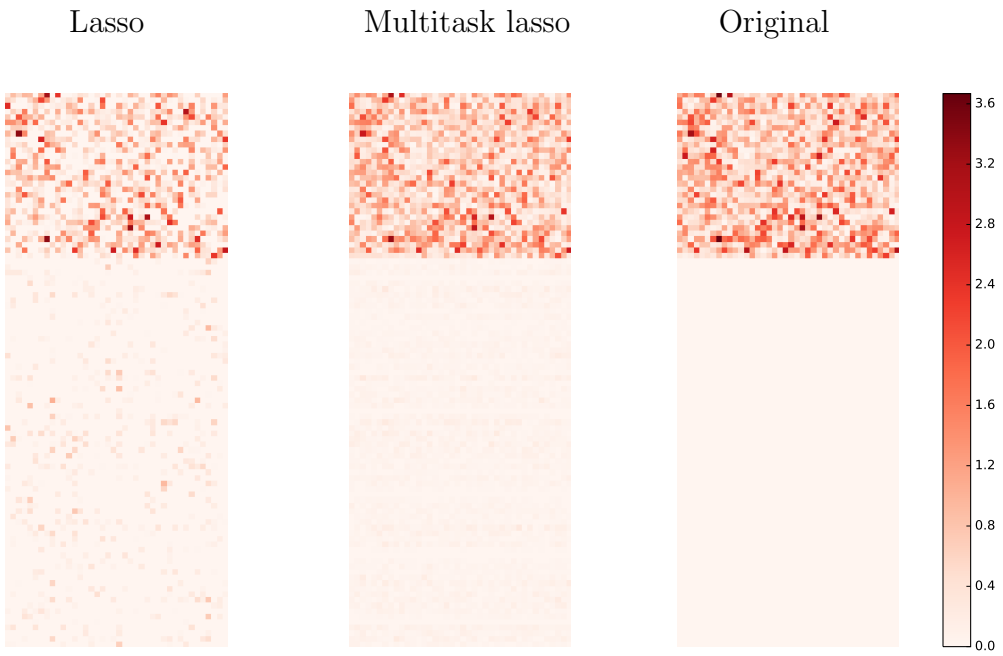


Figure 11: Coefficients for the lasso and the multitask lasso when $s = 30$ and $k = 40$ in the experiment described in Figure 10. The first 30 rows contain the relevant features.

where the Frobenius norm $||\cdot||_{\mathrm{F}}$ is equivalent to the ℓ_2 norm of the matrix once it is vectorized.

Figure 10 shows a comparison between the errors achieved by the lasso and the multitask lasso on a multitask regression problem where the same s predictors (out of a total of $p = 100$ predictors) are used to produce the response in k sparse linear regression models. The training data is equal to $n = 50$. The lasso fits each of the models separately, whereas the multitask lasso produces a joint fit. This allows to learn the model more effectively and achieve a lower error, particularly when the number of predictors is relatively large ($s = 30$). Figure 11 shows the actual coefficients fit by the lasso and the multitask lasso when $s = 30$ and $k = 40$. The multitask lasso is able to promote a group sparse structure which results in the correct identification of the relevant predictors. In contrast, the lasso fits sparse models that do not necessarily contain the same predictors, making it easier to include irrelevant predictors that seem relevant for a particular response because of the noise.

4.5 Proximal-gradient algorithm

In order to apply the group or the multitask lasso we need to solve a least-squares problem with an ℓ_1/ℓ_2-norm. In this section we adapt the proximal-gradient algorithm described in Lectures Notes 3 to this setting. The first step is to derive the proximal operator of this

norm.

Proposition 4.3 (Proximal operator of the ℓ_1/ℓ_2 norm)**.** *The solution to the optimization problem*

$$\text{minimize} \quad \frac{1}{2}\left|\left|\beta - \tilde{\beta}\right|\right|_2^2 + \alpha \left|\left|\tilde{\beta}\right|\right|_{1,2}, \tag{75}$$

where $\alpha > 0$, is obtained by applying a block soft-thresholding operator to β

$$\text{prox}_{\alpha ||\cdot||_{1,2}}(\beta) = \mathcal{BS}_\alpha(\beta), \tag{76}$$

where

$$\mathcal{BS}_\alpha(\beta)_{\mathcal{G}_i} := \begin{cases} \beta_{\mathcal{G}_i} - \alpha \frac{\beta_{\mathcal{G}_i}}{||\beta_{\mathcal{G}_i}||_2} & \text{if } ||\beta_{\mathcal{G}_i}||_2 \geq \alpha \\ 0 & \text{otherwise.} \end{cases} \tag{77}$$

The proof of this result is in Section A.10 of the appendix.

The proximal-gradient method alternates between block-thresholding and taking a gradient step to minimize the least-squares fit.

Algorithm 4.4 (Iterative Block-Thresholding Algorithm)**.** *We set the initial point $x^{(0)}$ to an arbitrary value in $\mathbb{R}^n$. Then we compute*

$$x^{(k+1)} = \mathcal{BS}_{\alpha_k \lambda}\left(x^{(k)} - \alpha_k A^T\left(Ax^{(k)} - y\right)\right), \tag{78}$$

until a convergence criterion is satisfied.

Convergence may be accelerated using the ideas discussed in Lecture Notes 3 to motivate the FISTA method.

References

The book by Hastie, Tibshirani and Wainwright [3] is a great reference on sparse regression. We also recommend [1].

[1] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning.*

[2] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[3] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC Press, 2015.

[4] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[5] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[6] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

A Proofs

A.1 Proof of Proposition 1.1

Let $X = U\Sigma V_T$ be the singular-value decomposition (SVD) of X. Under the conditions of the proposition, $\left(X^T X\right)^{-1} X^T y = V\Sigma U^T$. We begin by separating y into two components

$$y = UU^T y + \left(I - UU^T\right) y \tag{79}$$

where $UU^T y$ is the projection of y onto the column space of X. Note that $\left(I - UU^T\right) y$ is orthogonal to the column space of X and consequently to both $UU^T y$ and $X\tilde{\beta}$ for any $\tilde{\beta}$. By Pythagoras's Theorem

$$\left|\left|y - X\tilde{\beta}\right|\right|_2^2 = \left|\left|\left(I - UU^T\right) y\right|\right|_2^2 + \left|\left|UU^T y - X\tilde{\beta}\right|\right|_2^2. \tag{80}$$

The minimum value of this cost function that can be achieved by optimizing over $\tilde{beta}$ is $||y_{X^\perp}||_2^2$. This can be achieved by solving the system of equations

$$UU^T y = X\tilde{\beta} = U\Sigma V_T \tilde{\beta}. \tag{81}$$

Since $U^T U = I$ because $p \geq n$, multiplying both sides of the equality yields the equivalent system

$$U^T y = \Sigma V_T \tilde{\beta}. \tag{82}$$

Since X is full rank, Σ and V are square and invertible (and by definition of the SVD $V^{-1} = V^T$), so

$$\beta_{\text{ls}} = V\Sigma U^T y \tag{83}$$

is the unique solution to the system and consequently also of the least-squares problem.

A.2 Proof of Proposition 1.3

We model the noise as a random vector $\check{z}$ which has entries that are independent Gaussian random variables with mean zero and a certain variance σ. Note that $\check{z}$ is a random vector, whereas z is the *realization* of the random vector. Similarly, the data y that we observe is interpreted as a realization of a random vector $\check{y}$. The ith entry of

$$\check{y} = X\beta + \check{z} \tag{84}$$

is a Gaussian random variable with mean $(X\beta)_i$ and variance σ^2. The pdf of $\check{y}_i$ is consequently of the form

$$f_{\check{y}_i}(t) := \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(t - \left(X\tilde{\beta}\right)_i\right)^2}{2\sigma^2} \right). \tag{85}$$

By assumption, the entries of $\check{z}$ are independent, so the joint pdf of $\check{y}$ is equal to

$$f_{\check{y}}(\tilde{y}) := \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(\tilde{y}_i - \left(X\tilde{\beta}\right)_i\right)^2}{2\sigma^2} \right) \tag{86}$$

$$:= \frac{1}{\sqrt{(2\pi)^n}\sigma^n} \exp\left(-\frac{1}{2\sigma^2} \left|\left| \tilde{y} - X\tilde{\beta} \right|\right|_2^2 \right). \tag{87}$$

The likelihood is the probability density function of $\check{y}$ evaluated at the observed data y and interpreted as a function of $\tilde{\beta}$.

$$\mathcal{L}\left(\tilde{\beta}\right) = \frac{1}{\sqrt{(2\pi)^n}} \exp\left(-\frac{1}{2} \left|\left| y - X\tilde{\beta} \right|\right|_2^2 \right). \tag{88}$$

Since the function is nonnegative and the logarithm is a monotone function, we can take optimize over the logarithm of the likelihood to find the maximum-likelihood estimate. We conclude that it is given by the solution to the least-squares problem, since

$$\beta_{\mathrm{ML}} = \arg\max_{\tilde{\beta}} \mathcal{L}\left(\tilde{\beta}\right) \tag{89}$$

$$= \arg\max_{\tilde{\beta}} \log \mathcal{L}\left(\tilde{\beta}\right) \tag{90}$$

$$= \arg\min_{\tilde{\beta}} \left|\left| y - X\tilde{\beta} \right|\right|_2 \tag{91}$$

A.3 Proof of Lemma 1.4

Let 1 denote an n-dimensional vector of ones. The model with an intercept is equivalent to

$$y \approx \begin{bmatrix} X & 1 \end{bmatrix} \begin{bmatrix} \tilde{\beta} \\ \tilde{\beta}_0 \end{bmatrix}. \tag{92}$$

Applying Proposition 1.1 the least-squares fit is

$$\begin{aligned}
\begin{bmatrix} \beta \\ \beta_0 \end{bmatrix}_{\text{ls}} &= \left(\begin{bmatrix} X & 1 \end{bmatrix}^T \begin{bmatrix} X & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} X & 1 \end{bmatrix}^T y && (93) \\
&= \begin{bmatrix} X^T X & X^T 1 \\ 1^T X & n \end{bmatrix}^{-1} \begin{bmatrix} X^T y \\ 1^T y \end{bmatrix} && (94) \\
&= \begin{bmatrix} X^T X & 0 \\ 0 & n \end{bmatrix}^{-1} \begin{bmatrix} X^T y \\ 0 \end{bmatrix} && (95) \\
&= \begin{bmatrix} \left(X^T X\right)^{-1} & 0 \\ 0 & \frac{1}{n} \end{bmatrix} \begin{bmatrix} X^T y \\ 0 \end{bmatrix} && (96) \\
&= \begin{bmatrix} \left(X^T X\right)^{-1} X^T y \\ 0 \end{bmatrix}, && (97)
\end{aligned}$$

where have used the fact that $1^T y = 0$ and $X^T 1 = 0$ because the mean of y and of each of the columns of X is equal to zero.

A.4 Proof of Theorem 1.5

By Proposition 1.1

$$\begin{aligned}
||\beta - \beta_{\text{ls}}||_2^2 &= \left|\left|\beta - V\Sigma^{-1}U^T y\right|\right|_2^2 && (98) \\
&= \left|\left|\beta - V\Sigma^{-1}U^T \left(X\beta + z\right)\right|\right|_2^2 && (99) \\
&= \left|\left|V\Sigma^{-1}U^T z\right|\right|_2^2 && (100) \\
&= \left|\left|\Sigma^{-1}U^T z\right|\right|_2^2 && (101) \\
&= \sum_{j=1}^{p} \left(\frac{U_j^T z}{\sigma_j} \right)^2, && (102)
\end{aligned}$$

which implies

$$\frac{\left|\left|U^T z\right|\right|_2^2}{\sigma_{\max}^2} \leq ||\beta - \beta_{\text{ls}}||_2^2 \leq \frac{\left|\left|U^T z\right|\right|_2^2}{\sigma_{\min}^2}. \tag{103}$$

The distribution of the p-dimensional vector $U^T z$ is Gaussian with mean zero and covariance matrix

$$U^T \, \Sigma_z \, U = \sigma_z^2 I, \tag{104}$$

where $\Sigma_z = \sigma_z^2 I$ is the covariance of matrix of z. As a result, $\frac{1}{\sigma_z^2}\left|\left|U^T z\right|\right|_2^2$ is a chi-square random variable with p degrees of freedom. By Proposition A.2 in Lecture Notes 5 and the union bound

$$p\sigma_z^2 \, (1-\epsilon) \leq \left|\left|U^T z\right|\right|_2^2 \leq p\sigma_z^2 \, (1+\epsilon) \tag{105}$$

with probability at least $1 - \exp\left(-\frac{p\epsilon^2}{8}\right) - \exp\left(-\frac{p\epsilon^2}{2}\right) \geq 1 - 2\exp\left(-\frac{p\epsilon^2}{8}\right)$.

A.5 Proof of Theorem 2.3

We define the error

$$h := \beta - \beta_{\text{lasso}}. \tag{106}$$

The following lemma shows that the error satisfies the robust-sparsity condition in the definition of the restricted-eigenvalue property.

Lemma A.1. *In the setting of Theorem 2.3 $h := \beta - \beta_{\text{lasso}}$ satisfies*

$$||h_{T^c}||_1 \leq ||h_T||_1 \tag{107}$$

where T is the support of the nonzero entries of β.

Proof. Since β_{lasso} is feasible and β is supported on T

$$\begin{aligned}
\tau = ||\beta||_1 &\geq ||\beta_{\text{lasso}}||_1 && (108)\\
&= ||\beta + h||_1 && (109)\\
&= ||\beta + h_T||_1 + ||h_{T^c}||_1 && (110)\\
&\geq ||\beta||_1 - ||h_T||_1 + ||h_{T^c}||_1 . && (111)
\end{aligned}$$

□

This implies that by the restricted-eigenvalue property we have

$$||h||_2^2 \leq \frac{1}{\gamma n} ||X\ h||_2^2 . \tag{112}$$

The following lemma allows to bound the right-hand side.

Lemma A.2. *In the setting of Theorem 2.3*

$$||Xh||_2^2 \leq 2\, z^T X h. \tag{113}$$

Proof. Because β_{lasso} is the solution to the constrained optimization problem and β is also feasible we have

$$||y - X\beta_{\text{lasso}}||_2^2 \geq ||y - X\beta_{\text{lasso}}||_2^2 . \tag{114}$$

Substituting $y = X\beta + z$,

$$||z - Xh||_2^2 \geq ||z||_2^2 \tag{115}$$

which implies the result. □

Since $||h_{T^c}||_1 \leq ||h_T||_1$ and h_T only has s nonzero entries

$$||h||_1 \leq 2\sqrt{s}\, ||h||_2 \tag{116}$$

so by Lemma A.2, Hölder's inequality and (112)

$$||h||_2^2 \leq \frac{2\, z^T X h}{\gamma n} \tag{117}$$

$$\leq \frac{2\ \left|\left|X^T z\right|\right|_\infty ||h||_1}{\gamma n} \tag{118}$$

$$\leq \frac{4\sqrt{s}\, ||h||_2 \left|\left|X^T z\right|\right|_\infty}{\gamma n}. \tag{119}$$

The proof of the theorem is completed by the following lemma, that uses the assumption on the noise z to bound $\left|\left|X^T z\right|\right|_\infty$.

Lemma A.3. *In the setting of Theorem 2.3*

$$\mathrm{P}\left(\left|\left|X^T z\right|\right|_\infty > \sigma_z \sqrt{2\alpha\ n \log p}\right) \leq 2 \exp\left(-\left(\alpha - 1\right) \log p\right) \tag{120}$$

for any $\alpha > 2$.

Proof. $X_i^T z$ is Gaussian with variance $\sigma_z^2 \left|\left|X_i\right|\right|_2^2$, so for $t > 0$ by Lemma 3.5 in Lecture Notes 5

$$\mathrm{P}\left(\left|X_i^T z\right| > t\sigma_z \left|\left|X_i\right|\right|_2\right) \leq 2\exp\left(-\frac{t^2}{2}\right) \tag{121}$$

By the union bound,

$$\mathrm{P}\left(\left|\left|X^T z\right|\right|_\infty > t\sigma_z \max_i \left|\left|X_i\right|\right|_2\right) \leq 2\, p \exp\left(-\frac{t^2}{2}\right) \tag{122}$$

$$= 2\exp\left(-\frac{t^2}{2} + \log p\right) \tag{123}$$

Choosing $t = \sqrt{2\alpha \log p}$ for $\alpha > 2$ we obtain the desired result. □

A.6 Proof of Proposition 3.1

The ridge-regression cost function is equivalent to the least-squares cost function

$$\text{minimize} \quad \left|\left|\begin{bmatrix} y \\ 0 \end{bmatrix} - \begin{bmatrix} X \\ \lambda I \end{bmatrix} \tilde{\beta}\right|\right|_2^2. \tag{124}$$

By Proposition 1.1 the solution to this problem is

$$\beta_{\text{ridge}} := \left(\begin{bmatrix} X \\ \lambda I \end{bmatrix}^T \begin{bmatrix} X \\ \lambda I \end{bmatrix}\right)^{-1} \begin{bmatrix} X \\ \lambda I \end{bmatrix}^T \begin{bmatrix} y \\ 0 \end{bmatrix} \tag{125}$$

$$= \left(X^T X + \lambda^2 I\right)^{-1} X^T \left(X\beta + z\right) \tag{126}$$

$$= \left(V\Sigma^2 V^T + \lambda^2 V V^T\right)^{-1} \left(V\Sigma^2 V^T \beta + V\Sigma U^T z\right) \tag{127}$$

$$= V\left(\Sigma^2 + \lambda^2 I\right)^{-1} V^T \left(V\Sigma^2 V^T \beta + V\Sigma U^T z\right). \tag{128}$$

A.7 Proof of Lemma 3.2

Since $X_i = X_j$,

$$X\beta\left(\alpha\right) = \left(\alpha\, \beta_{\text{lasso},i} + (1-\alpha)\ \beta_{\text{lasso},j}\right) X_i + \left((1-\alpha)\ \beta_{\text{lasso},i} + \alpha\, \beta_{\text{lasso},j}\right) X_j + \sum_{k \notin \{i,j\}} \beta_{\text{lasso},k} X_k$$

$$= \beta_{\text{lasso},i} X_i + \beta_{\text{lasso},j} X_j + \sum_{k \notin \{i,j\}} \beta_{\text{lasso},k} X_k \tag{129}$$

$$= X\beta_{\text{lasso}}, \tag{130}$$

which implies $||y - X\beta(\alpha)||_2 = ||y - X\beta_{\text{lasso}}||_2$. Similarly,

$$||\beta(\alpha)||_1 = |\alpha\,\beta_{\text{lasso},i} + (1-\alpha)\;\beta_{\text{lasso},j}| + |(1-\alpha)\;\beta_{\text{lasso},i} + \alpha\,\beta_{\text{lasso},j}| + \sum_{k\notin\{i,j\}} |\beta_{\text{lasso},k}| \tag{131}$$

$$\leq \alpha\,|\beta_{\text{lasso},i}| + (1-\alpha)\,|\beta_{\text{lasso},j}| + (1-\alpha)\,|\beta_{\text{lasso},i}| + \alpha\,|\beta_{\text{lasso},j}| + \sum_{k\notin\{i,j\}} |\beta_{\text{lasso},k}|$$

$$= ||\beta_{\text{lasso}}||_1\,. \tag{132}$$

This implies that $\beta(\alpha)$ must also be a solution.

A.8 Proof of Lemma 3.3

Consider

$$\beta(\alpha)_i := \alpha\,\beta_{\mathcal{R},i} + (1-\alpha)\;\beta_{\mathcal{R},j}, \tag{133}$$

$$\beta(\alpha)_j := (1-\alpha)\;\beta_{\mathcal{R},i} + \alpha\,\beta_{\mathcal{R},j}, \tag{134}$$

$$\beta(\alpha)_k := \beta_{\mathcal{R},k}, \quad k \notin \{i,j\}\,. \tag{135}$$

for $0 < \alpha < 1$. By the same argument in (130) $||y - X\beta(\alpha)||_2 = ||y - X\beta_{\mathcal{R}}||_2$. We define

$$\beta'_{\mathcal{R},i} := \beta_{\mathcal{R},j}, \tag{136}$$

$$\beta'_{\mathcal{R},j} := \beta_{\mathcal{R},i}, \tag{137}$$

$$\beta'_{\mathcal{R},k} := \beta_{\mathcal{R},k}, \quad k \notin \{i,j\}\,. \tag{138}$$

Note that because $\mathcal{R}$ is invariant to the ordering of its argument $\mathcal{R}(\beta_{\mathcal{R}}) = \mathcal{R}(\beta'_{\mathcal{R}})$. Since $\beta(\alpha) = \alpha\beta_{\mathcal{R}} + (1-\alpha)\,\beta'_{\mathcal{R}}$, by strict convexity of $\mathcal{R}$

$$\mathcal{R}(\beta(\alpha)) < \alpha\;\mathcal{R}(\beta_{\mathcal{R}}) + (1-\alpha)\,\mathcal{R}(\beta'_{\mathcal{R}}) \tag{139}$$

$$= \mathcal{R}(\beta_{\mathcal{R}}) \tag{140}$$

if $\beta_{\mathcal{R},i} \neq \beta_{\mathcal{R},j}$. Since this would mean that $\beta_{\mathcal{R}}$ is not a solution to the regularization problem, this implies that $\beta_{\mathcal{R},i} = \beta_{\mathcal{R},j}$.

A.9 Proof of Lemma 4.2

We have that

$$||\beta + h||_{1,2} \geq ||\beta||_{1,2} + g^T h \tag{141}$$

for all possible $h \in \mathbb{R}^p$ if and only if

$$||\beta_{\mathcal{G}_i} + h_{\mathcal{G}_i}||_2 \geq ||\beta_{\mathcal{G}_i}||_2 + g_{\mathcal{G}_i}^T h_{\mathcal{G}_i} \tag{142}$$

for all possible $h_{\mathcal{G}_i} \in \mathbb{R}^{|\mathcal{G}_i|}$, for $1 \leq i \leq k$.

If $\beta_{\mathcal{G}_i} \neq 0$ the only vector $g_{\mathcal{G}_i}$ that satisfies (142) is the gradient of the ℓ_2 norm at $\beta_{\mathcal{G}_i}$

$$\nabla ||\cdot||_2 (\beta_{\mathcal{G}_i}) = \frac{\beta_{\mathcal{G}_i}}{||\beta_{\mathcal{G}_i}||_2}. \tag{143}$$

The fact that the gradient is of this form follows from the chain rule.

If $\beta_{\mathcal{G}_i} = 0$ any vector $g_{\mathcal{G}_i}$ with ℓ_2 norm bounded by one satisfies (142) by the Cauchy-Schwarz inequality.

A.10 Proof of Lemma 4.3

We can separate the minimization problem into the different groups

$$\min_{\tilde{\beta}} \frac{1}{2} \left|\left|\beta - \tilde{\beta}\right|\right|_2^2 + \alpha \left|\left|\tilde{\beta}\right|\right|_{1,2} = \sum_{i=1}^{k} \min_{\tilde{\beta}_{\mathcal{G}_i}} \frac{1}{2} \left|\left|\beta_{\mathcal{G}_i} - \tilde{\beta}_{\mathcal{G}_i}\right|\right|_2^2 + \alpha \left|\left|\tilde{\beta}_{\mathcal{G}_i}\right|\right|_2. \tag{144}$$

We can therefore minimize the different terms of the sum separately. Each term

$$\frac{1}{2} \left|\left|\beta_{\mathcal{G}_i} - \tilde{\beta}_{\mathcal{G}_i}\right|\right|_2^2 + \alpha \left|\left|\tilde{\beta}_{\mathcal{G}_i}\right|\right|_2 \tag{145}$$

is convex and has subgradients of the form

$$g\left(\tilde{\beta}_{\mathcal{G}_i}\right) := \tilde{\beta}_{\mathcal{G}_i} - \beta_{\mathcal{G}_i} + \alpha\, q\left(\tilde{\beta}_{\mathcal{G}_i}\right), \tag{146}$$

$$q\left(\tilde{\beta}_{\mathcal{G}_i}\right) := \begin{cases} \frac{\tilde{\beta}_{\mathcal{G}_i}}{||\tilde{\beta}_{\mathcal{G}_i}||_2} & \text{if } \tilde{\beta}_{\mathcal{G}_i} \neq 0, \\ 0 \quad \text{if } \tilde{\beta}_{\mathcal{G}_i} \neq 0. \end{cases} \tag{147}$$

This follows from Lemma 4.2 and the fact that the sum of subgradients of several functions is a subgradient of their sum.

Any minimizer $\hat{\beta}_{\mathcal{G}_i}$ of (145) must satisfy

$$g\left(\hat{\beta}_{\mathcal{G}_i}\right) = 0. \tag{148}$$

This implies that if $\hat{\beta}_{\mathcal{G}_i} \neq 0$ then

$$\beta_{\mathcal{G}_i} = \hat{\beta}_{\mathcal{G}_i} + \frac{\alpha\, \hat{\beta}_{\mathcal{G}_i}}{\left|\left|\hat{\beta}_{\mathcal{G}_i}\right|\right|_2} \tag{149}$$

$$= \left(\left|\left|\hat{\beta}_{\mathcal{G}_i}\right|\right|_2 + \alpha\right) \frac{\hat{\beta}_{\mathcal{G}_i}}{\left|\left|\hat{\beta}_{\mathcal{G}_i}\right|\right|_2}. \tag{150}$$

As a result, $\beta_{\mathcal{G}_i}$ and $\hat{\beta}_{\mathcal{G}_i}$ are collinear and

$$\left|\left|\hat{\beta}_{\mathcal{G}_i}\right|\right|_2 = ||\beta_{\mathcal{G}_i}||_2 - \alpha, \tag{151}$$

which can only hold if $||\beta_{\mathcal{G}_i}||_2 \geq \alpha$. In that case,

$$\hat{\beta}_{\mathcal{G}_i} = \beta_{\mathcal{G}_i} - \frac{\alpha\, \hat{\beta}_{\mathcal{G}_i}}{\left|\left|\hat{\beta}_{\mathcal{G}_i}\right|\right|_2} \tag{152}$$

$$= \beta_{\mathcal{G}_i} - \frac{\alpha\, \beta_{\mathcal{G}_i}}{||\beta_{\mathcal{G}_i}||_2}. \tag{153}$$

This establishes that as long as $||\beta_{\mathcal{G}_i}||_2 \geq \alpha$ (153) is a solution to the proximal problem.

If $\hat{\beta}_{\mathcal{G}_i} = 0$ then by (148)

$$\alpha \geq \left|\left|\hat{\beta}_{\mathcal{G}_i} - \beta_{\mathcal{G}_i}\right|\right|_2 \tag{154}$$

$$= ||\beta_{\mathcal{G}_i}||_2 \,. \tag{155}$$

This establishes that as long as $||\beta_{\mathcal{G}_i}||_2 \leq \alpha$ $\hat{\beta}_{\mathcal{G}_i} = 0$ is a solution to the proximal problem.

Learning signal representations

1 Introduction

In Lecture Notes 4 we described how to design representations that allow to represent signals with a small number of coefficients and how these sparse representation can be leveraged to compress and denoise signals. In these notes, we will consider the problem of learning a representation from a set of signals. In more detail, we assume that we have available a dataset of n signals $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ and our goal is to compute a small set of atoms $\Phi_1, \ldots, \Phi_k \in \mathbb{R}^d$ that allow to build a linear model for each signal

$$x_j \approx \sum_{i=1}^{k} \Phi_i \, A_{ij}, \quad 1 \leq j \leq n, \tag{1}$$

where $A_1, \ldots, A_n \in \mathbb{R}^k$ are coefficient vectors. Ideally, k should be significantly smaller than n, which means that the atoms are able to capture the common structure of the n signals. The learned atoms can be used for compression, denoising or as features for classification, whereas the coefficients in A may allow to cluster the different signals, as we will discuss later on.

If we group the signals into a matrix $X := \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}$, learning the atoms and the corresponding coefficients is equivalent to a matrix-factorization problem,

$$X \approx \begin{bmatrix} \Phi_1 & \Phi_2 & \cdots & \Phi_k \end{bmatrix} \begin{bmatrix} A_1 & A_2 & \cdots & A_n \end{bmatrix} = \Phi \, A \tag{2}$$

where $\Phi \in \mathbb{R}^{d \times k}$, $A \in \mathbb{R}^{k \times n}$. The assumption that k is smaller than n means that our goal is to approximate X with a low-rank matrix.

Learning signal representation is significantly more challenging than fitting models once the atoms are known. The reason is that any optimization problem of the form

$$\underset{\widetilde{\Phi}, \widetilde{A}}{\text{minimize}} \quad \left|\left| X - \widetilde{\Phi} \, \widetilde{A} \right|\right| + \mathcal{R}_1 \left(\widetilde{\Phi} \right) + \mathcal{R}_2 \left(\widetilde{A} \right), \tag{3}$$

where $||\cdot||$ is an arbitrary norm and $\mathcal{R}_1$ and $\mathcal{R}_2$ are regularization terms, is nonconvex due to the product between the variables. Figure 1 illustrates this by plotting the function $f(\phi, a) = (1 - \phi a)^2$ (i.e. there is only one 1D atom and one coefficient) which has minima along two separate lines: $(\alpha, 1/\alpha)$ and $(-\alpha, -1/\alpha)$ for any $\alpha > 0$. In higher dimensions, the number of local minima grows. Interestingly, if no regularization is added, it is possible to find a global minimum for the nonconvex problem by computing the singular-value decomposition, as we

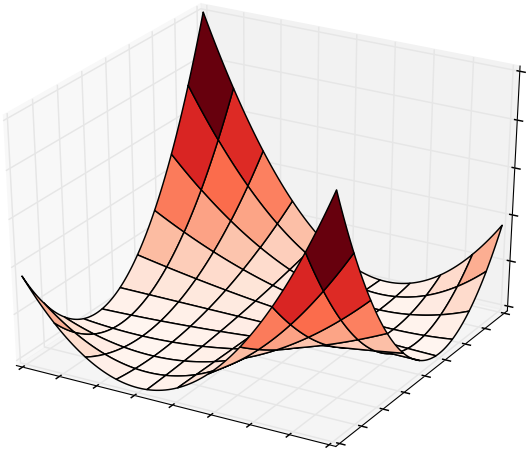


Figure 1: The function $f(\phi, a) = (1 - \phi a)^2$ has minima at $(\alpha, 1/\alpha)$ and $(-\alpha, -1/\alpha)$ for any $\alpha > 0$.

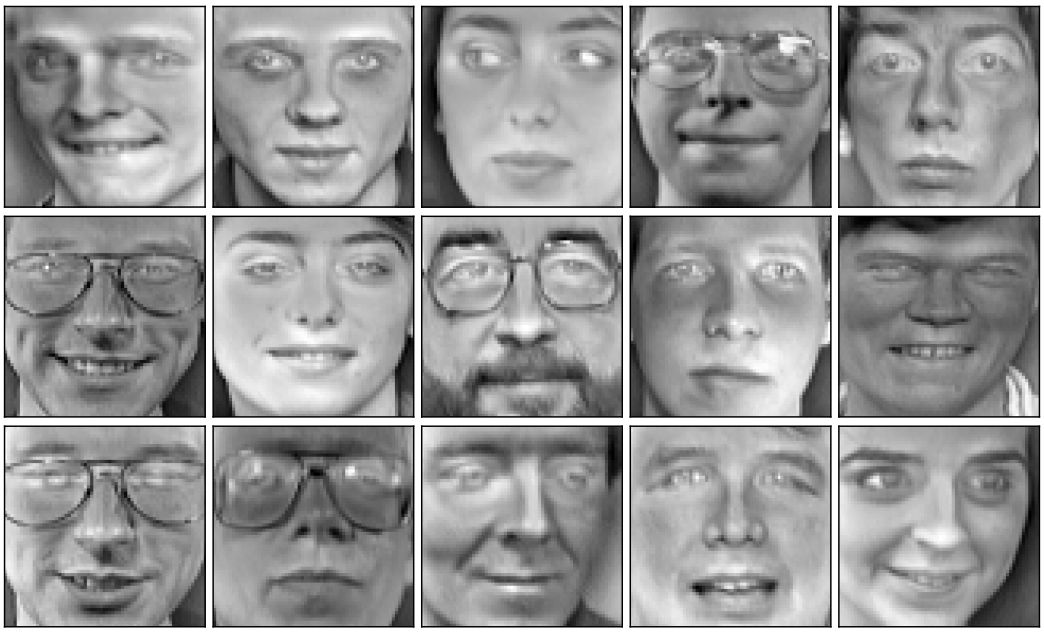


Figure 2: Examples from the face dataset. The dataset contains 10 face pictures of 40 different people.

discuss below, but this is no longer the case when we add regularization terms to the cost function.

Figure 2 shows some examples from a dataset that we will use to illustrate different models. It contains 10 face pictures of 40 different people, which were taken between April 1992 and April 1994 at AT&T Laboratories Cambridge[1].

2 K means

A possible way of representing a set of signals using a small number of atoms is to cluster the signals into several groups and assign an atom to each group. This clustering problem is typically known as k-means clustering. The aim is to learn k atoms $\Phi_1, \ldots, \Phi_k$ that minimize the cost function

$$\sum_{i=1}^{n} \left|\left|x_i - \Phi_{c(i)}\right|\right|_2^2, \tag{4}$$

$$c\left(i\right) := \arg\min_{1 \leq j \leq k} \left|\left|x_i - \Phi_j\right|\right|_2. \tag{5}$$

In words, $c\left(i\right)$ denotes the index of the atom that is assigned to the signal x_i.

After carrying out the clustering, the learned model can be expressed as factorization of the matrix X where the coefficients for each signal are restricted to just select one of the atoms,

$$X \approx \begin{bmatrix} \Phi_1 & \Phi_2 & \cdots & \Phi_k \end{bmatrix} \begin{bmatrix} e_{c(1)} & e_{c(2)} & \cdots & e_{c(n)} \end{bmatrix}, \tag{6}$$

where e_j is the standard-basis vector (all its entries are zero, except the jth entry which equals one).

Minimizing the cost function (4) over the choice of atoms and coefficients is computationally hard (NP-hard). However minimizing the cost function if the atoms or the coefficients are known is very easy. If the atoms are known, for each signal we just need to select the atom that is nearest. If the coefficients are known then the following lemma, proved in Section A.1 of the appendix, shows that the optimal atoms just correspond to the mean of each cluster.

Lemma 2.1. *For any $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ the solution to the optimization problem*

$$\underset{\phi}{\text{minimize}} \quad \sum_{i=1}^{n} \left|\left|x_i - \phi\right|\right|_2^2 \tag{7}$$

is equal to

$$\hat{\phi} = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{8}$$

[1] See http://scikit-learn.org/stable/datasets/olivetti_faces.html

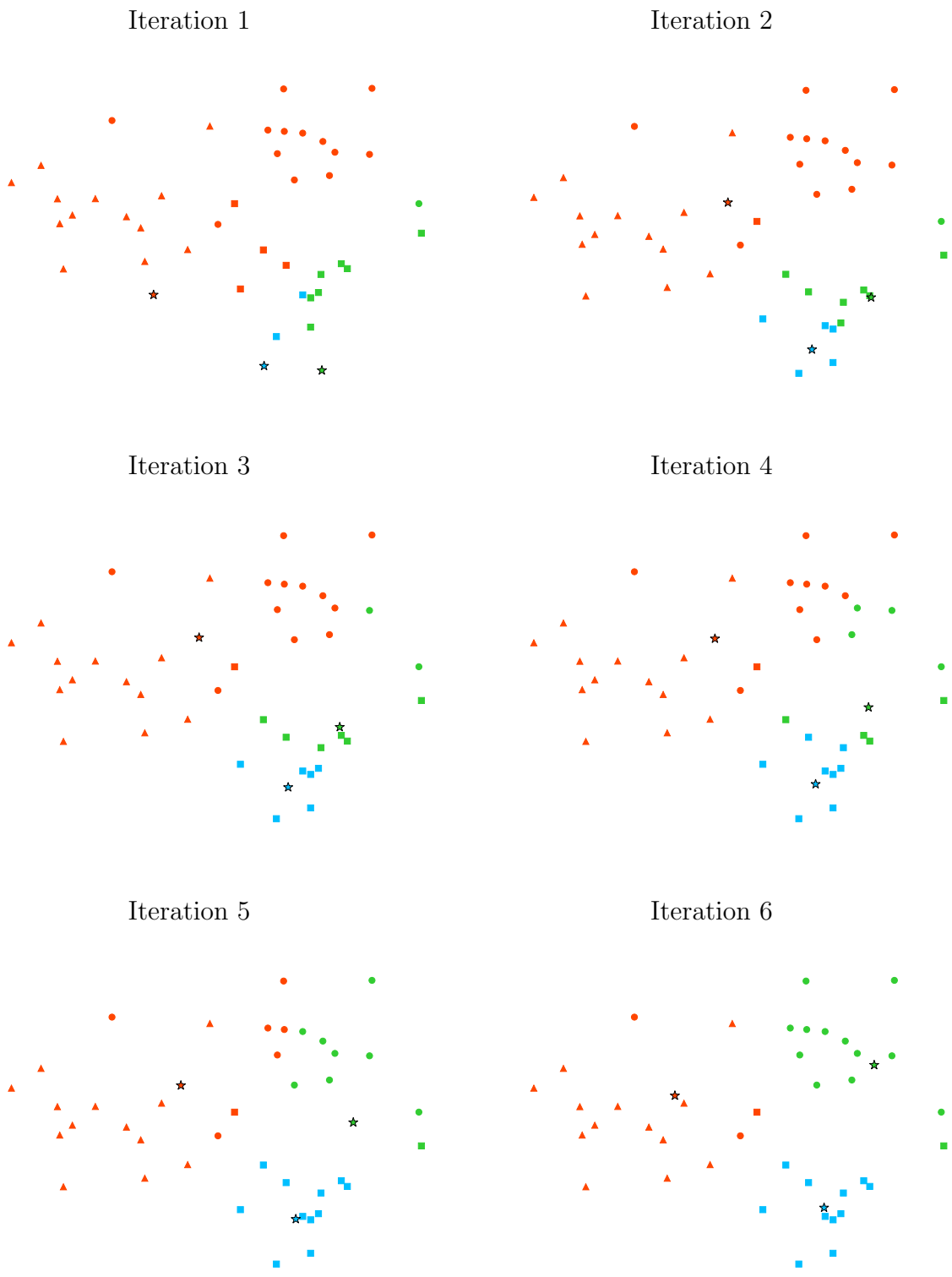


Figure 3: Iterations of Lloyd's algorithm for $k = 3$ clusters which are colored orange, blue and green. The stars indicate the atoms corresponding to each cluster. The original dataset corresponds to three clusters represented by the circles, triangles and squares.

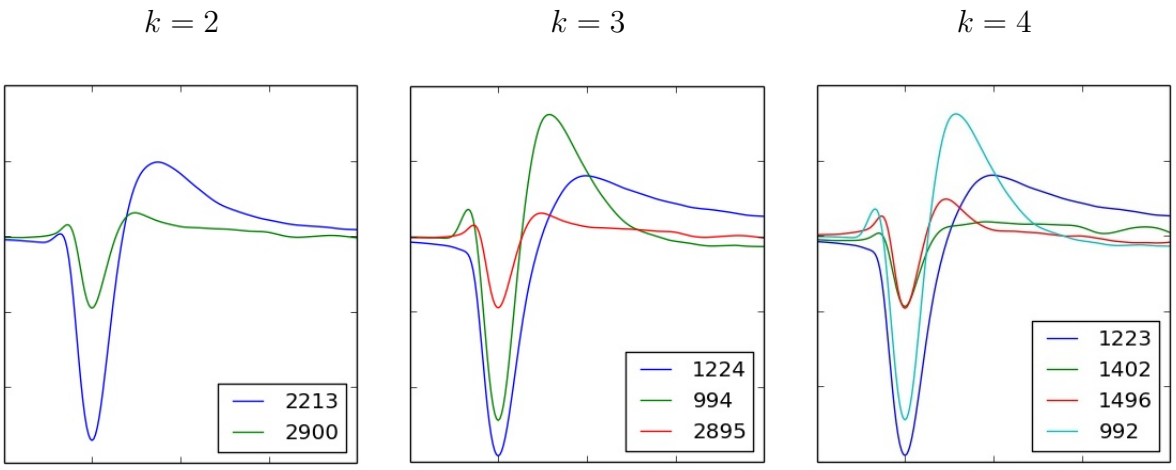


Figure 4: Results of applying k means for different values of k to neural data measured by the Chichilnisky lab at Stanford.

Lloyd's algorithm is a heuristic for the k-means problem that alternates between solving the two subproblems.

Algorithm 2.2 (Lloyd's algorithm)**.** *We initialize the atoms randomly and then alternate between updating the coefficients and the atoms.*

- *The assignment step selects the closest atom to each signal,*

$$c\left(i\right) := \arg\min_{1 \leq j \leq k} \left|\left|x_i - \Phi_j\right|\right|_2 . \tag{9}$$

- *The averaging step computes the mean of each cluster,*

$$\Phi_j := \frac{\sum_{i=1}^{n} \delta\left(c\left(j\right) = i\right)\, x_i}{\sum_{i=1}^{n} \delta\left(c\left(j\right) = i\right)}, \tag{10}$$

  where $\delta\left(c\left(j\right) = i\right)$ equals one if $c\left(j\right) = i$ and zero otherwise.

In practice, Lloyd's algorithm often finds a satisfactory solution to the k-means problem, although for some initializations it might get stuck in a bad local minimum (an atom may end up with no assignments for example). To avoid such situations we can run the method several times, using different random initializations, and select the best solution. Figure 3 illustrates Lloyd's algorithm by showing several iterations.

We end this section describing two applications of k means: one in neuroscience and another in image processing.

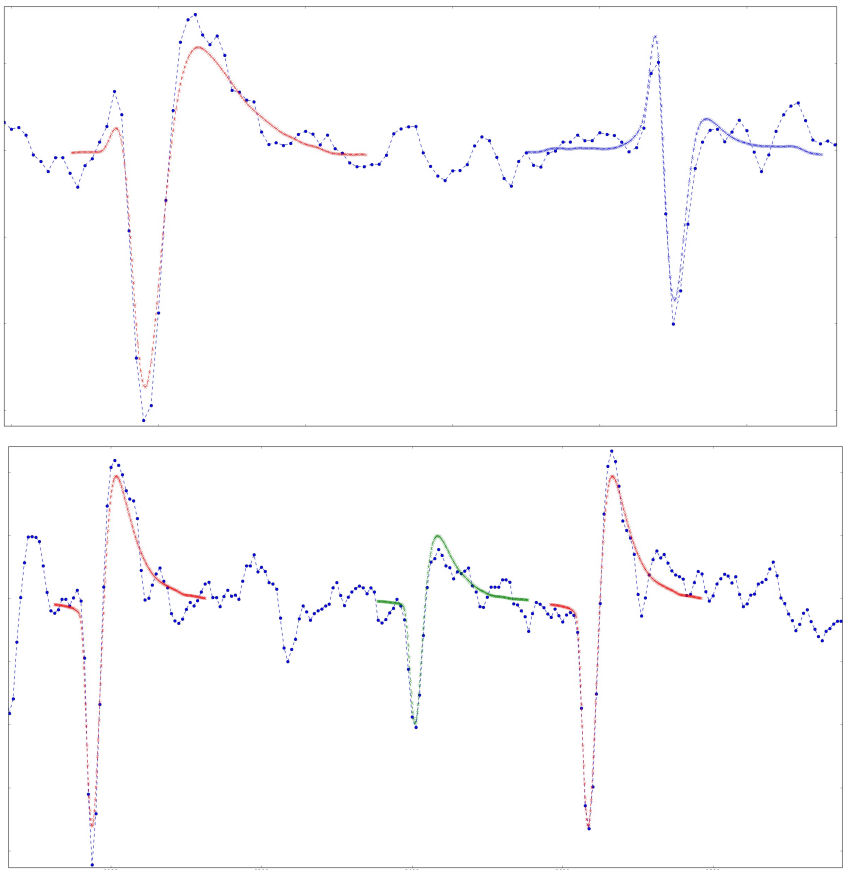


Figure 5: Atoms obtained by solving the k means problem superposed on the original neural data. The data was obtained by the Chichilnisky lab at Stanford.

In neuroscience, data is often collected by placing an array of electric sensors on tissue containing neurons. Each sensors measures the electric activity of several neurons that can be well modeled as sequences of spikes. These spikes have different shapes depending on what neuron generated them. The spike-sorting problem consists of clustering the measured spikes to assign them to the different cells. If we extract all spiking events from the data and align them adequately, we can apply solve the k means problem for these signals to sort the spikes (note that we have to fix the number of neurons k beforehand). Figure 4 shows the results of applying this procedure with different values of k to some real data measured by the Chichilnisky lab at Stanford. The atoms learnt using k means can then be used to obtain a fit to the original data as shown in Figure 5

Figures 6 and 7 show the results of solving the k means problem for the faces dataset from Figure 2 with different values of k. If k is small with respect to the number of different people in the dataset, which is equal to 40, then the atoms learnt by k means correspond to averaging the faces of different people. As we increase k, the atoms become sharper because

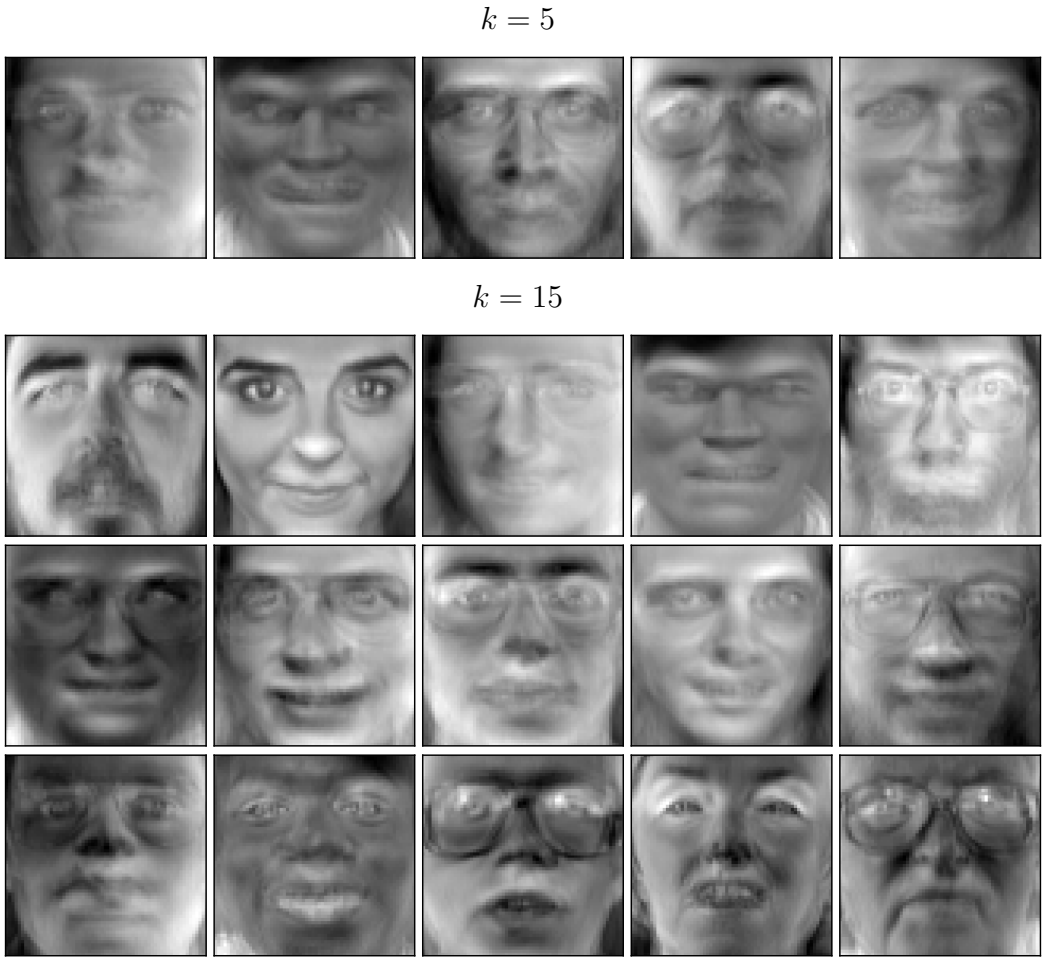


Figure 6: Atoms obtained by solving the k means problem for the faces dataset from Figure 2 with $k = 5$ and $k = 15$.

they are obtained by averaging less images, and some of them correspond to just one person.

3 Principal-component analysis

We already discussed principal-component analysis (PCA) as a dimensionality-reduction technique in Lecture Notes 5. In this section we review the method, providing a geometric and probabilistic interpretation. Then we explain how to learn signal representations using PCA and end by describing an application to collaborative filtering.

$k = 40$

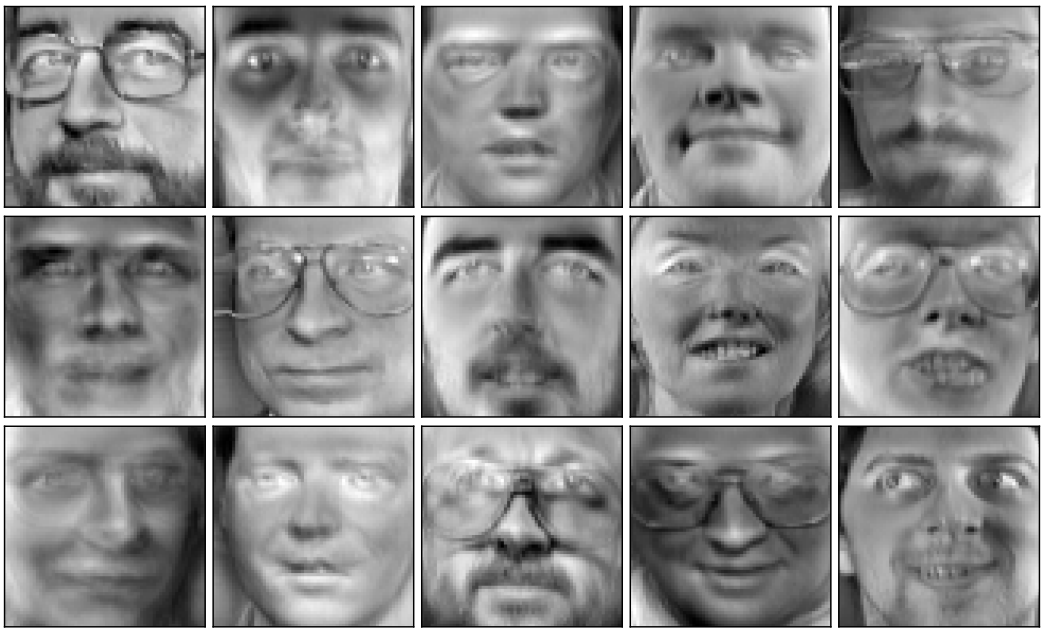


Figure 7: Atoms obtained by solving the k means problem for the faces dataset from Figure 2 with $k = 40$.

3.1 Algorithm

PCA allows to find directions in this space along which the data have a high variation. This is achieved by centering the data and then extracting the singular vectors corresponding to the largest singular values.

Algorithm 3.1 (Principal component analysis)**.** *Given n data vectors $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n \in \mathbb{R}^d$, we apply the following steps.*

1. *Center the data,*

$$x_i = \tilde{x}_i - \frac{1}{n} \sum_{i=1}^{n} \tilde{x}_i, \qquad 1 \leq i \leq n. \tag{11}$$

2. *Group the centered data in a data matrix $X \in \mathbb{R}^{d \times n}$*

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}. \tag{12}$$

3. *Compute the singular-value decomposition (SVD) of X and extract the left singular vectors corresponding to the k largest singular values. These are the first k principal components.*

3.2 PCA: Geometric interpretation

Once the data are centered, the energy of the projection of the data points onto different directions in the ambient space reflects the variation of the dataset along those directions. PCA selects the directions that maximize the ℓ_2 norm of the projection and are mutually orthogonal. The sum of the squared ℓ_2 norms of the projection of the centered data $x_1, x_2, \ldots, x_n$ onto a 1D subspace spanned by a unit-norm vector u can be expressed as

$$\sum_{i=1}^{n} \left|\left|\mathcal{P}_{\text{span}(u)}\, x_i\right|\right|_2^2 = \sum_{i=1}^{n} u^T x_i x_i^T u \tag{13}$$

$$= u^T X X^T u \tag{14}$$

$$= \left|\left|X^T u\right|\right|_2^2. \tag{15}$$

If we want to maximize the energy of the projection onto a subspace of dimension k, an option is to choose orthogonal 1D projections sequentially. First we choose a unit vector U_1 that maximizes $\left|\left|X^T u\right|\right|_2^2$ and is consequently the 1D subspace that is better adapted to the data. Then, we choose a second unit vector U_2 *orthogonal* to the first which maximizes $\left|\left|X^T u\right|\right|_2^2$ and hence is the 1D subspace that is better adapted to the data while being in

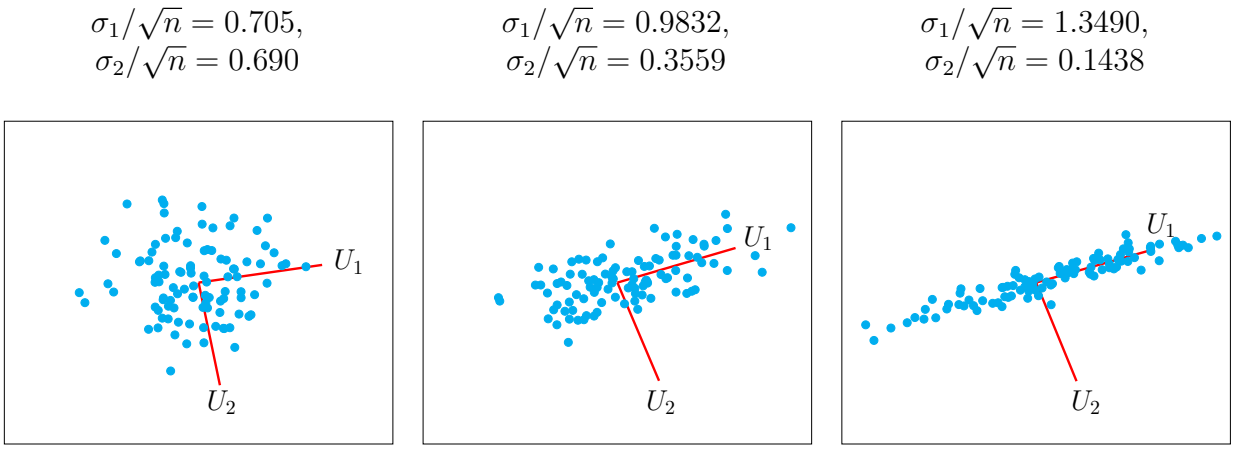


Figure 8: PCA of a dataset with $n = 100$ 2D vectors with different configurations. The two first singular values reflect how much energy is preserved by projecting onto the two first principal components.

the orthogonal complement of U_1. We repeat this procedure until we have k orthogonal directions, which are the first k principal components.

More formally, the left singular vectors $U_1, U_2, \ldots, U_k$ and the corresponding singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k$, are given by

$$\sigma_1 = \max_{||u||_2=1} \left|\left|X^T u\right|\right|_2, \tag{16}$$

$$U_1 = \arg\max_{||u||_2=1} \left|\left|X^T u\right|\right|_2, \tag{17}$$

$$\sigma_j = \max_{\substack{||u||_2=1 \\ u \perp u_1,\ldots,u_{j-1}}} \left|\left|X^T u\right|\right|_2, \quad 2 \leq j \leq k, \tag{18}$$

$$U_j = \arg\max_{\substack{||u||_2=1 \\ u \perp u_1,\ldots,u_{j-1}}} \left|\left|X^T u\right|\right|_2, \quad 2 \leq j \leq k. \tag{19}$$

This is established in Lemma A.1 of Lecture Notes 5.

Figure 8 provides an example in 2D. Note how each singular value is proportional to the energy that lies in the direction of the corresponding principal component.

PCA is equivalent to choosing the *best* (in terms of ℓ_2 norm) k 1D subspaces following a *greedy* procedure, since at each step we choose the best 1D subspace orthogonal to the previous ones. A natural question to ask is whether this method produces the best k-dimensional subspace. A priori this is not necessarily the case; many greedy algorithms produce suboptimal results. However, in this case the greedy procedure is indeed optimal: the subspace spanned by the first k principal components is the *best* subspace we can choose in terms of the ℓ_2-norm of the projections. The following result is borrowed from Lecture Notes 5 (Theorem 2.5).

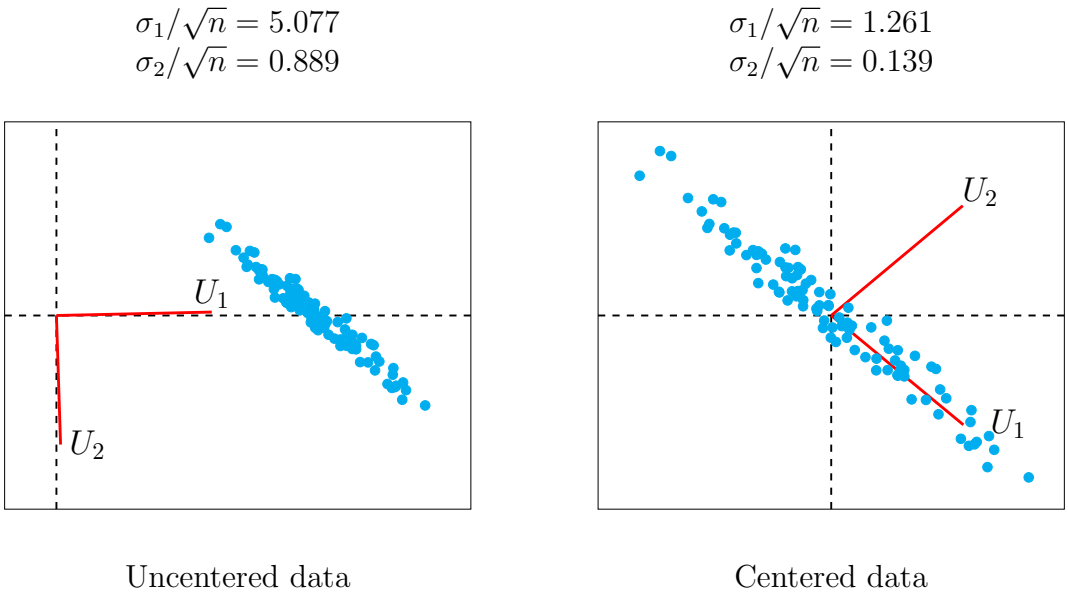


Figure 9: PCA applied to $n = 100$ 2D data points. On the left the data are not centered. As a result the dominant principal component U_1 lies in the direction of the mean of the data and PCA does not reflect the actual structure. Once we center, U_1 becomes aligned with the direction of maximal variation.

Theorem 3.2. *For any matrix $X \in \mathbb{R}^{d \times n}$ with left singular vectors $U_1, U_2, \ldots, U_n$ corresponding to the singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$, if we fix any $k \leq \min\{d, n\}$*

$$\sum_{i=1}^{n} \left|\left|\mathcal{P}_{\text{span}(U_1, U_2, \ldots, U_k)}\, x_i\right|\right|_2^2 \geq \sum_{i=1}^{n} ||\mathcal{P}_{\mathcal{S}}\, x_i||_2^2 \,, \tag{20}$$

for any subspace $\mathcal{S}$ of dimension k.

Figure 9 illustrates the importance of centering before applying PCA. Theorems **??** and 3.2 still hold if the data are not centered. However, the norm of the projection onto a certain direction no longer reflects the variation of the data. In fact, if the data are concentrated around a point that is far from the origin, the first principal component will tend be aligned in that direction. This makes sense as projecting onto that direction captures more energy. As a result, the principal components do not capture the directions of maximum variation *within* the cloud of data.

3.3 PCA: Probabilistic interpretation

Let us interpret our data, $x_1, x_2, \ldots, x_n$ in $\mathbb{R}^m$, as samples of a random vector $\mathbf{x}$ of dimension m. Recall that we are interested in determining the directions of maximum variation of the

data in ambient space. In probabilistic terms, we want to find the directions in which the data have *higher variance*. The covariance matrix of the data provides this information. In fact, we can use it to determine the variance of the data in any direction.

Lemma 3.3. *Let u be a unit vector,*

$$\operatorname{Var}\left(\boldsymbol{x}^T u\right) = u^T \Sigma_{\boldsymbol{x}} u. \tag{21}$$

Proof.

$$\begin{aligned}
\operatorname{Var}\left(\mathbf{x}^T u\right) &= \mathrm{E}\left(\left(\mathbf{x}^T u\right)^2\right) - \mathrm{E}^2\left(\mathbf{x}^T u\right) && (22)\\
&= \mathrm{E}\left(u\mathbf{x}\mathbf{x}^T u\right) - \mathrm{E}\left(u^T \mathbf{x}\right)\mathrm{E}\left(\mathbf{x}^T u\right) && (23)\\
&= u^T\left(\mathrm{E}\left(\mathbf{x}\mathbf{x}^T\right) - \mathrm{E}\left(\mathbf{x}\right)\mathrm{E}\left(\mathbf{x}\right)^T\right) u && (24)\\
&= u^T \Sigma_{\mathbf{x}} u. && (25)
\end{aligned}$$

□

Of course, if we only have access to samples of the random vector, we do not know the covariance matrix of the vector. However we can approximate it using the empirical covariance matrix.

Definition 3.4 (Empirical covariance matrix)**.** *The empirical covariance of the vectors $x_1, x_2, \ldots, x_n$ in $\mathbb{R}^m$ is equal to*

$$\begin{aligned}
\overline{\Sigma}_n &:= \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \overline{x}_n\right)\left(x_i - \overline{x}_n\right)^T && (26)\\
&= \frac{1}{n} X X^T, && (27)
\end{aligned}$$

where $\overline{x}_n$ is the sample mean, as defined in Definition 1.3 of Lecture Notes 4, and X is the matrix containing the centered data as defined in (12)*.*

If we assume that the mean of the data is zero (i.e. that the data have been centered using the true mean), then the empirical covariance is an unbiased estimator of the true covariance matrix:

$$\begin{aligned}
\mathrm{E}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T\right) &= \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}\left(\mathbf{x}_i\mathbf{x}_i^T\right) && (28)\\
&= \Sigma_{\mathbf{x}}. && (29)
\end{aligned}$$

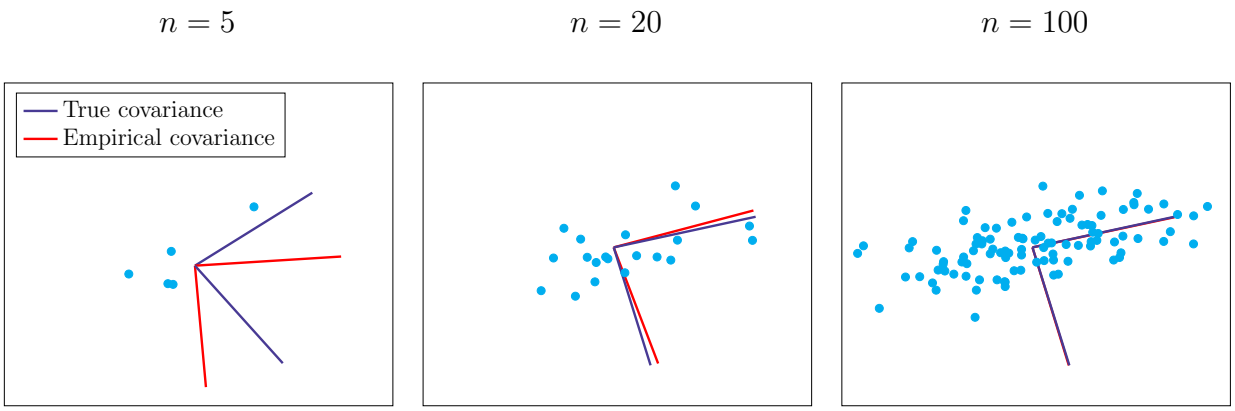


Figure 10: Principal components of n data vectors samples from a 2D Gaussian distribution. The eigenvectors of the covariance matrix of the distribution are also shown.

If the higher moments of the data $\mathrm{E}\left(X_i^2 X_j^2\right)$ and $\mathrm{E}\left(X_i^4\right)$ are finite, by Chebyshev's inequality the entries of the empirical covariance matrix converge to the entries of the true covariance matrix. This means that in the limit

$$\begin{aligned} \mathrm{Var}\left(\mathbf{x}^T u\right) &= u^T \Sigma_{\mathbf{x}} u & (30) \\ &\approx \frac{1}{n} u^T X X^T u & (31) \\ &= \frac{1}{n} \left|\left| X^T u \right|\right|_2^2 & (32) \end{aligned}$$

for any unit-norm vector u. In the limit the principal components correspond to the directions of maximum variance of the underlying random vector. These directions correspond to the eigenvectors of the true covariance matrix, which maximize the quadratic form $u^T \Sigma_{\mathbf{x}} u$. Figure 10 illustrates how the principal components converge to the eigenvectors of $\Sigma_{\mathbf{x}}$.

3.4 PCA as a matrix-approximation method

Let $U \Sigma V^T$ denote the SVD of the data matrix X, which we assume to be centered for simplicity. PCA allows us to obtain a rank k decomposition of the form (2) which is optimal in a certain sense, as shown in the following proposition which is proved in Section A.2 of the appendix.

Proposition 3.5 (Best rank-k approximation)**.** *Let $U \Sigma V^T$ be the SVD of M. We denote by $U_{1:k}$ the matrix that contains the first k left singular vectors of M, $\Sigma_{1:k}$ a $k \times k$ diagonal matrix containing the k largest singular values of X and $V_{1:k}$ the matrix containing the first k right singular vectors. $U_{1:k} \Sigma_{1:k} V_{1:k}^T$ is the best rank-k approximation of M in Frobenius*

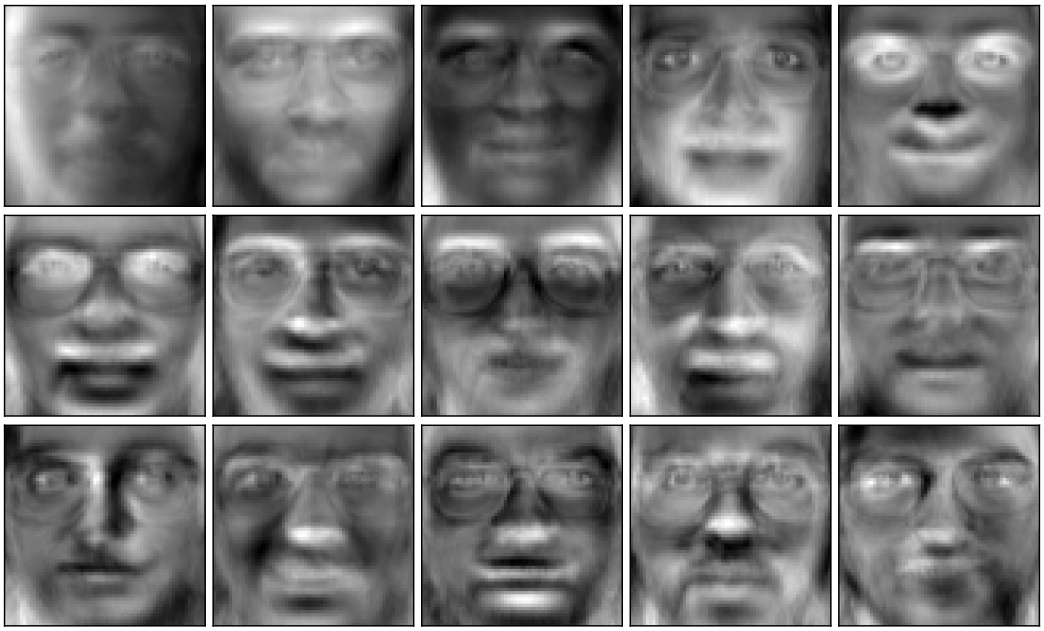


Figure 11: Principal components corresponding to the faces dataset from Figure 2.

norm,

$$U_{1:k}\Sigma_{1:k}V_{1:k}^T = \arg \min_{\left\{\tilde{M} \mid \text{rank}\left(\tilde{M}\right)=k\right\}} \left|\left|M - \tilde{M}\right|\right|_{\text{F}}^2. \tag{33}$$

We can interpret the principal components $\Phi = U_{1:k}$ as orthogonal atoms, whereas $A = \Sigma_{1:k}V_{1:k}^T$ corresponds to the coefficient matrix. Figure 11 shows the result of applying PCA to the faces dataset from Figure 2.

3.5 Collaborative filtering

We now describe an application of PCA to collaborative filtering, where the aim is to pool together information from many users to obtain a model of their behavior. In particular, for movie data, we consider the ratings given by a set of users to a set of movies. If the some of the users have similar tastes, then the ratings will be correlated. PCA allows to uncover this low-rank structure in the data. We demonstrate this through a simple example. Bob,

Molly, Mary and Larry rate the following six movies from 1 to 5,

$$A := \begin{array}{c} \begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} \end{array} \\ \begin{pmatrix} 1 & 1 & 5 & 4 \\ 2 & 1 & 4 & 5 \\ 4 & 5 & 2 & 1 \\ 5 & 4 & 2 & 1 \\ 4 & 5 & 1 & 2 \\ 1 & 2 & 5 & 5 \end{pmatrix} \end{array} \begin{array}{l} \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{array} \tag{34}$$

We subtract the average rating,

$$\mu := \frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}, \tag{35}$$

$$\tag{36}$$

from each entry in the matrix and then compute its singular value decomposition

$$A - \bar{A} = USV^T = U \begin{bmatrix} 7.79 & 0 & 0 & 0 \\ 0 & 1.62 & 0 & 0 \\ 0 & 0 & 1.55 & 0 \\ 0 & 0 & 0 & 0.62 \end{bmatrix} V^T, \tag{37}$$

where

$$\bar{A} := \begin{bmatrix} \mu & \mu & \cdots & \mu \\ \mu & \mu & \cdots & \mu \\ \cdots & \cdots & \cdots & \cdots \\ \mu & \mu & \cdots & \mu \end{bmatrix}. \tag{38}$$

The fact that the first singular value is significantly larger than the rest suggests that the matrix may be well approximated by a rank-1 matrix. This is the case (for ease of comparison the values of A are shown in brackets):

$$\bar{A} + \sigma_1 U_1 V_1^T = \begin{array}{c} \begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} \end{array} \\ \begin{pmatrix} 1.34\,(1) & 1.19\,(1) & 4.66\,(5) & 4.81\,(4) \\ 1.55\,(2) & 1.42\,(1) & 4.45\,(4) & 4.58\,(5) \\ 4.45\,(4) & 4.58\,(5) & 1.55\,(2) & 1.42\,(1) \\ 4.43\,(5) & 4.56\,(4) & 1.57\,(2) & 1.44\,(1) \\ 4.43\,(4) & 4.56\,(5) & 1.57\,(1) & 1.44\,(2) \\ 1.34\,(1) & 1.19\,(2) & 4.66\,(5) & 4.81\,(5) \end{pmatrix} \end{array} \begin{array}{l} \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{array} \tag{39}$$

The first left singular vector is equal to

$$U_1 = \begin{pmatrix} \text{D. Knight} & \text{Spiderman 3} & \text{Love Act.} & \text{B.J.'s Diary} & \text{P. Woman} & \text{Superman 2} \\ -0.45 & -0.39 & 0.39 & 0.39 & 0.39 & -0.45 \end{pmatrix}.$$

It can be interpreted as an atom in the sense that centered scores for each person are proportional to U_1. Alternatively, it can be interpreted as coefficients corresponding to the atoms V_1 below, which allow to cluster the movies into action (+) and romantic (-) movies.

The first right singular vector is equal to

$$V_1 = \begin{pmatrix} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} \\ 0.48 & 0.52 & -0.48 & -0.52 \end{pmatrix}. \tag{40}$$

It can be interpreted as an atom in the sense that centered scores for each movie are proportional to V_1. Alternatively, it can be interpreted as coefficients corresponding to the atom U_1, which allow to cluster the users that have similar tastes (Bob and Molly vs Mary and Larry).

This example is obviously very simple, but it illustrates the interest of low-rank models in collaborative filtering. These models reveal the correlation structure of the data, uncovering hidden factors that determine users' preferences and can be used for clustering. Of course, in practice there will be more than one factor.

4 Nonnegative matrix factorization

4.1 Optimization problem

As explained in the previous section, PCA computes the best low-rank approximation to the data matrix in Frobenius norm. However, depending on the application the atoms obtained from the decomposition are not very interpretable. For example, in Figure 11 the atoms may have negative pixels and the coefficients negative values, so it is difficult to interpret them as face atoms that can be added to form a face. This suggests computing a decomposition where both atoms and coefficients are nonnegative, with the hope that this will allow us to learn a more interpretable model.

A nonnegative matrix factorization of the data matrix may be obtained by solving the

Figure 12: Atoms obtained by applying nonnegative matrix factorization to the faces dataset from Figure 2.

optimization problem,

$$\begin{aligned} \text{minimize} \quad & \left|\left| X - \tilde{\Phi}\ \tilde{A} \right|\right|_{\text{F}}^2 && (41) \\ \text{subject to} \quad & \tilde{\Phi}_{i,j} \geq 0, && (42) \\ & \tilde{A}_{i,j} \geq 0, \qquad \text{for all } i, j && (43) \end{aligned}$$

where $\tilde{\Phi} \in \mathbb{R}^{d \times k}$ and $\tilde{A} \in \mathbb{R}^{k \times n}$ for a fixed k. This is a nonconvex problem which is computationally hard, due to the nonnegative constraint. Several methods to compute local optima have been suggested in the literature, as well as alternative cost functions to replace the Frobenius norm. We refer interested readers to [2]. Figure 12 shows the atoms obtained by applying this method to the faces dataset from Figure 2. Due to the nonnegative constraint, the atoms resemble portions of faces (the black areas have very small values) which capture features such as the eyebrows, the nose, the eyes, etc.

4.2 Topic modeling

Topic modeling aims to learn the thematic structure of a text corpus automatically. We will illustrate this application with a simple example. We take six newspaper articles and compute the frequency of a list of words in each of them. Our final goal is to separate the

words into different clusters that hopefully correspond to different topics. The following matrix contains the counts for each word and article. Each entry contains the number of times that the word corresponding to column j is mentioned in the article corresponding to row i.

$$A = \begin{array}{c} \begin{array}{cccccccccc} \text{singer} & \text{GDP} & \text{senate} & \text{election} & \text{vote} & \text{stock} & \text{bass} & \text{market} & \text{band} & \textit{Articles} \end{array} \\ \begin{pmatrix} 6 & 1 & 1 & 0 & 0 & 1 & 9 & 0 & 8 \\ 1 & 0 & 9 & 5 & 8 & 1 & 0 & 1 & 0 \\ 8 & 1 & 0 & 1 & 0 & 0 & 9 & 1 & 7 \\ 0 & 7 & 1 & 0 & 0 & 9 & 1 & 7 & 0 \\ 0 & 5 & 6 & 7 & 5 & 6 & 0 & 7 & 2 \\ 1 & 0 & 8 & 5 & 9 & 2 & 0 & 0 & 1 \end{pmatrix} \end{array} \begin{array}{c} \textit{Articles} \\ \text{a} \\ \text{b} \\ \text{c} \\ \text{d} \\ \text{e} \\ \text{f} \end{array}$$

Computing the singular-value decomposition of the matrix– after subtracting the mean of each entry as in (37)– we determine that the matrix is approximately low rank

$$A - \bar{A} = USV^T = U \begin{bmatrix} 19.32 & 0 & 0 & 0 & & \\ 0 & 14.46 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.99 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.77 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.67 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.93 \end{bmatrix} V^T. \tag{44}$$

Unfortunately the singular vectors do not have an intuitive interpretation as in Section **??**. In particular, they do not allow to cluster the words

$$\begin{array}{rcccccccc} & & \text{a} & \text{b} & \text{c} & \text{d} & \text{e} & \text{f} \\ U_1 & = & (-0.51 & -0.40 & -0.54 & -0.11 & -0.38 & -0.38) \\ U_2 & = & (\ 0.19 & -0.45 & -0.19 & -0.69 & -0.2 & -0.46) \\ U_3 & = & (\ 0.14 & -0.27 & -0.09 & -0.58 & -0.69 & -0.29) \end{array} \tag{45}$$

or the articles

$$\begin{array}{rcccccccccc} & & \text{singer} & \text{GDP} & \text{senate} & \text{election} & \text{vote} & \text{stock} & \text{bass} & \text{market} & \text{band} \\ V_1 & = & (-0.38 & 0.05 & 0.4 & 0.27 & 0.4 & 0.17 & -0.52 & 0.14 & -0.38) \\ V_2 & = & (\ 0.16 & -0.46 & 0.33 & 0.15 & 0.38- & 0.49 & 0.1 & -0.47 & 0.12\ ) \\ V_3 & = & (-0.18 & -0.18 & -0.04 & -0.74 & -0.05 & 0.11 & -0.1 & -0.43 & -0.43) \end{array} \tag{46}$$

A problem here is that the singular vectors have negative entries that are difficult to interpret. In the case of rating prediction, negative ratings mean that a person does not like a movie. In contrast articles either are about a topic or they are not: it makes sense to add atoms

corresponding to different topics to approximate the word count of a document but not to subtract them. Following this intuition, we apply nonnegative matrix factorization to obtain two matrices $W \in \mathbb{R}^{m\times k}$ and $H \in \mathbb{R}^{k\times n}$ such that

$$M \approx WH, \quad W_{i,j} \geq 0, \qquad 1 \leq i \leq m,\ 1 \leq j \leq k, \tag{47}$$

$$H_{i,j} \geq 0, \qquad 1 \leq i \leq k,\ 1 \leq i \leq n. \tag{48}$$

In our example, we set $k = 3$. H_1, H_2 and H_3 can be interpreted as word-count atoms, but also as coefficients that weigh the contribution of W_1, W_2 and W_3.

$$\begin{array}{cccccccccccc} & & \text{singer} & \text{GDP} & \text{senate} & \text{election} & \text{vote} & \text{stock} & \text{bass} & \text{market} & \text{band} & \\ H_1 & = & (\ 0.34 & 0 & 3.73 & 2.54 & 3.67 & 0.52 & 0 & 0.35 & 0.35\ ) & \\ H_2 & = & (\ 0 & 2.21 & 0.21 & 0.45 & 0 & 2.64 & 0.21 & 2.43 & 0.22\ ) & \\ H_3 & = & (\ 3.22 & 0.37 & 0.19 & 0.2 & 0 & 0.12 & 4.13 & 0.13 & 3.43\ ) & \end{array} \tag{49}$$

The latter interpretation allows to cluster the words into topics. The first topic corresponds to the entries that are not zero (or very small) in H_1: senate, election and vote. The second corresponds to H_2: GDP, stock and market. The third corresponds to H_3: singer, bass and band.

The entries of W allow to assign the topics to articles. b, e and f are about politics (topic 1), d and e about economics (topic 3) and a and c about music (topic 3)

$$\begin{array}{cccccccc} & & \text{a} & \text{b} & \text{c} & \text{d} & \text{e} & \text{f} \\ W_1 & = & (0.03 & 2.23 & 0 & 0 & 1.59 & 2.24) \\ W_2 & = & (\ 0.1 & 0 & 0.08 & 3.13 & 2.32 & 0\ ) \\ W_3 & = & (2.13 & 0 & 2.22 & 0 & 0 & 0.03) \end{array} \tag{50}$$

Finally, we check that the factorization provides a good fit to the data. The product WH is equal to

$$\begin{array}{c} \begin{array}{ccccccccc} \text{singer} & \text{GDP} & \text{senate} & \text{election} & \text{vote} & \text{stock} & \text{bass} & \text{market} & \text{band} \end{array} \\ \begin{pmatrix} 6.89\ (6) & 1.01\ (1) & 0.53\ (1) & 0.54\ (0) & 0.10\ (0) & 0.53\ (1) & 8.83\ (9) & 0.53\ (0) & 7.36\ (8) \\ 0.75\ (1) & 0\ (0) & 8.32\ (9) & 5.66\ (5) & 8.18\ (8) & 1.15\ (1) & 0\ (0) & 0.78\ (1) & 0.78\ (0) \\ 7.14\ (8) & 0.99\ (1) & 0.44\ (0) & 0.47\ (1) & 0\ (0) & 0.47\ (0) & 9.16\ (9) & 0.48\ (1) & 7.62\ (7) \\ 0\ (0) & 7\ (6.91) & 0.67\ (1) & 1.41\ (0) & 0\ (0) & 8.28\ (9) & 0.65\ (1) & 7.60\ (7) & 0.69\ (0) \\ 0.53\ (0) & 5.12\ (5) & 6.45\ (6) & 5.09\ (7) & 5.85\ (5) & 6.97\ (6) & 0.48\ (0) & 6.19\ (7) & 1.07\ (2) \\ 0.86\ (1) & 0.01\ (0) & 8.36\ (8) & 5.69\ (5) & 8.22\ (9) & 1.16\ (2) & 0.14\ (0) & 0.79\ (0) & 0.9\ (1) \end{pmatrix} \end{array} \begin{array}{c} \textit{Art.} \\ \text{a} \\ \text{b} \\ \text{c} \\ \text{d} \\ \text{e} \\ \text{f} \end{array}$$

For ease of comparison the values of A are shown in brackets.

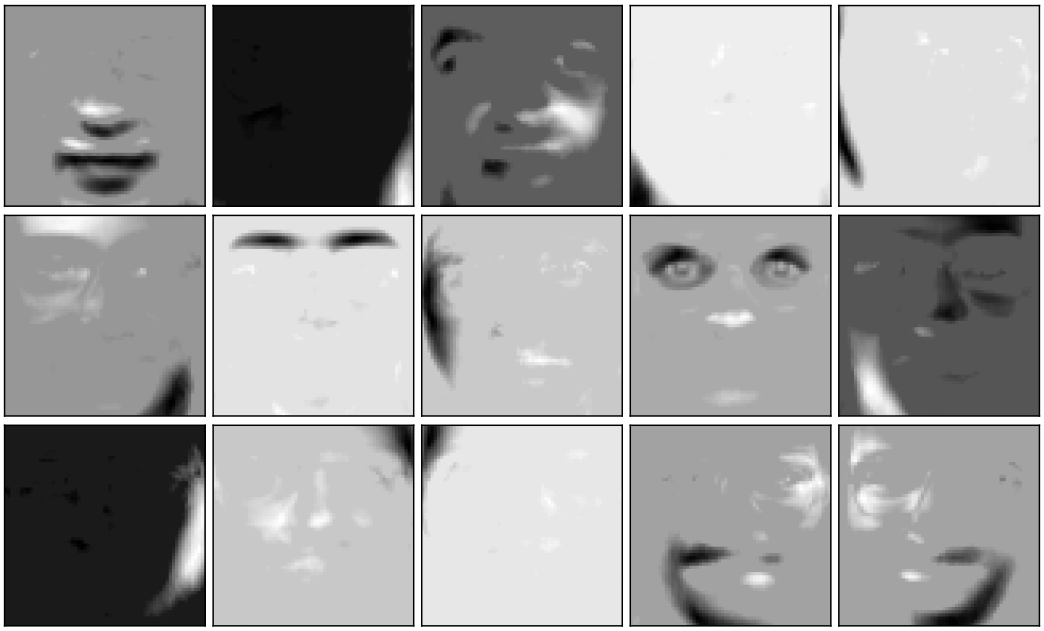


Figure 13: Atoms obtained by applying sparse PCA to the faces dataset from Figure 2.

5 Sparse principal-component analysis

In certain cases, it may be desirable to learn sparse atoms that are able to represent a set of signals. In the case of the faces dataset, this may force the representation to isolate specific face features such as the mouth, the eyes, etc. In order to fit such a model, we can incorporate a sparsity constraint on the atoms by using the ℓ_1 norm

$$\text{minimize} \quad \left|\left|X - \tilde{\Phi}\ \tilde{A}\right|\right|_2^2 + \lambda \sum_{i=1}^{k} \left|\left|\tilde{\Phi}_i\right|\right|_1 \tag{51}$$

$$\text{subject to} \quad \left|\left|\tilde{\Phi}_i\right|\right|_2 = 1, \qquad 1 \leq i \leq k. \tag{52}$$

Due to the sparsity-inducing constraint, this problem is computationally hard, as in the case of nonnegative matrix factorization. We refer the interested reader to [6] for algorithms to compute local minima. Figure 13 shows the atoms obtained by applying this method to the faces dataset from Figure 2. The model indeed learns very localized atoms that represent face features.

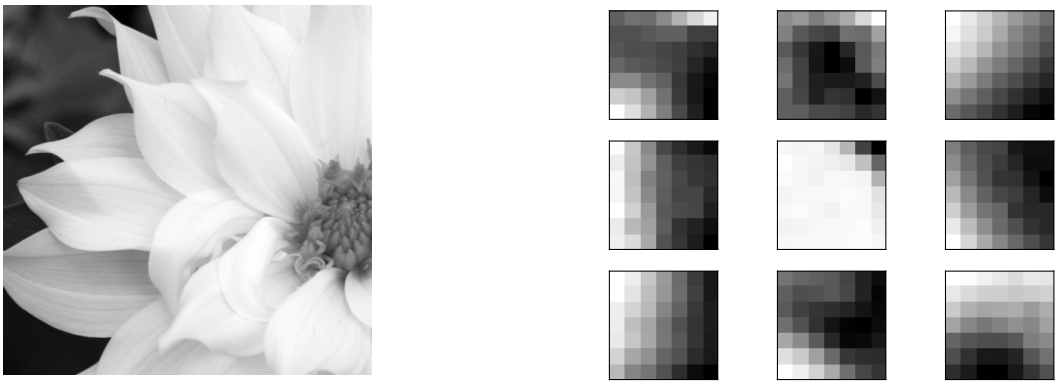


Figure 14: Atoms (right) learnt from patches extracted from a natural image (left).

6 Sparse coding

In Lecture Notes 4 we studied several sparsifying transforms that allow to decompose data into a small number of atoms in order to perform compression or denoising. Dictionary-learning or sparse-coding techniques allow to learn these transforms directly from the data. This is very useful in situations where a dataset with a large number of signals cannot be compactly represented in any predefined dictionary. The aim is to learn a dictionary $\Phi \in \mathbb{R}^{d \times k}$ such that $X \approx \Phi A$, where the matrix of coefficients $A \in \mathbb{R}^{k \times n}$ is very sparse. Following the heuristic that penalizing the ℓ_1 norm promotes sparse solutions, this may be achieved by solving the following optimization program,

$$\min_{\widetilde{\Phi},\, \widetilde{A}} \left|\left| X - \widetilde{\Phi}\widetilde{A} \right|\right|_{\mathrm{F}}^2 + \lambda \left|\left| \widetilde{A} \right|\right|_1 \quad \text{such that } \left|\left| \widetilde{\Phi}_i \right|\right|_2 = 1, \quad 1 \leq i \leq k. \tag{53}$$

Note that the formulation is very similar to sparse PCA, with the crucial difference that we are promoting sparsity in the coefficient matrix, as opposed to in the atoms. An efficient method to compute a local minimum of the nonconvex problem is to apply techniques based on stochastic gradient descent [4].

Figure 14 shows patches learnt from a natural image. The corresponding dictionary can be used to denoise other images quite effectively, as shown in Figure 15. See [3] for other applications of dictionary learning in image processing. Finally, we note that interestingly sparse coding was first proposed in neuroscience, as an explanation of the Gabor-like receptive fields of neurons in the visual cortex [5].

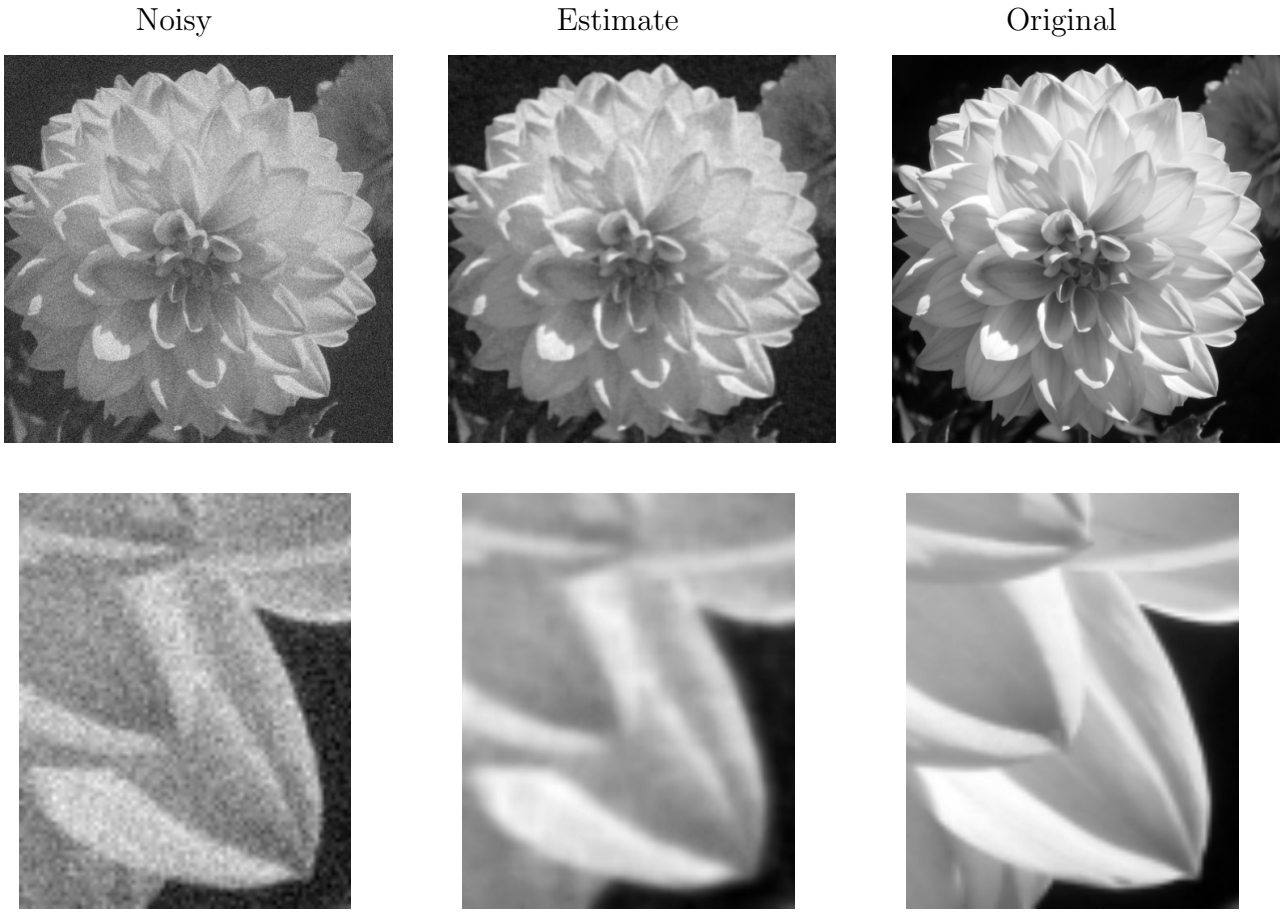


Figure 15: Denoising results using the dictionary learnt from the image shown in Figure 14.

References

The tutorial [3] is an excellent reference on the application of matrix-decomposition techniques in machine learning and image processing. Chapters 7 and 8 of [1] describe low-rank models in statistics. The numerical experiments shown in these notes were implemented using scikit-learn, which is available online at `http://scikit-learn.org`. In particular, a script to apply different matrix-decomposition techniques to the faces dataset is available here.

[1] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC Press, 2015.

[2] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[3] J. Mairal, F. Bach, and J. Ponce. Sparse modeling for image and vision processing. *arXiv preprint arXiv:1411.3230*, 2014.

[4] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60, 2010.

[5] B. A. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[6] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.

A Proofs

A.1 Proof of Lemma 2.1

The function

$$f(\phi) = \frac{1}{2}\sum_{i=1}^{n} ||x_i - \phi||_2^2 \tag{54}$$

is convex and its gradient is equal to

$$\nabla f(\phi) = \sum_{i=1}^{n} \phi - x_i \tag{55}$$

$$= n\phi - \sum_{i=1}^{n} x_i. \tag{56}$$

Setting the gradient to zero yields the result.

A.2 Proof of Proposition 3.5

Let M be an arbitrary rank k matrix with singular value decomposition $U_M \Sigma_M V_M^T$. By Pythagoras' Theorem

$$||X - M||_{\mathrm{F}}^2 = \left|\left|X - U_M U_M^T X\right|\right|_{\mathrm{F}}^2 + \left|\left|M - U_M U_M^T X\right|\right|_{\mathrm{F}}^2$$

because the column space of $X - U_M U_M^T X$ is orthogonal to the column space of M. Now, recall that by Theorem 3.2,

$$\left|\left|U_{1:k} U_{1:k}^T X\right|\right|_{\mathrm{F}}^2 \leq \left|\left|U_M U_M^T X\right|\right|_{\mathrm{F}}^2$$

since $U_M U_M^T$ represents a projection onto a k-dimensional subspace. We conclude that

$$\begin{aligned} ||X - M||_{\mathrm{F}}^2 &\geq \left|\left|X - U_M U_M^T X\right|\right|_{\mathrm{F}}^2 \\ &\geq \left|\left|U_{1:k} U_{1:k}^T X\right|\right|_{\mathrm{F}}^2 . \end{aligned}$$

Low-rank models

1 Matrix completion

1.1 The matrix-completion problem

The Netflix Prize was a contest organized by Netflix from 2007 to 2009 in which teams of data scientists tried to develop algorithms to improve the prediction of movie ratings. The problem of predicting ratings can be recast as that of completing a matrix from some of its entries, as illustrated in Figure 1. This problem is known as *matrix completion*.

At first glance, the problem of completing a matrix such as this one

$$\begin{bmatrix} 1 & ? & 5 \\ ? & 3 & 2 \end{bmatrix} \tag{1}$$

may seem completely ill posed. We can just fill in the missing entries arbitrarily! In more mathematical (and perhaps slightly pedantic) terms, the completion problem is equivalent to an underdetermined system of equations

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} M_{11} \\ M_{21} \\ M_{12} \\ M_{22} \\ M_{13} \\ M_{23} \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 5 \\ 2 \end{bmatrix}. \tag{2}$$

In order to solve the problem, we need to make an assumption on the structure of the matrix that we aim to complete. Recall that in compressed sensing we made the assumption that the original signal was sparse. Even though the recovery problem in compressed sensing is also underdetermined, recovery is possible as long as the number of measurements is proportional to the number of nonzero entries (up to logarithmic factors). In the case of matrix completion, we will make the assumption that the original matrix is low rank. This implies that there exists a high correlation between the entries of the matrix, which may make it possible to infer the missing entries from the observations. As a very simple example,

Figure 1: A depiction of the Netflix challenge in matrix form. Each row corresponds to a user that ranks a subset of the movies, which correspond to the columns. The figure is due to Mahdi Soltanolkotabi.

consider the following matrix

$$\begin{bmatrix} 1 & 1 & 1 & 1 & ? & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ ? & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{3}$$

Setting the missing entries to 1 yields a rank 1 matrix, whereas setting them to any other number yields a rank 2 or rank 3 matrix.

The low-rank assumption implies that if the matrix has dimensions $m \times n$ then it can be factorized into two matrices that have dimensions $m \times r$ and $r \times n$. This factorization allows to encode the matrix using $r\,(m+n)$ parameters. If the number of observed entries is larger than $r\,(m+n)$ parameters then it may be possible to recover the missing entries. However, this is not enough to ensure that the problem is well posed, as we will see in the following section.

1.2 When does matrix completion make sense?

The results of matrix completion will obviously depend on the subset of entries that are observed. For example, completion is impossible unless we observe at least one entry in each

row and column. To see why let us consider a rank 1 matrix for which we don't observe the second row,

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ ? & ? & ? & ? \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ ? \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}. \tag{4}$$

As long as we set the missing row to equal the same value, we will obtain a rank-1 matrix consistent with the measurements. In this case, the problem is not well posed.

In general, we need samples that are distributed across the whole matrix. This may be achieved by sampling entries uniformly at random. Although this model does not completely describe matrix completion problems in practice (some users tend to rate more movies, some movies are very popular and are rated by many people), making the assumption that the revealed entries are random simplifies theoretical analysis and avoids dealing with adversarial cases designed to make deterministic patterns fail.

We now turn to the question of what matrices can be completed from a subset of entries samples uniformly at random. Intuitively, matrix completion can be achieved when the information contained in the entries of the matrix is *spread out* across multiple entries. If the information is very localized then it will be impossible to reconstruct the missing entries. Consider a simple example where the matrix is sparse

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 23 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{5}$$

If we don't observe the nonzero entry, we will naturally assume that it was equal to zero.

The problem is not restricted to sparse matrices. In the following matrix the last row does not seem to be correlated to the rest of the rows,

$$M := \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ -3 & 3 & -3 & 3 \end{bmatrix}. \tag{6}$$

This is revealed by the singular-value decomposition of the matrix, which allows to decompose

it into two rank-1 matrices.

$$M = U\,\Sigma V^T \tag{7}$$

$$= \begin{bmatrix} 0.5 & 0 \\ 0.5 & 0 \\ 0.5 & 0 \\ 0.5 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 8 & 0 \\ 0 & 6 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ -0.5 & 0.5 & -0.5 & 0.5 \end{bmatrix} \tag{8}$$

$$= 8 \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix} + 6 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} -0.5 & 0.5 & -0.5 & 0.5 \end{bmatrix} \tag{9}$$

$$= \sigma_1 U_1 V_1^T + \sigma_2 U_2 V_2^T. \tag{10}$$

The first rank-1 component of this decomposition has information that is very spread out,

$$\sigma_1 U_1 V_1^T = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{11}$$

The reason is that most of the entries of V_1 are nonzero and have the same magnitude, so that each entry of U_1 affects every single entry of the corresponding row. If one of those entries is missing, we can still recover the information from the other entries.

In contrast, the information in the second rank-1 component is very localized, due to the fact that the corresponding left singular vector is very sparse,

$$\sigma_2 U_2 V_2^T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -3 & 3 & -3 & 3 \end{bmatrix}. \tag{12}$$

Each entry of the right singular vector only affects one entry of the component. If we don't observe that entry then it will be impossible to recover.

This simple example shows that sparse singular vectors are problematic for matrix completion. In order to quantify to what extent the information is spread out across the low-rank matrix we define a coherence measure that depends on the singular vectors.

Definition 1.1 (Coherence). *Let $U \Sigma V^T$ be the singular-value decomposition of an $n \times n$ matrix M with rank r. The coherence μ of M is a constant such that*

$$\max_{1 \leq j \leq n} \sum_{i=1}^{r} U_{ij}^2 \leq \frac{n\mu}{r} \tag{13}$$

$$\max_{1 \leq j \leq n} \sum_{i=1}^{r} V_{ij}^2 \leq \frac{n\mu}{r}. \tag{14}$$

This condition was first introduced in [3]. Its exact formulation is not too important. The point is that matrix completion from uniform samples only makes sense for matrices which are incoherent, and therefore do not have spiky singular values. There is a direct analogy with the super-resolution problem, where sparsity is not a strong enough constraint to make the problem well posed and the class of signals of interest has to be further restricted to signals with supports that satisfy a minimum separation.

1.3 The nuclear norm

In compressed sensing and super-resolution we penalize the ℓ_1 norm of the recovered signal to promote sparse estimates. Similarly, for matrix completion we penalize a certain matrix norm to promote low-rank structure.

First, let us introduce the trace operator and an inner product for matrices.

Definition 1.2 (Trace). *The trace of an $n \times n$ matrix is defined as*

$$\text{trace}\,(M) := \sum_{i=1}^{n} M_{ii}. \tag{15}$$

Definition 1.3 (Matrix inner product). *The inner product between two $m \times n$ matrices A and B is*

$$\langle A, B \rangle := \text{trace}\left(A^T B\right) \tag{16}$$

$$= \sum_{i=1}^{m} \sum_{i=1}^{n} A_{ij} B_{ij}. \tag{17}$$

Note that this inner product is equivalent to the inner product of the matrices if we vectorize them.

We now define three matrix norms.

Definition 1.4 (Matrix norms)**.** *Let $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$ be the singular values of $M \in \mathbb{R}^{m \times n}$, $m \geq n$. The operator norm is equal to the maximum singular value*

$$||M|| := \max_{||u||_2 \leq 1} ||M\,u||_2 \tag{18}$$

$$= \sigma_1. \tag{19}$$

The Frobenius norm is the norm induced by the inner product from Definition 1.3. It is equal to the ℓ_2 norm of the vectorized matrix, or equivalently to the ℓ_2 norm of the singular values

$$||M||_{\mathrm{F}} := \sqrt{\sum_i M_{ij}^2} \tag{20}$$

$$= \sqrt{\mathrm{trace}\,(M^T M)} \tag{21}$$

$$= \sqrt{\sum_{i=1}^{n} \sigma_i^2}. \tag{22}$$

The nuclear norm is equal to the ℓ_1 norm of the singular values

$$||M||_* := \sum_{i=1}^{n} \sigma_i. \tag{23}$$

The following proposition, proved in Section A.1 of the appendix, is analogous to Hölder's inequality for vectors.

Proposition 1.5. *For any matrix $A \in \mathbb{R}^{m \times n}$,*

$$||A||_* = \sup_{||B|| \leq 1} \langle A, B \rangle\,. \tag{24}$$

A direct consequence of the proposition is that the nuclear norm satisfies the triangle inequality. This implies that it is a norm (it clearly satisfies the other properties of a norm) and hence a convex function.

Corollary 1.6. *For any $m \times n$ matrices A and B*

$$||A + B||_* \leq ||A||_* + ||B||_*\,. \tag{25}$$

Proof.

$$||A + B||_* = \sup_{\{C \mid ||C|| \leq 1\}} \langle A + B, C \rangle \tag{26}$$

$$= \sup_{||C|| \leq 1} \langle A, C \rangle + \sup_{||D|| \leq 1} \langle B, D \rangle \tag{27}$$

$$= ||A||_* + ||B||_*\,. \tag{28}$$

□

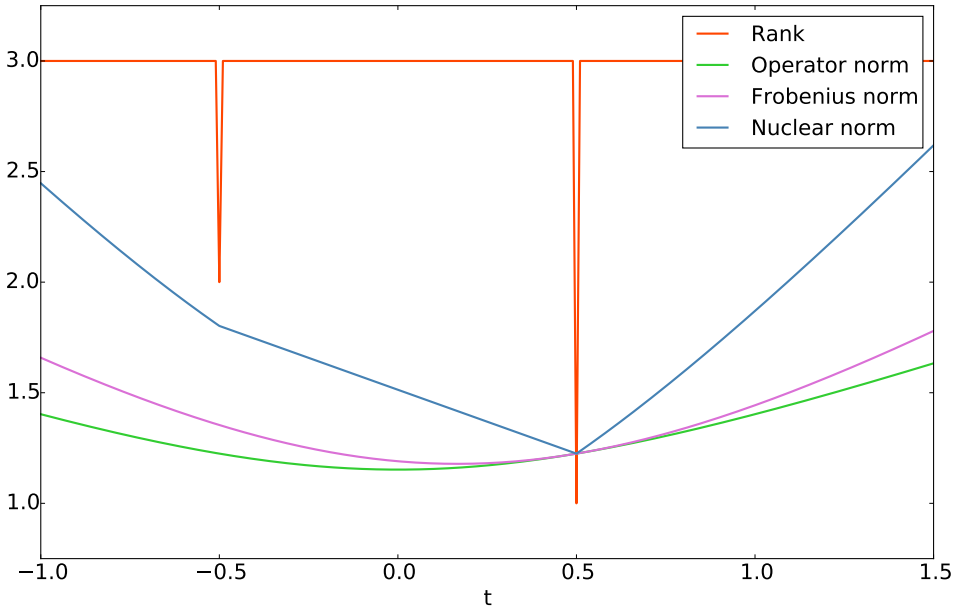


Figure 2: Values of different norms for the matrix $M(t)$ defined by (29). The rank of the matrix for each t is marked in orange.

Penalizing the nuclear norm induces low-rank structure, just like penalizing the ℓ_1 norm of a vector induces sparse structure. A justification for this is that the nuclear norm is the ℓ_1 norm of the singular values and therefore minimizing it tends to set most of them to zero. In order to provide a more concrete example, let us consider the following matrix

$$M(t) := \begin{bmatrix} 0.5+t & 1 & 1 \\ 0.5 & 0.5 & t \\ 0.5 & 1-t & 0.5 \end{bmatrix}, \tag{29}$$

which is parametrized by the parameter t. In Figure 2 we compare the rank, the operator norm, the Frobenius norm and the nuclear norm of $M(t)$ for different values of t. The value of t that minimizes the rank is the same as the one that minimizes the nuclear norm. In contrast, the values of t that minimize the operator and Frobenius norms are different. This justifies using the nuclear norm as a proxy for the rank.

As we discussed in the previous sections, we are interested in recovering low-rank matrices from a subset of their entries. Let y be a vector containing the revealed entries and let Ω be the corresponding entries. Ideally, we would like to select the matrix with the lowest rank that corresponds to the measurements,

$$\min_{\widetilde{X} \in \mathbb{R}^{m \times n}} \operatorname{rank}\left(\widetilde{X}\right) \quad \text{such that } \widetilde{X}_\Omega = y. \tag{30}$$

Unfortunately, this optimization problem is computationally hard to solve. Substituting the rank with the nuclear norm yields a tractable alternative:

$$\min_{\widetilde{X}\in\mathbb{R}^{m\times n}} \left|\left|\widetilde{X}\right|\right|_* \quad \text{such that } \widetilde{X}_\Omega = y. \tag{31}$$

The cost function is convex and the constraint is linear, so this is a convex program. In practice, the revealed entries are usually noisy. They do not correspond exactly to entries from a low-rank matrix. We take this into account by removing the equality constraint and adding a data-fidelity term penalizing the ℓ_2-norm error over the revealed entries in the cost function,

$$\min_{\widetilde{X}\in\mathbb{R}^{m\times n}} \frac{1}{2}\left|\left|\widetilde{X}_\Omega - y\right|\right|_2^2 + \lambda\left|\left|\widetilde{X}\right|\right|_*, \tag{32}$$

where $\lambda > 0$ is a regularization parameter.

We now apply this method to the following completion problem:

$$\begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} & \\ \left(\begin{array}{cccc} 1 & ? & 5 & 4 \\ ? & 1 & 4 & 5 \\ 4 & 5 & 2 & ? \\ 5 & 4 & 2 & 1 \\ 4 & 5 & 1 & 2 \\ 1 & 2 & ? & 5 \end{array}\right) & \begin{array}{l} \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{array} \end{array} \tag{33}$$

In more detail we apply the following steps:

1. We compute the average observed rating and subtract it from each entry in the matrix. We denote the vector of centered ratings by y.
2. We solve the optimization problem (31).
3. We add the average observed rating to the solution of the optimization problem and round each entry to the nearest integer.

The result is pretty good,

$$\begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} & \\ \left(\begin{array}{cccc} 1 & 2\ (1) & 5 & 4 \\ 2\ (2) & 1 & 4 & 5 \\ 4 & 5 & 2 & 2\ (1) \\ 5 & 4 & 2 & 1 \\ 4 & 5 & 1 & 2 \\ 1 & 2 & 5\ (5) & 5 \end{array}\right) & \begin{array}{l} \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{array} \end{array} \tag{34}$$

For comparison the original ratings are shown in brackets.

1.4 Theoretical guarantees

In this section we will explain how to establish theoretical guarantees for matrix completion via nuclear norm minimization. The following theorem is proved in [6].

Theorem 1.7 (Matrix completion)**.** *If the matrix M has rank r and coherence μ, the solution to the optimization problem*

$$\min_{\widetilde{X}\in\mathbb{R}^{m\times n}} \left|\left|\widetilde{X}\right|\right|_* \quad \text{such that } \widetilde{X}_\Omega = y \tag{35}$$

achieves exact recovery with high probability as long as the number of samples is proportional to $\mu r\,(n+m)$ (up to logarithmic terms).

For matrices that are incoherent, the coherence parameter is equal to a constant, so that nuclear norm allows to reconstruct the matrix from a minimal number of measurements (recall that a low-rank matrix depends on $r\,(n+m)$ parameters), up to logarithmic factors[1].

In order to prove that a certain matrix is the solution to the optimization problem we use the same technique that we used to analyze compressed sensing and super-resolution: we build a dual certificate. The dual certificate is a subgradient with a certain structure, so we first define the subgradients of the nuclear norm.

Proposition 1.8 (Subgradients of the nuclear norm)**.** *Let $M = U\Sigma V^T$ be the singular-value decomposition of M. Any matrix of the form*

$$G := UV^T + W \qquad ||W|| \leq 1, \tag{36}$$

$$U^T W = 0, \tag{37}$$

$$W\,V = 0 \tag{38}$$

is a subgradient of the nuclear norm at M.

Proof. Note that by definition $||G|| \leq 1$. For any matrix H

$$||M+H||_* \geq \langle G, M+H\rangle \quad \text{by Proposition 1.5} \tag{39}$$

$$= \left\langle UV^T, M\right\rangle + \langle G, H\rangle \tag{40}$$

$$= ||M||_* + \langle G, H\rangle \quad \text{as in (137).} \tag{41}$$

□

The following proposition provides a dual certificate for the nuclear-norm minimization problem.

[1] In fact, one can show that a multiplicative logarithmic factor is necessary in order to make sure that all rows and columns are sampled. See [4].

Proposition 1.9 (Dual certificate for nuclear-norm minimization)**.** *Let $M = U\Sigma V^T$ be the singular-value decomposition of M. A matrix Q supported on Ω, i.e. such that*

$$Q_{\Omega^c} = 0, \tag{42}$$

is a dual certificate of the optimization problem

$$\min_{\widetilde{X}\in\mathbb{R}^{m\times n}} \left|\left|\widetilde{X}\right|\right|_* \quad \text{such that } \widetilde{X}_\Omega = y \tag{43}$$

as long as

$$Q = UV^T + W, \qquad ||W|| < 1, \tag{44}$$

$$U^T W = 0, \tag{45}$$

$$W\,V = 0. \tag{46}$$

Proof. By Proposition 1.8 Q is a subgradient of the nuclear norm at M. Any matrix that is feasible for the optimization problem can be expressed as $M + H$ where $H_\Omega = 0$ because the revealed entries must be equal to M_Ω. This immediately implies that $\langle Q, H\rangle = 0$. We conclude that

$$||M + H||_* \geq ||M||_* + \langle Q, H\rangle \tag{47}$$

$$= ||M||_* . \tag{48}$$

This proves that M is a solution. A variation of this argument that uses the strict inequality in (44) establishes that if Q exists then M is the unique solution [3]. □

In order to show that matrix completion via nuclear-norm minimization succeeds, we need to show that such a dual certificate exists with high probability. For this we will need the matrix to be incoherent, since otherwise UV^T may have large entries which are not in Ω. This would make it very challenging to construct Q in a way that $UV^T = Q - W$ for a matrix W with bounded operator norm. The first guarantees for matrix completion were obtained by constructing such a certificate in [3] and [4]. Subsequently, the results were improved in [6], where it is shown that an *approximate* dual certificate also allows to establish exact recovery, and simplifies the proofs significantly.

1.5 Algorithms

In this section we describe a proximal-gradient method to solve Problem 32. Recall that proximal-gradient methods allow to solve problems of the form

$$\text{minimize} \quad f(x) + g(x), \tag{49}$$

where f is differentiable and we can apply the proximal operator prox_g efficiently.

Recall that the proximal operator norm of the ℓ_1 norm is a soft-thresholding operator. Analogously, the proximal operator of the nuclear norm is applied by soft-thresholding the singular values of the matrix. The result is proved in Section 1.10 of the appendix.

Proposition 1.10 (Proximal operator of the nuclear norm)**.** *The solution to*

$$\min_{\widetilde{X}\in\mathbb{R}^{m\times n}} \frac{1}{2}\left|\left|Y-\widetilde{X}\right|\right|_{\text{F}}^2 + \tau\left|\left|\widetilde{X}\right|\right|_* \tag{50}$$

is $\mathcal{D}_\tau(Y)$, obtained by soft-thresholding the singular values of $Y = U\,\Sigma V^T$

$$\mathcal{D}_\tau(Y) := U\,\mathcal{S}_\tau(\Sigma)\,V^T, \tag{51}$$

$$\mathcal{S}_\tau(\Sigma)_{ii} := \begin{cases} \Sigma_{ii}-\tau & \text{if } \Sigma_{ii} > \tau, \\ 0 & \text{otherwise.} \end{cases} \tag{52}$$

Algorithm 1.11 (Proximal-gradient method for nuclear-norm regularization)**.** *Let Y be a matrix such that $Y_\Omega = y$ and let us abuse notation by interpreting $X_\Omega^{(k)}$ as a matrix which is zero on Ω^c. We set the initial point $X^{(0)}$ to Y. Then we iterate the update*

$$X^{(k+1)} = \mathcal{D}_{\alpha_k\lambda}\left(X^{(k)} - \alpha_k\left(X_\Omega^{(k)} - Y\right)\right), \tag{53}$$

where $\alpha_k > 0$ is the step size.

1.6 Alternating minimization

Minimizing the nuclear norm to recover a low-rank matrix is an effective method but it has a drawback: it requires repeatedly computing the singular-value decomposition of the matrix, which can be computationally heavy for large matrices. A more efficient alternative is to parametrize the matrix as AB where $A \in \mathbb{R}^{m\times k}$ and $B \in \mathbb{R}^{k\times n}$, which requires fixing a value for the rank of the matrix k (in practice this can be set by cross validation). The two components A and B can then be fit by solving the optimization problem

$$\min_{\widetilde{A}\in\mathbb{R}^{m\times k},\widetilde{B}\in\mathbb{R}^{k\times n}} \left|\left|\left(\widetilde{A}\widetilde{B}\right)_\Omega - y\right|\right|_2. \tag{54}$$

This nonconvex problem is usually tackled by alternating minimization. Indeed, if we fix $\widetilde{B} = B$ the optimization problem over $\widetilde{A}$ is just a least-squares problem

$$\min_{\widetilde{A}\in\mathbb{R}^{m\times k}} \left|\left|\left(\widetilde{A}B\right)_\Omega - y\right|\right|_2 \tag{55}$$

and the same is true for the optimization problem over $\widetilde{B}$ if we fix $\widetilde{A} = A$. Iteratively solving these least-squares problems allows to find a local minimum of the cost function. Under certain assumptions, one can even show that a certain initialization coupled with this procedure guaranteees exact recovery, see [8] for more details.

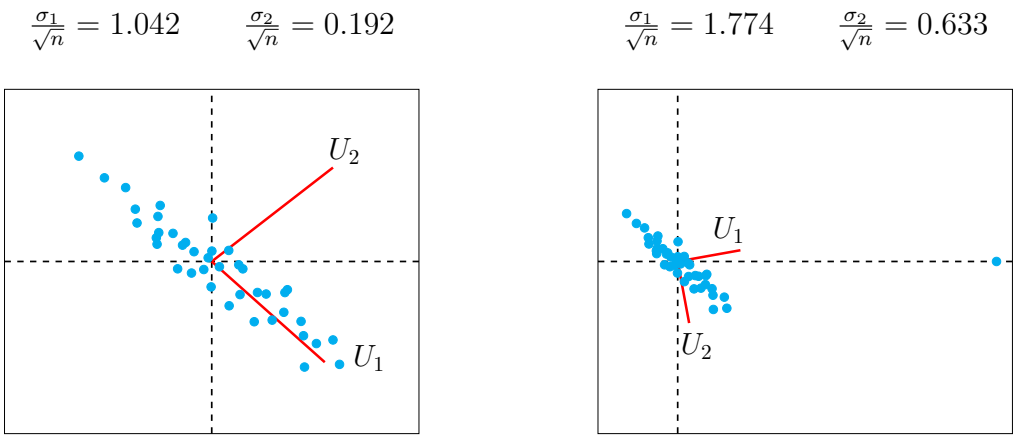


Figure 3: Principal components of a dataset before and after the inclusion of an outlier.

2 Low rank + sparse model

2.1 The effect of outliers on principal-component analysis

It is well known that outliers may severely distort the results of applying principal-component analysis to a dataset. Figure 3 shows the dramatic effect that just one outlier can have on the principal components. Equivalently, if several entries of a matrix are entirely uncorrelated with the rest, this may disrupt any low-rank structure that might be present. To illustrate this consider a rank-1 matrix of movie ratings

$$\begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} & \\ \left(\begin{array}{cccc} 1 & 1 & 5 & 5 \\ 1 & 1 & 5 & 5 \\ 5 & 5 & 1 & 1 \\ 5 & 5 & 1 & 1 \\ 5 & 5 & 1 & 1 \\ 1 & 1 & 5 & 5 \end{array}\right) & \begin{array}{l} \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{array} \end{array} \tag{56}$$

Now imagine that Bob randomly assigns a 5 instead of a 1 to The Dark Knight by mistake and that Larry hates Superman 2 because one of the actresses reminds him of his ex,

$$\begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} \\ \left(\begin{array}{cccc} 5 & 1 & 5 & 5 \\ 1 & 1 & 5 & 5 \\ 5 & 5 & 1 & 1 \\ 5 & 5 & 1 & 1 \\ 5 & 5 & 1 & 1 \\ 1 & 1 & 5 & 1 \end{array}\right) & \begin{array}{l} \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{array} \end{array} \tag{57}$$

Now let us compare the singular-value decomposition after subtracting the mean rating with

$$U\Sigma V^T = U \begin{bmatrix} 8.543 & 0 & 0 & 0 \\ 0 & 4.000 & 0 & 0 \\ 0 & 0 & 2.649 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} V^T \tag{58}$$

and without outliers

$$A - \bar{A} = U\Sigma V^T = U \begin{bmatrix} 9.798 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} V^T. \tag{59}$$

The matrix is now rank 3 instead of rank 1. In addition, the first left singular vector,

$$\begin{array}{cccccccc} & \text{D. Knight} & & \text{Sp. 3} & \text{Love Act.} & \text{B.J.'s Diary} & \text{P. Woman} & \text{Sup. 2} \\ U_1 = & (-0.2610 & & -0.4647 & 0.4647 & 0.4647 & 0.4647 & -0.2610) \end{array} \tag{60}$$

does not allow to cluster the movies as effectively as when there are no outliers

$$\begin{array}{cccccccc} & \text{D. Knight} & & \text{Sp. 3} & \text{Love Act.} & \text{B.J.'s Diary} & \text{P. Woman} & \text{Sup. 2} \\ U_1 = & (-0.4082 & & -0.4082 & 0.4082 & 0.4082 & 0.4082 & -0.4082)\,. \end{array} \tag{61}$$

2.2 Low rank + sparse model

In order to tackle situations where a low-rank matrix may be corrupted with outliers, we define a low-rank + sparse model where the matrix is modeled as the sum of a low-rank component L and a sparse component S. Figure 4 shows an example with simulated data. If we are able to separate the two components from the data, we can apply PCA to L in order

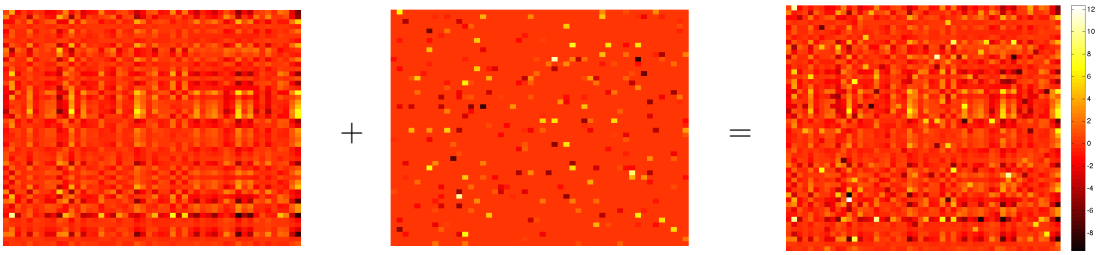


Figure 4: M is obtained by summing a low-rank matrix L and a sparse matrix S.

to retrieve the low-rank structure that is distorted by the presence of S. The problem of separating both components is very related to matrix completion: if we knew the location of the outliers then we could just apply a matrix-completion method to the remaining entries in order to recover the low-rank component.

We now consider under what assumptions the decomposition of a low rank and a sparse component is unique. Clearly the low-rank component cannot be sparse, otherwise there can be many possible decompositions

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 23 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 47 & 0 \end{bmatrix} \tag{62}$$

$$= \tag{63}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 23 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 47 & 0 \end{bmatrix}. \tag{64}$$

We can avoid this by considering low-rank components with low coherence.

Similarly, the sparse component cannot be low rank. This occurs when its support is highly concentrated on a small number of columns or rows,

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{65}$$

$$= \tag{66}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{67}$$

A simple assumption that precludes the support of the sparse component from being too concentrated is that it is distributed uniformly at random. As in the case of the revealed entries in matrix completion, this assumption doesn't usually hold in practice but it often provides a good approximation.

2.3 Robust principal-component analysis

Following the ideas that we discussed in previous lecture notes and in Section 1.3, we compute the decomposition by penalizing the nuclear norm of L, which induces low-rank structure, and the ℓ_1 norm of S, which induces sparse structure.

$$\min_{\widetilde{L},\widetilde{S}\in\mathbb{R}^{m\times n}} \left|\left|\widetilde{L}\right|\right|_* + \lambda \left|\left|\widetilde{S}\right|\right|_1 \quad \text{such that } \widetilde{L}+\widetilde{S}=M, \tag{68}$$

where M is the matrix of data. To be clear, $||\cdot||_1$ is the ℓ_1 norm of the vectorized matrix and $\lambda > 0$ is a regularization parameter. This algorithm introduced by [2,5] is often called *robust PCA*, since it aims to obtain a low-rank component that is not affected by the presence of outliers.

The regularization parameter λ determines the weight of the two structure-inducing norms. It may be chosen by cross validation, but its possible values may be restricted by considering some simple examples. If the matrix that we are decomposing is equal to a rank-1 matrix, then the sparse component should be set to zero. Since

$$\left|\left|\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}\right|\right|_* = n, \qquad \left|\left|\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}\right|\right|_1 = n^2, \tag{69}$$

we must have $\lambda > \frac{1}{n}$ for this to be the case. In contrast, if the matrix contains just one nonzero entry then the low-rank component should be set to zero. Since

$$\left|\left|\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\right|\right|_* = 1, \qquad \left|\left|\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\right|\right|_1 = 1 \tag{70}$$

this will be the case if $\lambda < 1$. Finally, if the matrix just has a row of ones it is unclear whether we should assign it to the low-rank or the sparse component, so the value of the two terms in the cost function should have a similar value. Since

$$\left|\left|\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\right|\right|_* = \sqrt{n}, \qquad \left|\left|\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\right|\right|_1 = n \tag{71}$$

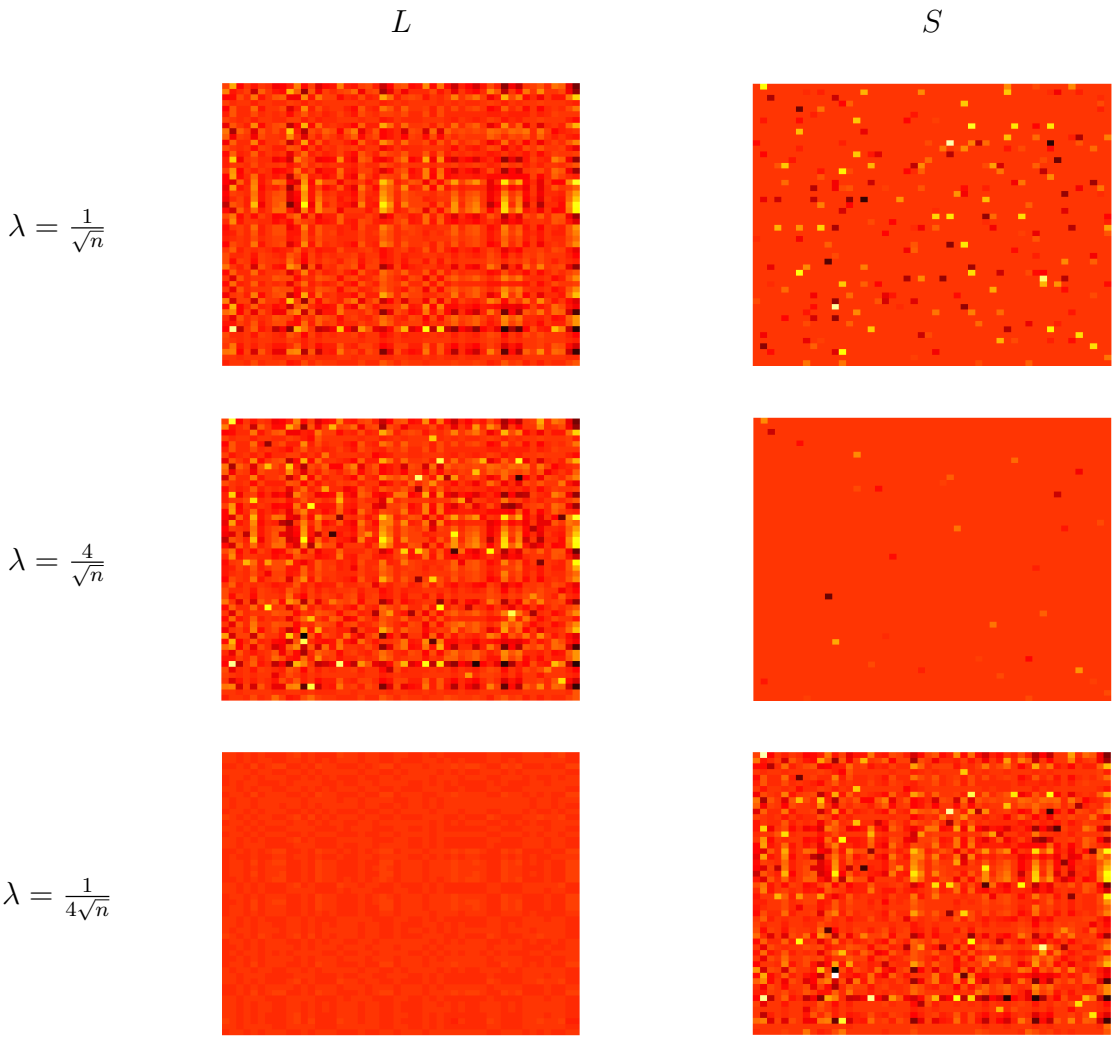


Figure 5: Results of solving Problem (68) for different values of the regularization parameter.

this suggests setting $\lambda \approx \frac{1}{\sqrt{n}}$.

Figure 5 shows the results of solving Problem (68) for different values of λ. If λ is too small, then it is *cheap* to increase the content of the sparse component, which won't be very sparse as a result. Similarly, if λ is too large, then the low-rank component won't be low-rank, as the nuclear-norm term has less influence. Setting λ correctly allows to achieve a perfect decomposition.

2.4 Theoretical guarantees

The following theorem from [2] provides theoretical guarantees for the decomposition of low-rank and sparse matrices using convex optimization. We omit some details in the statement of the result, which can be found in [2].

Theorem 2.1 (Exact decomposition via convex programming)**.** *Let $M = L + S$ be a $n \times n$ matrix where L is rank r and has coherence μ and the support of S is distributed uniformly at random. Problem* (68) *with $\lambda = 1/\sqrt{n}$ recovers L and S exactly as long as the rank of L is of order n/μ up to logarithmic factors and the sparsity level of S is bounded by a certain constant times n^2.*

As long as the low-rank component is incoherent and the support of the sparse component is uniformly distributed, convex programming allows to obtain a perfect decomposition up to values of the rank of L and the sparsity level of S that are essentially optimal.

The proof of this result is based on the construction of an approximate dual certificate. This certificate approximates the certificate described in the following proposition.

Proposition 2.2 (Dual certificate for robust PCA)**.** *Let $M = L + S$, where $L = U\Sigma V^T$ is the singular-value decomposition of L and Ω is the support of S. A matrix Q of the form*

$$Q = UV^T + W = \lambda \operatorname{sign}(S) + F \tag{72}$$

where

$$||W|| < 1, \qquad U^T W = 0, \qquad W V = 0, \tag{73}$$

$$F_\Omega = 0, \qquad ||F||_\infty < \lambda, \tag{74}$$

is a dual certificate of the optimization problem

$$\min_{\widetilde{L},\widetilde{S}\in\mathbb{R}^{m\times n}} \left|\left|\widetilde{L}\right|\right|_* + \lambda \left|\left|\widetilde{S}\right|\right|_1 \quad \text{such that } \widetilde{L} + \widetilde{S} = M. \tag{75}$$

Proof. Let us consider a feasible pair $L+L'$ and $S+S'$. Since $L+S = M$, $L+L'+S+S' = M$ implies $L' + S' = 0$. The conditions on Q imply that Q is a subgradient of the nuclear norm at L and that $\frac{1}{\lambda}Q$ is a subgradient of the ℓ_1 norm at S, which implies

$$||L + L'||_* + \lambda ||S + S'||_1 \geq ||L||_* + \langle Q, L' \rangle + \lambda ||S||_1 + \lambda \left\langle \frac{1}{\lambda} Q, S' \right\rangle \tag{76}$$

$$= ||L||_* + \lambda ||S||_1 + \langle Q, L' + S' \rangle \tag{77}$$

$$= ||L||_* + \lambda ||S||_1 . \tag{78}$$

Modifying the argument slightly to use the strict inequalities in (73) and (74) allows to prove that L and S are the unique solution. □

2.5 Algorithms

In order to derive an algorithm to solve Problem (68), we will begin by considering a canonical problem with linear equality constraints

$$\begin{aligned} &\text{minimize} \quad f(x) && (79)\\ &\text{subject to} \quad Ax = y. && (80) \end{aligned}$$

The Lagrangian of this problem is

$$\mathcal{L}(x, z) := f(x) + \langle z, Ax - y \rangle, \tag{81}$$

where z is a Lagrange multiplier or dual variable. The dual function is obtained by minimizing the Lagrangian over the primal variable x,

$$g(z) := \inf_x f(x) + \langle z, Ax - y \rangle. \tag{82}$$

If strong duality holds, for a primal solution x^* and a dual solution z^* we have

$$\begin{aligned} f(x^*) &= g(z^*) && (83)\\ &= \inf_x \mathcal{L}(x, z^*) && (84)\\ &\leq f(x^*). && (85) \end{aligned}$$

This implies that if we know z^*, we can compute x^* by solving

$$\text{minimize} \quad \mathcal{L}(x, z^*). \tag{86}$$

The dual-ascent method tries to find a solution to the dual problem by using gradient ascent on the dual function. In more detail, to compute the gradient of the dual function at z we first minimize the Lagrangian over the primal variable to obtain the minimizer $\hat{x}$. The gradient is then equal to

$$\nabla g(z) = A\hat{x} - y. \tag{87}$$

Algorithm 2.3 (Dual ascent)**.** *We set an initial value $z^{(0)}$. Then we iterate between updating the primal and the dual variables.*

- *Primal-variable update*

$$x^{(k)} = \arg\min_x \mathcal{L}\left(x, z^{(k)}\right). \tag{88}$$

- *Dual-variable update*

$$z^{(k+1)} = z^{(k)} + \alpha^{(k)} \left(Ax^{(k)} - y\right) \tag{89}$$

for a step size $\alpha^{(k)} \geq 0$.

It turns out that the dual-ascent method is not very stable [1]. This may be tackled by defining an augmented Lagrangian

$$\mathcal{L}_\rho(x,z) := f(x) + \langle z, Ax - y\rangle + \frac{\rho}{2}||Ax-y||_2^2, \tag{90}$$

which corresponds to the Lagrangian of the modified (yet equivalent) problem

$$\begin{aligned} \text{minimize} \quad & f(x) + \frac{\rho}{2}||Ax-y||_2^2 && (91)\\ \text{subject to} \quad & Ax = y, && (92) \end{aligned}$$

where $\rho > 0$ is a parameter. Applying dual ascent using the augmented Lagrangian yields the method of multipliers.

Algorithm 2.4 (Method of multipliers)**.** *We set an initial value $z^{(0)}$. Then we iterate between updating the primal and the dual variables.*

- *Primal-variable update*

$$x^{(k)} = \arg\min_x \mathcal{L}_\rho\left(x, z^{(k)}\right). \tag{93}$$

- *Dual-variable update*

$$z^{(k+1)} = z^{(k)} + \rho\left(Ax^{(k)} - y\right). \tag{94}$$

Note that in the dual-variable update we have set the step size to ρ. This can be justified as follows. Setting $z^{(k+1)}$ so that

$$\nabla_x \mathcal{L}\left(x^{(k)}, z^{(k+1)}\right) = 0 \tag{95}$$

means that the Lagrangian of the original problem is minimized in each iteration. Given that

$$\nabla_x \mathcal{L}_\rho\left(x^{(k)}, z^{(k)}\right) = \nabla_x f\left(x^{(k)}\right) + A^T\left(z^{(k)} + \rho\left(Ax - y\right)\right) \tag{96}$$

and

$$\nabla_x \mathcal{L}_\rho\left(x^{(k)}, z^{(k)}\right) = 0, \tag{97}$$

setting $z^{(k+1)}$ as in (94) yields

$$\begin{aligned} \nabla_x \mathcal{L}\left(x^{(k)}, z^{(k+1)}\right) &= \nabla_x f\left(x^{(k)}\right) + A^T\left(z^{(k)} + \rho\left(Ax - y\right)\right) && (98)\\ &= 0. && (99) \end{aligned}$$

Applying the same ideas to a composite objective function

$$\begin{aligned} \text{minimize} \quad & f_1(x_1) + f_2(x_2) && (100)\\ \text{subject to} \quad & Ax_1 + Bx_2 = y && (101) \end{aligned}$$

yields the alternating direction method of multipliers (ADMM).

Algorithm 2.5 (Alternating direction method of multipliers)**.** *We set an initial value $z^{(0)}$. Then we iterate between updating the primal and the dual variables.*

- *Primal-variable updates*

$$x_1^{(k)} = \arg\min_x \mathcal{L}_\rho \left(x, x_2^{(k-1)}, z^{(k)} \right), \tag{102}$$

$$x_2^{(k)} = \arg\min_x \mathcal{L}_\rho \left(x_1^{(k)}, x, z^{(k)} \right). \tag{103}$$

- *Dual-variable update*

$$z^{(k+1)} = z^{(k)} + \rho \left(A x_1^{(k)} + B x_2^{(k)} - y \right). \tag{104}$$

This description of ADMM is drawn from the review article [1], to which we refer the interested reader for more details. We now apply the method to Problem (68). In this case the augmented Lagrangian is of the form

$$||L||_* + \lambda ||S||_1 + \langle Z, L + S - Y \rangle + \frac{\rho}{2} ||L + S - M||_{\mathrm{F}}^2. \tag{105}$$

The primal update for the low-rank component is obtained by applying Proposition 1.10.

$$L^{(k)} = \arg\min_L \mathcal{L}_\rho \left(L, S^{(k-1)}, Z^{(k)} \right) \tag{106}$$

$$= \arg\min_L ||L||_* + \left\langle Z^{(k)}, L \right\rangle + \frac{\rho}{2} \left|\left| L + S^{(k-1)} - M \right|\right|_{\mathrm{F}}^2 \tag{107}$$

$$= \mathcal{D}_{1/\rho} \left(\frac{1}{\rho} Z^{(k)} + S^{(k-1)} - M \right). \tag{108}$$

The primal update for the sparse component is obtained by recalling that softhresholding is the proximal operator for the ℓ_1 norm.

$$S^{(k)} = \arg\min_S \mathcal{L}_\rho \left(S, L^{(k-1)}, Z^{(k)} \right) \tag{109}$$

$$= \arg\min_S \lambda ||S||_1 + \left\langle Z^{(k)}, S \right\rangle + \frac{\rho}{2} \left|\left| L^{(k-1)} + S - M \right|\right|_{\mathrm{F}}^2 \tag{110}$$

$$= \mathcal{S}_{\lambda/\rho} \left(\frac{1}{\rho} Z^{(k)} + L^{(k)} - M \right). \tag{111}$$

This yields ADMM for robust PCA, also described under the name ALM (augmented Lagrangian method) in the literature [9].

Algorithm 2.6 (ADMM for robust PCA)**.** *We set an initial value $Z^{(0)}$. Then we iterate between updating the primal and the dual variables.*

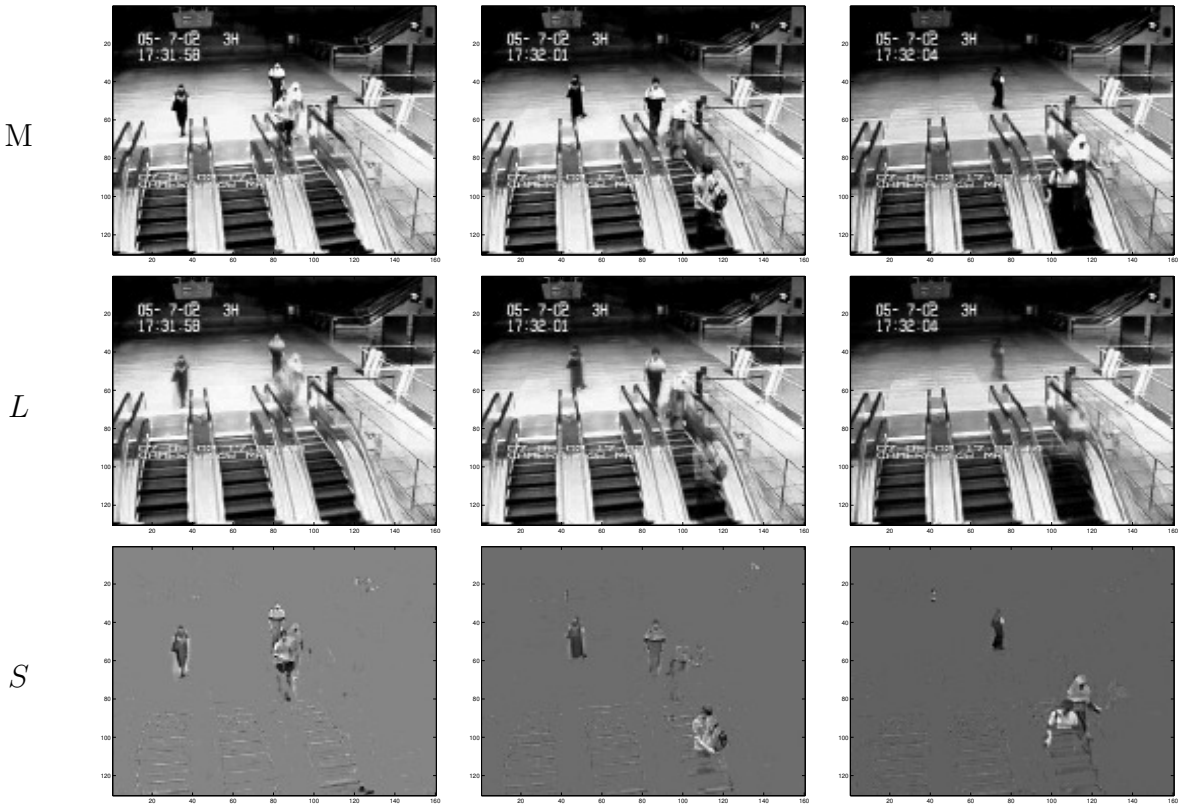


Figure 6: Background subtraction results from a video. This example is due to Stephen Becker. The code is available at `http://cvxr.com/tfocs/demos/rpca`.

- *Primal-variable updates*

$$L^{(k)} = \mathcal{D}_{1/\rho}\left(\frac{1}{\rho}Z^{(k)} + S^{(k-1)} - M\right), \tag{112}$$

$$S^{(k)} = \mathcal{S}_{\lambda/\rho}\left(\frac{1}{\rho}Z^{(k)} + L^{(k)} - M\right). \tag{113}$$

- *Dual-variable update*

$$Z^{(k+1)} = Z^{(k)} + \rho\left(L^{(k)} + S^{(k)} - M\right). \tag{114}$$

2.6 Background subtraction

In computer vision, the problem of background subtraction is that of separating the background and foreground of a video sequence. Imagine that we take a video of a static back-

ground. We then stack the video frames in a matrix M, where each column corresponds to a vectorized frame. If the background is completely static, then all the frames are equal to a certain vector $f \in \mathbb{R}^m$ (m is the number of pixels in each frame) and the matrix is rank 1

$$M = \begin{bmatrix} f & f & \cdots & f \end{bmatrix} = f \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{115}$$

If the background is not completely static, but instead experiences gradual changes, then the matrix containing the frames will be approximately low rank. Now, assume that there are sudden events in the foreground. If these events occupy a small part of the field of view and do not last very long, then the corresponding matrix can be modeled as sparse (most entries are equal to zero). These observations motivate applying the robust PCA method to background subtraction. We stack the frames as columns of a matrix and separate the matrix into a low-rank and a sparse component. The results of applying this method to a real video sequence are shown in Figure 6.

References

Apart from the references cited in the text, the book [7] discusses matrix completion in Chapter 7.

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[2] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[3] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[4] E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *Information Theory, IEEE Transactions on*, 56(5):2053–2080, 2010.

[5] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

[6] D. Gross. Recovering low-rank matrices from few coefficients in any basis. *Information Theory, IEEE Transactions on*, 57(3):1548–1566, 2011.

[7] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC Press, 2015.

[8] P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.

[9] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

A Proofs

A.1 Proof of Proposition 1.5

The following simple lemma will be very useful. We omit the proof.

Lemma A.1. *For any $m \times n$ matrices A and B*

$$\operatorname{trace}(AB) = \operatorname{trace}(BA). \tag{116}$$

The proof relies on the following two lemmas.

Lemma A.2. *For any $Q \in \mathbb{R}^{m\times n}$, $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$, if $U^TU = I$ and $V^TV = I$ then*

$$||UQV|| = ||Q||. \tag{117}$$

Proof. By the definition of the operator norm,

$$\begin{aligned}
\left|\left|UQV^T\right|\right| &= \sup_{||x||_2=1} ||UQVx||_2 && (118)\\
&= \sup_{||x||_2=1} \sqrt{x^TV^TQ^TU^TUQVx} && (119)\\
&= \sup_{||x||_2=1} \sqrt{x^TV^TQ^TQVx} && (120)\\
&= \sup_{||y||_2=1} \sqrt{y^TQ^TQy} \quad \text{because } ||x||_2 = ||Vx||_2 && (121)\\
&= ||Q||. && (122)
\end{aligned}$$

□

Lemma A.3. *For any $Q \in \mathbb{R}^{n\times n}$*

$$\max_{1\le i\le n} |Q_{ii}| \le ||Q||. \tag{123}$$

Proof. We denote the standard basis vectors by e_i, $1 \leq i \leq n$. Since $||e_i||_2 = 1$,

$$\max_{1 \leq i \leq n} |Q_{ii}| \leq \max_{1 \leq i \leq n} \sqrt{\sum_{j=1}^{n} Q_{ji}^2} \tag{124}$$

$$= \max_{1 \leq i \leq n} ||Q\, e_i||_2 \tag{125}$$

$$\leq \sup_{||x||_2 = 1} ||Q\, x||_2 \,. \tag{126}$$

□

By Lemma A.1, if the singular value decomposition of A is $U \Sigma V^T$ then

$$\sup_{||B|| \leq 1} \operatorname{tr}\left(A^T B\right) = \sup_{||B|| \leq 1} \operatorname{tr}\left(V\, \Sigma\, U^T B\right) \tag{127}$$

$$= \sup_{||B|| \leq 1} \operatorname{tr}\left(\Sigma\, B U^T V\right). \tag{128}$$

By Lemma A.2 BU^TV has operator norm equal to $||B|| = 1$. By Lemma A.3 this implies that its diagonal entries have magnitudes bounded by one. We conclude that

$$\sup_{||B|| \leq 1} \operatorname{tr}\left(A^T B\right) \leq \sup_{\left\{M \mid \max_{1 \leq i \leq n} |M_{ii}| \leq 1\right\}} \operatorname{tr}\left(\Sigma\, M\right) \tag{129}$$

$$\leq \sup_{\left\{M \mid \max_{1 \leq i \leq n} |M_{ii}| \leq 1\right\}} \sum_{i=1}^{n} M_{ii}\, \sigma_i \tag{130}$$

$$\leq \sum_{i=1}^{n} \sigma_i \tag{131}$$

$$= ||A||_* \,. \tag{132}$$

To complete the proof, we need to show that the equality holds. Note that UV^T has operator norm equal to one because its r singular values (recall that r is the rank of A) are equal to one. We have

$$\left\langle A, UV^T \right\rangle = \operatorname{trace}\left(A^T U V^T\right) \tag{133}$$

$$= \operatorname{trace}\left(V\, \Sigma\, U^T U V^T\right) \tag{134}$$

$$= \operatorname{trace}\left(V^T V\, \Sigma\right) \quad \text{by Lemma A.1} \tag{135}$$

$$= \operatorname{trace}\left(\Sigma\right) \tag{136}$$

$$= ||A||_* \,. \tag{137}$$

A.2 Proof of Proposition 1.10

Due to the Frobenius norm term, the cost function is strictly convex. This implies that any point at which there exists a subgradient that is equal to zero is the solution to the optimization problem. The subgradients of the cost function at X are of the form,

$$X - Y + \tau G, \tag{138}$$

where G is a subgradient of the nuclear norm at X. If we can show that

$$\frac{1}{\tau}\left(Y - D_\tau\left(Y\right)\right) \tag{139}$$

is a subgradient of the nuclear norm at $D_\tau\left(Y\right)$ then $D_\tau\left(Y\right)$ is the solution.

Let us separate the singular-value decomposition of Y into the singular vectors corresponding to singular values greater than τ, denoted by U_0 and V_0 and the rest

$$Y = U\,\Sigma V^T \tag{140}$$

$$= \begin{bmatrix} U_0 & U_1 \end{bmatrix} \begin{bmatrix} \Sigma_0 & 0 \\ 0 & \Sigma_1 \end{bmatrix} \begin{bmatrix} V_0 & V_1 \end{bmatrix}^T. \tag{141}$$

Note that $D_\tau\left(Y\right) = U_0\left(\Sigma_0 - \tau I\right)V_0^T$, so that

$$\frac{1}{\tau}\left(Y - D_\tau\left(Y\right)\right) = U_0 V_0^T + \frac{1}{\tau} U_1 \Sigma_1 V_1^T. \tag{142}$$

By construction all the singular values of $U_1\Sigma_1 V_1^T$ are smaller than τ, so

$$\left|\left|\frac{1}{\tau} U_1 \Sigma_1 V_1^T\right|\right| \leq 1. \tag{143}$$

In addition, by definition of the singular-value decomposition $U_0^T U_1 = 0$ and $V_0^T V_1 = 0$. As a result, (142) is a subgradient of the nuclear norm at $D_\tau\left(Y\right)$ and the proof is complete.