Sparse Coding Models of Natural Images: Algorithms for Efficient Inference and Learning of Higher-Order Structure

Pierre Jerome Garrigues

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2009-71
http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-71.html

May 20, 2009

Sparse Coding Models of Natural Images: Algorithms for Efficient Inference and Learning of Higher-Order Structure

by

Pierre Jérôme Garrigues

INGEN (Ecole Polytechnique, Paris) 2003
M.S. (University of California, Berkeley) 2005

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Engineering-Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Bruno Olshausen, Chair
Professor Laurent El Ghaoui
Professor Michael DeWeese

Spring 2009

The dissertation of Pierre Jérôme Garrigues is approved:

Chair Date

Date

Date

University of California, Berkeley

Sparse Coding Models of Natural Images: Algorithms for Efficient Inference and Learning of Higher-Order Structure

by

Pierre Jérôme Garrigues

Abstract

Sparse Coding Models of Natural Images: Algorithms for Efficient Inference and Learning of Higher-Order Structure

by

Pierre Jérôme Garrigues

Doctor of Philosophy in Engineering-Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Bruno Olshausen, Chair

The concept of sparsity is widely used in the signal processing, machine learning, and statistics communities for model fitting and solving inverse problems. It is also important in neuroscience as it is thought to underlie the neural representations used in the brain. In this thesis, I derive new algorithms for learning higher-order structure in sparse coding models of images, and I present an improved algorithm for inferring sparse representations with sequential observations.

It has been shown that adapting a dictionary of basis functions to the statistics of natural images so as to maximize sparsity in the coefficients results in a set of dictionary elements whose spatial properties resemble those of primary visual cortex receptive fields. The operation to compute the sparse coefficients can be implemented via an ℓ_1-penalized least-square problem commonly referred to as Basis Pursuit Denoising or Lasso. However, the resulting sparse coefficients still exhibit pronounced statistical dependencies, thus violating the independence assumption of the sparse coding model. I propose in this thesis two models that attempt to capture the dependencies among the basis function coefficients. The first model includes a pairwise coupling term in the prior over the coefficient activity states. When adapted to

the statistics of natural images, the coupling terms converge to a solution involving facilitatory and inhibitory interactions among neighboring basis functions. In the second model, the prior is a mixture of Laplacian distributions, where the statistical dependencies among the basis function coefficients are modeled through the scale parameters. I show that I can leverage the efficient algorithms developed for Basis Pursuit Denoising to derive improved inference algorithms with the Laplacian scale mixture prior.

I also propose in this thesis a new algorithm, RecLasso, to solve the Lasso with online observations. I introduce an optimization problem that allows us to compute an homotopy from the current solution to the solution after observing a new data point. I compare RecLasso to Lars and Coordinate Descent, and present an application to compressive sensing with sequential observations. I also propose an algorithm to automatically update the regularization parameter after observing a new data point.

Chair Date

Acknowledgements

This dissertation would not have been possible without the help and support of many friends and colleagues.

First, I would like to thank my advisor Bruno. I was truly inspired by his dedication to research, his ability to communicate scientific ideas, and his vision of the important problems in the field. I would also like to thank Laurent and Mike for accepting such supportive committee members. Laurent's convex optimization course had a profound influence on my research, and was the start of a fruitful collaboration.

I could not have thought of a better place to do research than the Redwood Center for Theoretical Neuroscience. The work in this thesis would not have been possible without the constant exchange of ideas, constructive criticism, and genuine enthusiasm for research of all its members. My office mates Amir, Charles, Jack, Jascha and Jimmy always made me look forward to being in the lab. Thank you to Chris for introducing me to compressive sensing, to Kilian for showing me what it really means to be into science, and to David for your help and constant encouragement.

I spent wonderful years at Berkeley thanks to my friends Jean, Joe, Nicholaus, Nicolas, Pascal, Sebastien, Stephane, and many others. I would like to thank my parents, Antoine and Marie-Laure, who were very supportive all along and are great role models. Thank you to Vincent for being the best bro. Thank you to my grandmother Miette for encouraging me to pursue my interests in mathematics when I was in high school, and giving me high standards of achievement. Finally, I would like to thank Susan for everything, you make me very happy.

Dedicated to my parents, Antoine and Marie-Laure, and my brother, Vincent.

Contents

Chapter 1

Introduction

1.1 Motivation

1.1.1 Natural images

Natural images are the typical images that we see as we interact with our environment. For instance, a person taking a walk in a forest would encounter images such as the ones in Figure 1.1. The retina senses the visual world via its photoreceptor cells which absorb photons and signal the light information via a change in membrane potential. This information is used by the visual system to form a *representation* of the visual world that is subsequently used to interact with the world and accomplish tasks such as navigation or object recognition and grasping. To gain insight into what this representation might be I will study in this work natural *photographic* images. Such images are sensed by the CCD array of the digital camera that turns light into discrete signals, and the image in its raw format is represented as a collection of pixels.

When looking at the images in Figure 1.1, one does not perceive the individual pixel values, but rather trees, foliage, or the three-dimensional structure of the scene.

The problem of image representation is to understand how the pixels are mapped to our percepts. This is a task that we have to perform as humans to interact with the world, and our brains have evolved an incredibly sophisticated machinery to accomplish the task of interpreting the visual signals sensed in our retina. No computer algorithm to date is able to replicate a human's ability to interpret complex visual inputs.

We consider images as vectors $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, where the x_i's are the pixels and n is the number of pixels. Images are high-dimensional; with modern digital cameras, n is typically in the order of 10 millions. However, natural images occupy a very small portion of $\mathbb{R}^n$. Indeed, if we pick an element at random in this space, it is extremely unlikely that it will look anything like the type of visual input we typically encounter. The field of natural image statistics aims to capture the complex structure of the space of natural images. Understanding the statistics of natural images will allow us to derive representations such that the underlying causes giving rise to our percepts are explicit. Such a representation can then be used to teach a machine how to interpret images and replicate the human visual system's performance. Furthermore, the learned representations provide a hypothesis for how the brain is representing images, and the resulting predictions can then be tested experimentally.

1.1.2 Efficient coding hypothesis

The efficient coding hypothesis [Barlow, 1961][Attneave, 1954] assumes that nervous systems exploit the statistical dependencies contained in sensory signals. It supposes that the brain has internalized the statistics of its sensory inputs and represents them optimally. Hence, we can gain insight into what types of representations are used in the brain and how they are computed by developing probabilistic models of natural

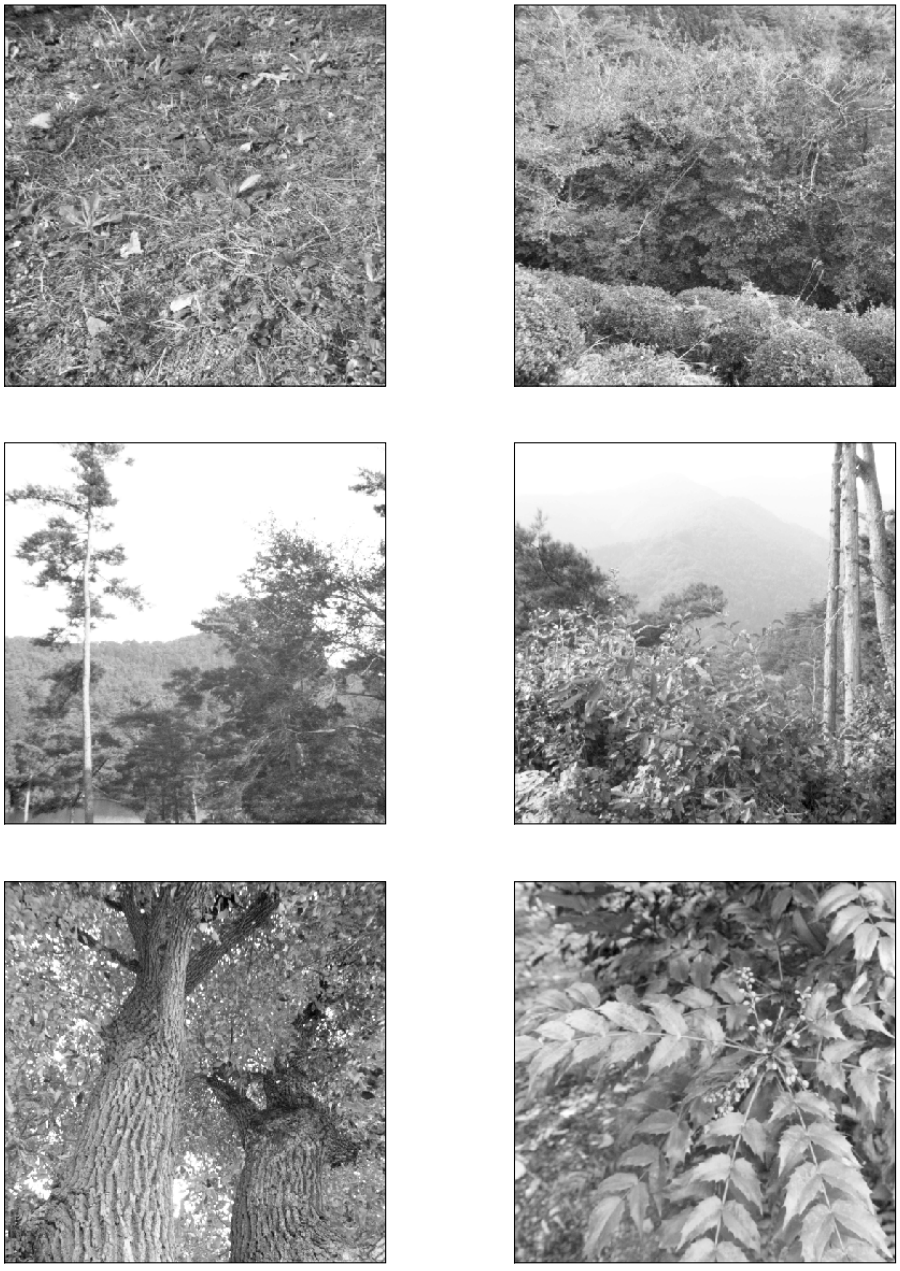


Figure 1.1: Natural scenes taken from the Kyoto database [Doi *et al.*, 2003].

images that capture their structure.

For instance, natural images are *redundant* in that there exist statistical dependencies among the pixel values. In a biological system, it is therefore wasteful to represent each "pixel" value individually, and the visual system should reduce redundancy by removing statistical dependencies [Barlow, 1961]. We will see that an important characteristic of natural images is that they are *sparse*, and that this property can be used to explain early visual processing.

1.1.3 Applications to inverse problems

Having a prior $p(x)$ for natural images is necessary for all inverse problems encountered in image processing. Let us consider for instance the problem of image denoising. An image x is corrupted by some noise ν, resulting in the noisy image

$$y = x + \nu. \tag{1.1}$$

The goal of denoising is to recover the original image x. This problem is ill-posed: there are many combinations of x and ν such that (1.1) is verified. However, few of those combinations are such that x is a natural image having high probability under the image model $p(x)$, which allows us to regularize this problem. We consider as our solution the maximum a posteriori (MAP) estimate of x given y. Using Bayes' rule we can write

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} = \frac{p_\nu(y - x)p(x)}{p(y)},$$

where p_ν is the probability distribution of the noise. If the noise is independent identically distributed (i.i.d.) Gaussian with variance σ^2, we have

$$p_\nu(\nu) = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\frac{1}{2\sigma^2}\|\nu\|_2^2}.$$

The solution of the denoising problem is therefore given by

$$\hat{x} = \arg\max_x \; p_\nu(y - x)p(x).$$

Image priors have been used successfully for image denoising [Portilla *et al.*, 2003][Elad and Aharon, 2006]. Other inverse problems using image priors are in-painting [Roth and Black, 2009], matting [Levin and Weiss, 2007], or deconvolution [Fergus *et al.*, 2006]. The image priors that are used in these algorithms are far from capturing all the structure in natural images. Constructing better models of images is therefore an important problem as it will result in increased performance in image inverse problems.

1.2 Second order image models

Images are composed of large smooth regions where pixel values change rather slowly. Hence, the pixels in images are highly correlated. We review in this Section probabilistic models that capture the second-order correlations in images, and look at their descriptive power.

1.2.1 Amplitude spectrum of images

The pixel correlations are captured by the pixel autocorrelation function, whose Fourier transform is the power spectrum. We first review an empirical observation concerning the decay rate of the power spectrum in *individual* natural images. Let $x[i, j]$ denote the pixel at the position (i, j) in an image x whose dimensions are p by

p pixels. The 2D Fourier transform of x is given by

$$X[f_i, f_j] = \sum_{i,j=0}^{p} x[i,j] e^{-j\frac{2\pi}{N}(f_i i + f_j j)} \tag{1.2}$$

$$= A[f_i, f_j] e^{j\Theta(f_i, f_j)}, \tag{1.3}$$

where $A[f_i, f_j]$ is the amplitude spectrum of x, and $\Theta[f_i, f_j]$ is the phase. The power spectrum $A[f_i, f_j]^2$ gives information about the distribution of the signal's energy among the different spatial frequencies.

Let $A[f]^2$ be the power spectrum of $A[f_i, f_j]^2$ averaged over all orientations, i.e. such that $f_i^2 + f_j^2 = f^2$. We show in Figure 1.2 the average amplitude spectrum $A[f]$ for each image that appeared in Figure 1.1. We can see that even though these images are rather different and contain diverse types of structures, their amplitude spectrum decays with a similar profile that can be approximated by $1/f$ [Field, 1987]. This property is in fact consistent over most natural images.

1.2.2 A Gaussian image model

Let $\mathcal{D} = \{x^{(1)}, \ldots, x^{(N)}\}$ be a dataset of image patches $x_i \in \mathbb{R}^n$ that have been randomly selected from the collection of images shown in Figure 1.1. We show in Figure 1.3 a collection of 100 such patches. We denote by p^* the empirical distribution, i.e.

$$p^*(x) = \frac{1}{N} \sum_{k=1}^{N} \delta(x - x^{(k)}).$$

The empirical correlation between pixel i and pixel j is

$$< x_i x_j >_{p^*} = \frac{1}{N} \sum_{k=1}^{N} x_i^{(k)} x_j^{(k)},$$

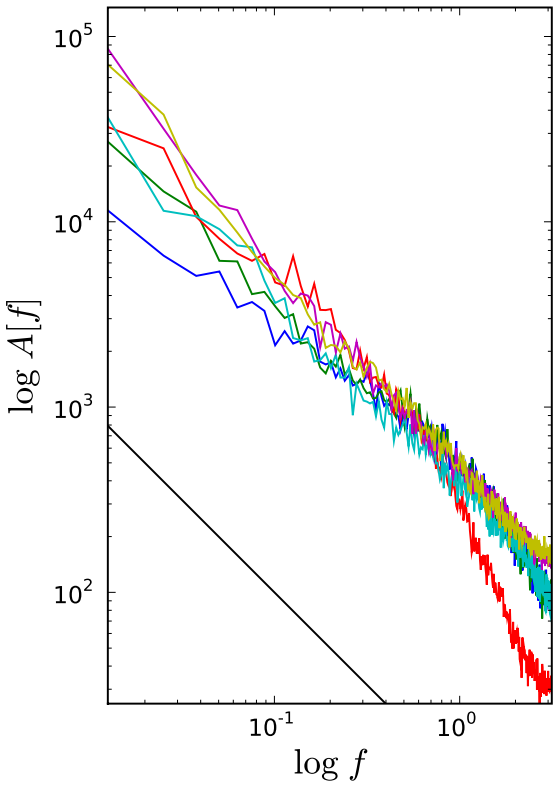


Figure 1.2: Amplitude spectrum decay of the images in Figure 1.1, where the spectrum is averaged over all orientations. The black line corresponds to a slope of -1.

Figure 1.3: Image patches of dimension 16×16 selected from a collection of gray-scale natural images whose pixel values are between 0 and 255. We show patches whose pixel variance is above 20 which has the effect of rejecting patches coming from uniform regions such as the sky.

where $< . >_{p^*}$ denotes the expectation with respect to the distribution p^*. There are many probabilistic models $p(x)$ that capture the second-order correlations, i.e. where $< x_i x_j >_p = < x_i x_j >_{p^*}$ for every pixel i and j. It can be shown that the family of distributions with fixed second-order correlations and highest entropy is the family of Gaussian distributions parameterized by a mean μ and covariance matrix Σ

$$p(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} \det \Sigma^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}.$$

Hence, for our ensemble of image patches $\mathcal{D}$, we wish to find the parameters of the Gaussian model μ and Σ that best fit the data which is typically done via maximum likelihood

$$\max_{\mu,\Sigma} \ \langle \log p(x \mid \mu, \Sigma) \rangle_{p^*} .$$

It can be shown easily that the derivative of the cost function is zero when

$$\begin{cases} \hat{\mu} = \frac{1}{N} \sum_{k=1}^{N} x^{(k)} \\ \hat{\Sigma} = \frac{1}{N} \sum_{k=1}^{N} (x^{(k)} - \hat{\mu})(x^{(k)} - \hat{\mu})^T \end{cases}$$

Note that this estimate of the covariance matrix is biased. Now that we have a Gaussian model, we can rotate image patches such that the representation is aligned with the axis of maximum variance. The covariance matrix $\hat{\Sigma}$ is symmetric and can be written $\hat{\Sigma} = V \Gamma V^T$, where Γ is a diagonal matrix whose elements are the eigenvalues of $\hat{\Sigma}$, and V is an orthogonal matrix or rotation. Let $a = V^T x$. We have

$$\left\langle a a^T \right\rangle = \left\langle V^T x (V^T x)^T \right\rangle = V^T \left\langle x x^T \right\rangle V = V^T \hat{\Sigma} V = V^T V \Gamma V^T V = \Gamma.$$

Hence, the representation a does not have second-order correlations. The filters that map x to a are the columns of V and displayed in Figure 1.4. We can see that they

resemble a Fourier transform, which is also a consequence of the Toeplitz structure of the covariance matrix. We show in Figure 1.5 the decay of the corresponding eigenvalues.

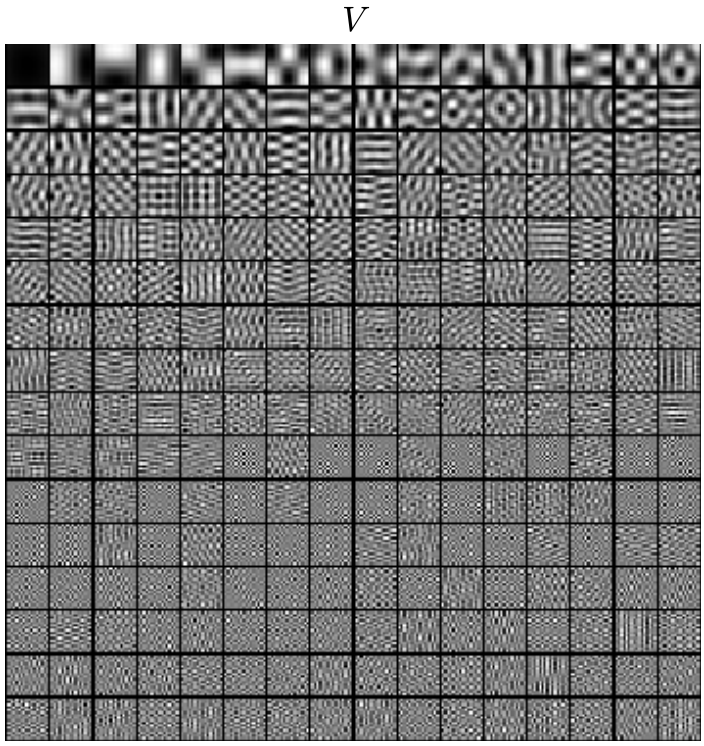


Figure 1.4: Eigenvectors of the covariance matrix for an ensemble of 100000 16×16 image patches.

1.2.3 Samples from the model

We can get an intuition for the descriptive power of the Gaussian model by sampling from the model and looking at whether the samples look "natural." We have seen that fitting a Gaussian to a collection of image patches leads to a Covariance matrix

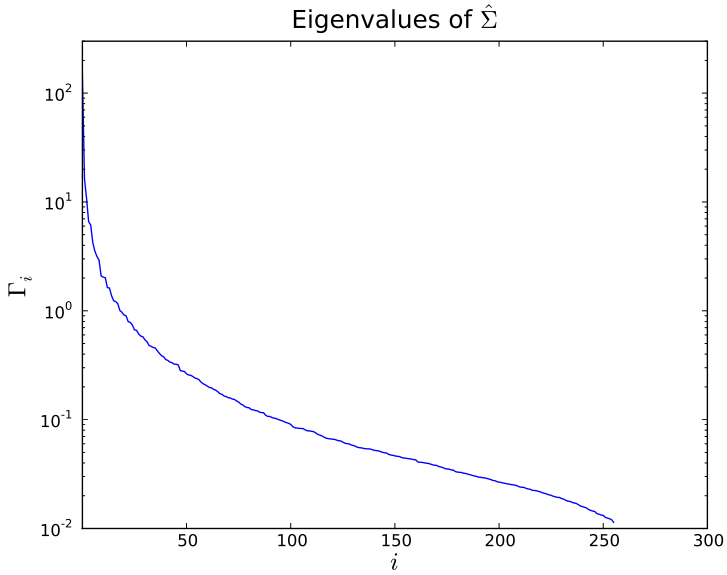


Figure 1.5: Eigenvalues of the covariance matrix for an ensemble of 100000 16×16 image patches.

that is diagonal in the Fourier basis, and where the power spectrum decays as $1/f^2$. To sample a large image from such a distribution, we first sample a white noise image and filter it such that its spectrum obeys the $1/f^2$ property. We can see in Figure 1.6 that such a sample does not capture the edge structure in images, and the resulting image looks "cloudy." We show in Figure 1.6 an image that has been whitened, i.e. its second order correlations have been removed [Olshausen and Field, 1997]. We can see that most of its interesting edge-like structure remains.

It is intuitive that the Gaussian model is not sufficient to capture the structure in natural images. Edges are indeed characterized by higher-order correlations, and therefore cannot be captured by a second-order model. The Gaussian model has been widely used in image denoising, an approach that is known as Wiener filtering.

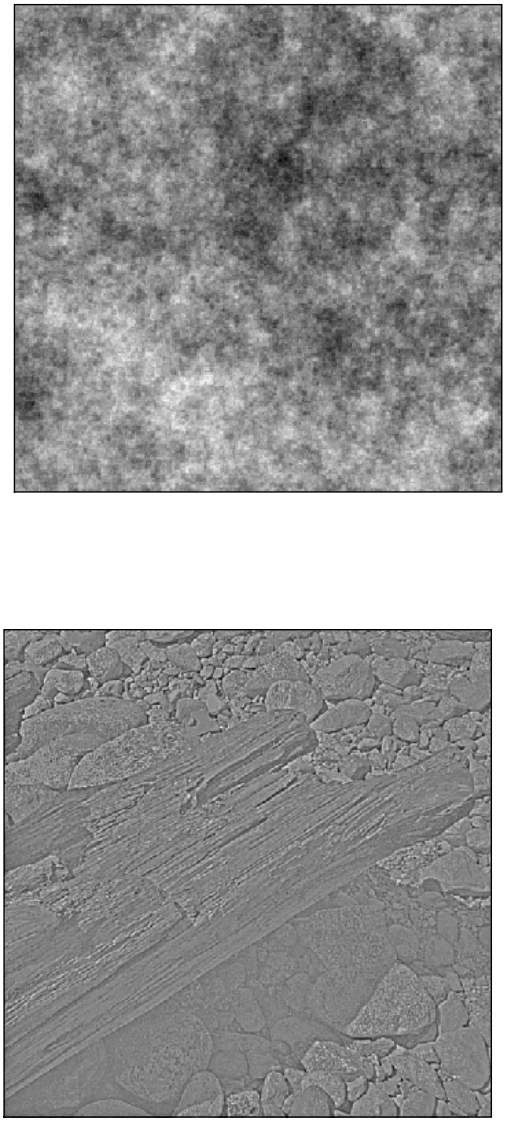


Figure 1.6: **Top** A "pink noise" sample from the Gaussian model. **Bottom** A whitened image.

1.3 Analysis-based image prior

In this Section we review a class of probabilistic models that have the ability to model higher-order dependencies. The principle in *analysis*-based modeling is to model the probability distribution of *forward* projections of natural images. Note that the Gaussian model is an analysis-based model as $a = Vx$ can be modeled with a factorial Gaussian distribution.

1.3.1 Evidence of sparsity in natural scenes

The kurtosis of a random variable is a measure of how a distribution is "peaked" around its mean. A random variable with high kurtosis has its realizations mostly around its mean, with some large deviations. The kurtosis of random variable Z is defined as

$$\kappa(Z) = \frac{\mathbb{E}[(Z - \mathbb{E}[Z])^4]}{\mathbb{E}[(Z - \mathbb{E}[Z])^2]}.$$

For instance, a Gaussian random variable has its kurtosis equal to 3.

It has been observed that the histogram of linear responses of natural images to oriented edge filters have kurtotic histograms [Field, 1994]. The intuition is that natural images are composed of extended smooth regions where the response of a filter is small, and discontinuities at contours where the response of the filter is large if the orientation of the contour matches its orientation. Kurtosis is an important property of natural images that provides evidence for higher-order statistical dependencies. For instance, all projections of a multivariate Gaussian random variable are Gaussian themselves. As an illustration, we convolved the image Lena with an oriented filter as shown in Figure 1.7. We see in Figure 1.8 that the histogram of the convolved image is indeed heavy-tailed and has kurtosis 7.4.

Kurtosis for subbands of natural images varies with filter bandwidth and is max-

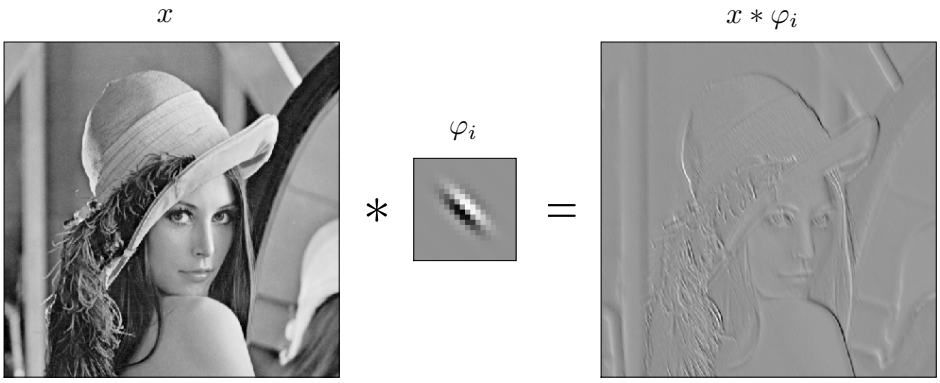


Figure 1.7: Convolution of Lena with an oriented Gabor filter.

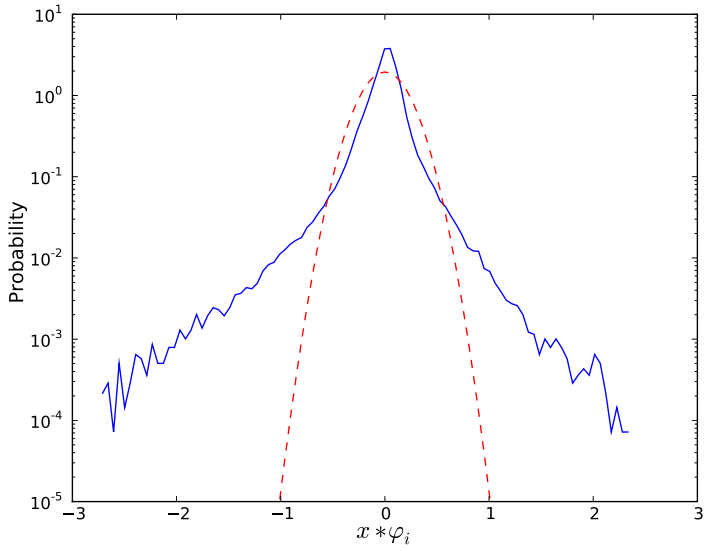


Figure 1.8: Histogram of the convolved image. It is peaked around zero and has heavy tails. The estimated kurtosis is 7.4. We show in dashed red the log-probability of the Gaussian distribution with equal variance.

imal at roughly one octave [Field, 1994]. The filters in a wavelet transform have a frequency support of one octave and have been used successfully in image processing. The probability density function of a wavelet coefficients is modeled in [Mallat, 1989] with a generalized Gaussian distribution

$$p(a) \propto e^{-|\lambda a|^p}.$$

Denoising individual wavelet coefficients using such a model is known as "coring" [Simoncelli and Adelson, 1996], where the non-linear operator resembles a "soft" threshold. It provides superior results to the Wiener filter, as it has a better ability to preserve the edge structure.

1.3.2 Independent component analysis

The goal of Independent Component Analysis (ICA) is to find a linear mapping $W \in \mathbb{R}^{n \times n}$ such that the ouptouts $a = Wx$ are independent and have sparse distributions [Bell and Sejnowski, 1997]. Let w_i^T be the i^{th} row of W. The likelihood of an image x in this model is given by

$$p(x \mid W) = \det(W) \prod_{i=1}^{n} q(w_i^T x),$$

where q is a heavy tailed distribution. Common choices for q shown in Figure 1.9 are

$$q(a_i) = \begin{cases} \frac{1}{2}\lambda e^{-\lambda|a_i|}, & \text{Laplacian distribution} \\ \frac{\Gamma\left(\frac{\alpha+1}{2}\right)}{\sqrt{\alpha\pi}\Gamma\left(\frac{\alpha}{2}\right)} \left(1 + \frac{a_i^2}{\alpha}\right)^{\frac{\alpha+1}{2}}, & \text{Student-t distribution} \end{cases} \tag{1.4}$$

For our ensemble of image patches $\mathcal{D}$, we wish to find the parameters of the ICA model W that best fit the data. Maximizing the log-likelihood leads to the cost

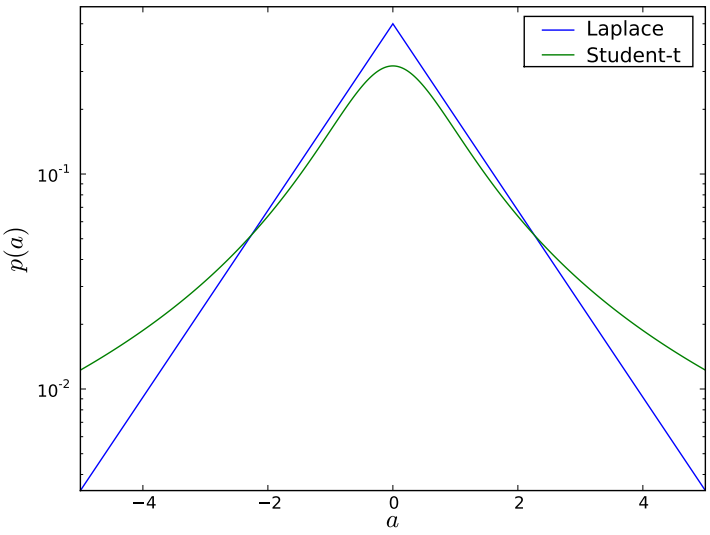


Figure 1.9: Laplacian distribution with $\lambda = 1$ and Student-t distribution with parameter $\alpha = 1$. The Laplacian distribution is more peaked at zero whereas the Student-t has heavier tails.

function

$$\langle \log p(x \mid W) \rangle_{p^*} = \log \det(W) + \left\langle \sum_{i=1}^{n} \log q(w_i^T x) \right\rangle_{p^*},$$

and when it is maximized leads to the solution shown in Figure 1.10 learned using the FastICA algorithm [Hyvärinen, 1999]. We can see that the learned filters are localized and oriented, and resemble the receptive fields of simple cells in visual cortex. Edges are important in natural images, and can be detected by linear filtering.

1.3.3 Circular dependencies

The ICA model supposes that the outputs a_i are independent. However, when learning the optimal transfrom W, the responses of the elements of the code are in fact *not* independent. An interesting example of the types of filters learned by ICA that exhibit dependencies are filters in quadrature pair. Let φ_i and φ_j denote such fil-

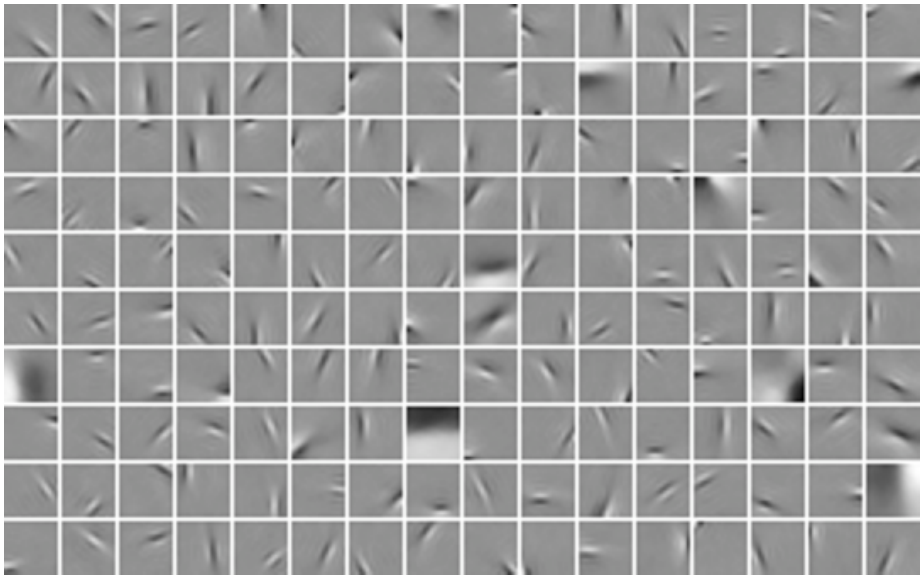


Figure 1.10: Learned ICA filters for 16×16 image patches. The patches have their dimensionality reduced to 160 which is the number of learned filters.

ters shown in Figure 1.11. We denote as their responses $a_i = \varphi_i^T x$ and $a_j = \varphi_j^T x$, and we get an estimate of their distribution by convolving those filters with a set of images. The histograms of a_i and a_j are both sparse and similar to the histogram shown in Figure 1.8. If the responses a_i and a_j were independent, then their joint probability would be given by $p(a_i, a_j) = p(a_i)p(a_j)$. We illustrate this distribution in Figure 1.12. Note that the iso-contours are diamond-shaped. However, the iso-contours of the joint distribution $p(a_i, a_j)$ are in fact *circular* [Zetzsche *et al.*, 1999]. Hence the coefficients a_i and a_j are *not* independent. This property is not limited to quadrature pair filters, and is consistent over pairs of Gabor-like filters that have a similar position, orientation, and scale. We show such an example in Figure 1.13.

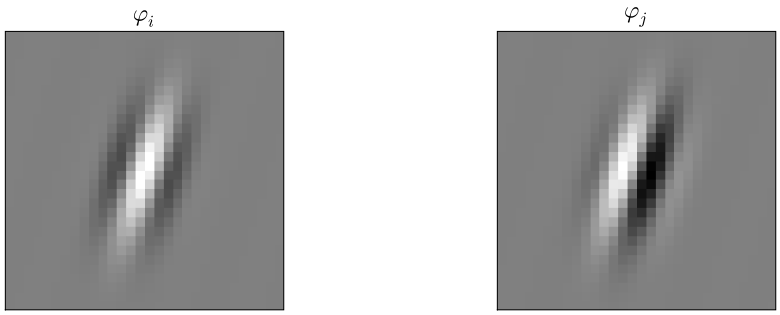


Figure 1.11: Quadrature pair filters.

1.3.4 Gaussian scale mixtures

Those circular dependencies are informative and should be exploited. The Gaussian scale mixture (GSM) introduced in [Wainwright *et al.*, 2001b] proposes a probabilistic model that makes it possible to model the distribution of two random variables (a_i, a_j) such that their marginals $p(a_i)$ and $p(a_j)$ are sparse, and their joint distribution $p(a_i, a_j)$ has circular iso-contours.

The random variable a_i is a Gaussian scale mixture if it can be written $a_i = z_i u_i$,

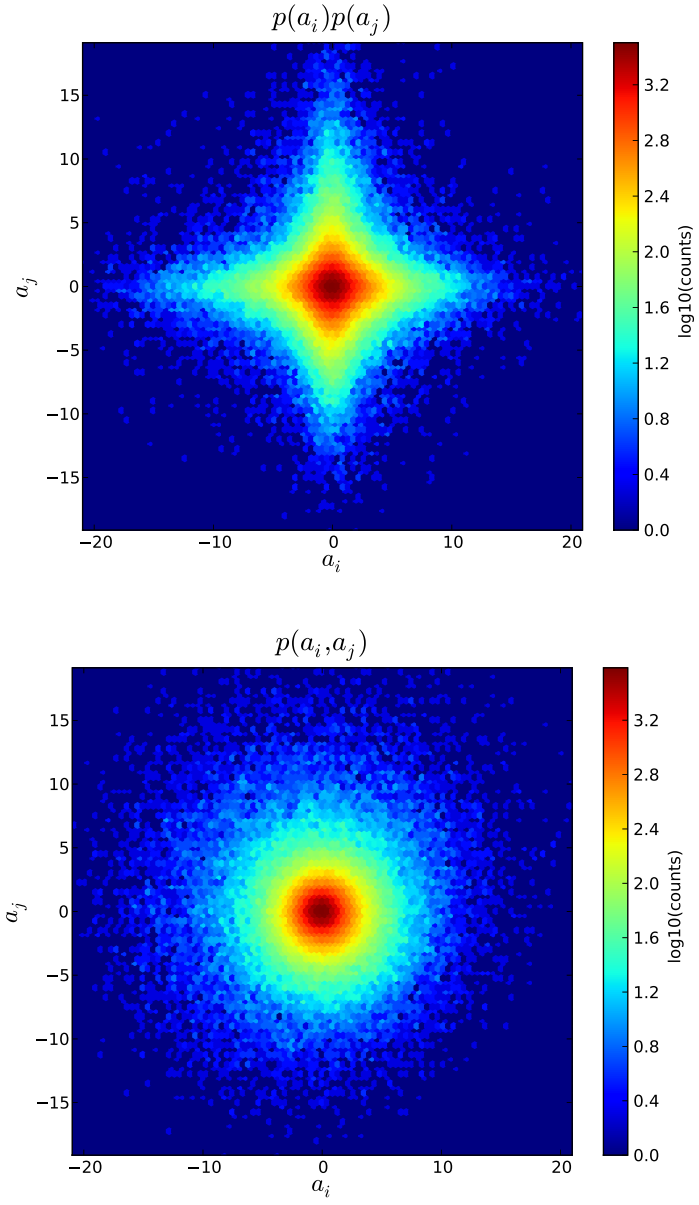


Figure 1.12: Joint statistics of the responses of quadrature pair filters.

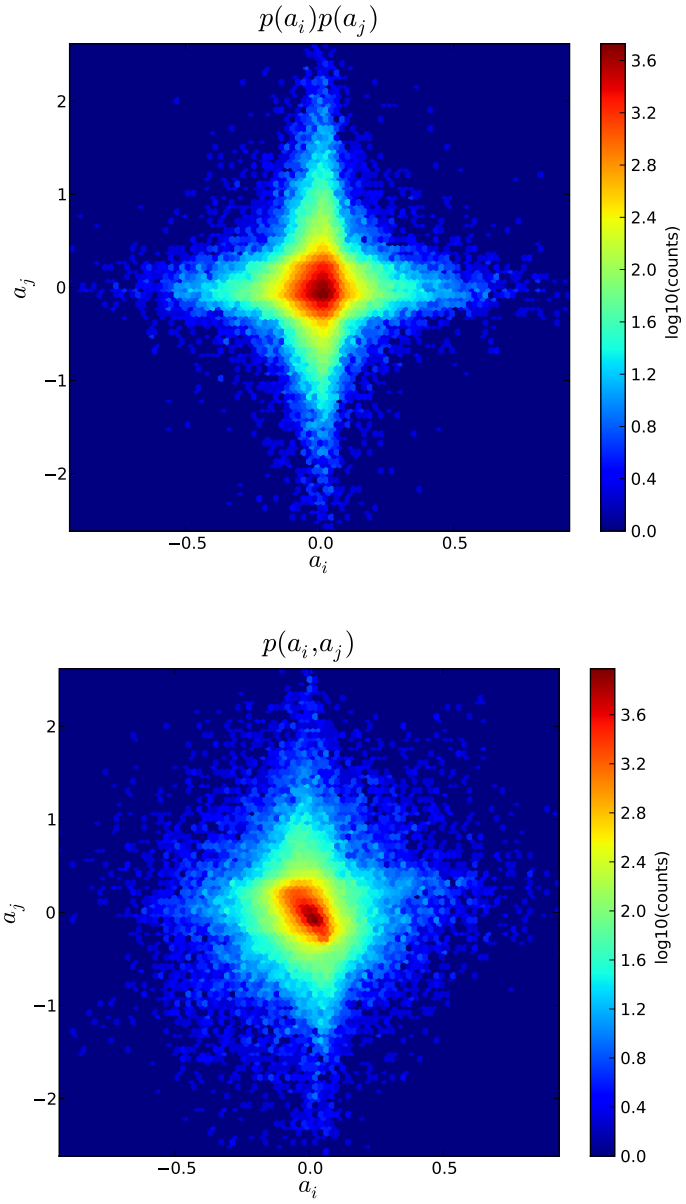


Figure 1.13: Joint statistics of the responses of filters that have a similar position, orientation, and scale.

where u_i is a zero mean Gaussian with unit-variance, and z_i is a random variable that is positive only and is called the *multiplier* variable. The marginal of a_i is given by

$$\begin{aligned} p(a_i) &= \int_z p(a_i \mid z) p(z) dz \\ &= \int_z \frac{1}{\sqrt{2\pi z^2}} e^{-\frac{a_i^2}{2z^2}} p(z) dz. \end{aligned}$$

Hence a_i is a continuous mixture of Gaussian distributions, and for most choices of $p(z)$ this leads to a distribution with heavy tails. It has been shown in [Wainwright *et al.*, 2001b] that GSMs fit well the coefficients of a wavelet transform.

The dependencies between a_i and a_j can be modeled via dependencies between their multiplier variables z_i and z_j. Consider the simple case where these multiplier variables are the same, i.e.

$$\begin{cases} a_i = z u_i \\ a_j = z u_j, \end{cases}$$

and u_i and u_j are independent. The corresponding graphical model is shown in Figure 1.14. The nodes corresponding to the observed variables a_i and a_j are shaded in gray. Given the latent variable z, a_i and a_j are independent, i.e. $p(a_i, a_j \mid z) = p(a_i \mid z) p(a_j \mid z)$. The joint probability is given by

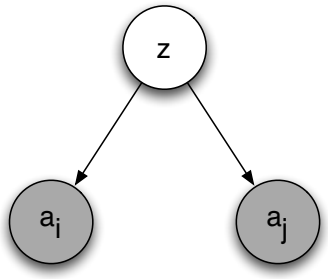


Figure 1.14: Graphical model of a simple GSM model.

$$\begin{aligned} p(a_i, a_j) &= \int_z p(a_i, a_j \mid z) p(z) dz \\ &= \int_z p(a_i \mid z) p(a_j \mid z) p(z) dz \\ &= \int_z \left(\frac{1}{\sqrt{2\pi z^2}} \right)^2 e^{-\frac{a_i^2 + a_j^2}{2z^2}} p(z) dz. \end{aligned}$$

Hence the joint probability depends on $a_i^2 + a_j^2$, and its iso-contours are circular-symmetric.

The joint dependencies among the elements of a wavelet code are modeled using multiplier variables whose dependencies are governed by a tree-structure Markov model in [Wainwright *et al.*, 2001b], and using a Markov random field within a subband in [Lyu and Simoncelli, 2006]. Modeling the statistics of wavelet coefficients with GSMs leads to state-of-the-art image denoising algorithms [Portilla *et al.*, 2003]. The density components model [Karklin and Lewicki, 2005] is an interesting example of a GSM where the dependencies among the multiplier variables are modeled using latent variables. Let $a = Wx$ be the responses of image patches x to a linear transform W learned using ICA. Karklin and Lewicki write $a = z \odot u$, where $\odot$ denotes the element-wise multiplication, z is the vector of muliplier variables, and u is Gaussian. The log of the vector of multiplier variables is decomposed in a linear basis $\Psi \in \mathbb{R}^{n \times d}$

$$\log z = \Psi b,$$

where $b \in \mathbb{R}^d$ is a vector of latent variables that are independent and have sparse marginals such as (1.4).

The ICA framework has also been extended to account for the residual dependencies in the coefficients. In Independent Subspace Analysis [Hyvärinen and Hoyer, 2000], the coefficients a_i are assumed to be divided into subspaces such that dependencies within the subspaces are allowed, but not across the subspaces. In topographic

ICA [Hyvärinen *et al.*, 2001], a topographic ordering is imposed such that the distance between two coefficients is defined using their higher-order correlations. In tree-based component analysis [Bach and Jordan, 2004], the dependencies among the coefficients are restricted to have a tree-structure that is also learned.

1.4 Synthesis-based image prior

In a synthesis-based model we seek to reconstruct the image x using latent variables that represent the underlying causes of the image. Hence in such a generative model the causes are represented explicitly, which makes it easier to interpret the inferred representations as compared to analysis-based models.

1.4.1 Analysis vs Synthesis

Let $\Phi = [\varphi_1, \ldots, \varphi_m] \in \mathbb{R}^{n \times m}$ be a matrix whose columns are the basis functions. The *analysis* coefficients are given by

$$a = \Phi^T x = (\varphi_1^T x, \ldots, \varphi_m^T x)^T.$$

The analysis coefficients are the correlation of the input signal with all the elements of the code. For example if Φ is a wavelet transform, the analysis coefficients are the wavelet coefficients.

Let $s \in \mathbb{R}^m$ such that

$$x = \Phi s = \sum_{i=1}^{m} s_i \varphi_i.$$

The coefficients s can be used to reconstruct x and are called the *synthesis* coefficients. In general the analysis coefficients cannot be used for synthesis, unless $\Phi\Phi^T = \kappa I_n$ for some constant κ. A matrix verifying this property is called a tight frame, and an

example is the steerable pyramid [Simoncelli *et al.*, 1992].

It has been shown that in multiscale, oriented image pyramids *overcompleteness*, i.e. having more features than the dimensionality of the space ($m > n$), is necessary to ascribe meaning to coefficients [Simoncelli *et al.*, 1992]. In this case the set of synthesis coefficients $\{s : \ \Phi s = x\}$ is infinite. The representation that is best able to reveal the structure in the signal is the one that is maximally sparse [Barlow, 1961], where sparsity is defined as the number of nonzero coefficients and is usually referred to as the ℓ_0 norm. The maximally sparse solution is hence the solution of the optimization problem

$$\min_s \ \|s\|_0 \quad \text{subject to} \quad \Phi s = x. \tag{1.5}$$

This problem is combinatorial and cannot be solved in polynomial time. However, replacing the ℓ_0 norm with the ℓ_1 norm leads to the following convex optimization problem commonly referred to as Basis Pursuit (BP) [Chen *et al.*, 1999]

$$\min_s \ \|s\|_1 \quad \text{subject to} \quad \Phi s = x, \tag{1.6}$$

where the ℓ_1 norm is the sum of the absolute values

$$\|s\|_1 = \sum_{i=1}^{m} |s_i|.$$

The solution of BP are also sparse, and it has been shown that under some conditions on the dictionary Φ the solutions of (1.5) and (1.6) are in fact identical [Donoho, 2006b].

1.4.2 Sparse coding model

In the sparse coding model introduced in [Olshausen and Field, 1996] one seeks to reconstruct an image x in a possibly overcomplete dictionary of features $\{\varphi_1, \ldots, \varphi_m\}$. The proposed generative model is

$$x = \Phi s + \nu = \sum_{i=1}^{m} s_i \varphi_i + \nu, \tag{1.7}$$

where $\nu \sim \mathcal{N}(0, \sigma^2 I_n)$ is small Gaussian noise, and accounts for the part of x that cannot be well modeled by the features φ_i. The coefficients s are independent latent variables that define the representation of x in the dictionary Φ. The corresponding graphical model is shown in Figure 1.15. In the sparse coding model the coefficients

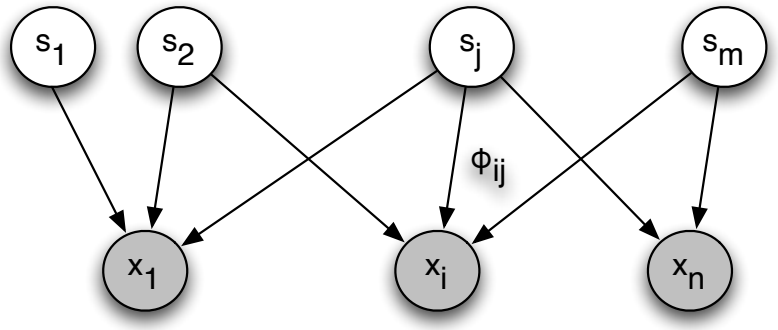


Figure 1.15: Graphical model representation of the sparse coding model.

s have sparse distributions such as the Laplacian or Student-t (1.4). The probability distribution of x is given by

$$\begin{aligned} p(x) &= \int_s p(x \mid s) p(s) ds \\ &= \int_s \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} e^{-\frac{1}{2\sigma^2}\|x - \Phi s\|_2^2} p(s) ds \end{aligned}$$

Note that in general we cannot compute $p(x)$ analytically. However, it is easy to sample from this generative model as long as we can sample from $p(s)$ easily.

1.4.3 Inference

Given an image x, we wish to compute its sparse representation $\hat{s}$ in the dictionary Φ. We consider the MAP estimate given by

$$\begin{aligned} \hat{s} &= \arg\max_s \; p(s \mid x) \\ &= \arg\max_s \; p(x \mid s)p(s) \\ &= \arg\min_s \; \frac{1}{2\sigma^2}\|x - \Phi s\|_2^2 - \sum_{i=1}^{m} \log p(s_i). \end{aligned} \tag{1.8}$$

The sparse representation is therefore the solution of an optimization problem that is the sum of a reconstruction term and a sparsity-inducing term. In the case where the prior on the coefficient is the Laplacian distribution, the objective function takes the form

$$\frac{1}{2\sigma^2}\|x - \Phi s\|_2^2 + \lambda\|s\|_1. \tag{1.9}$$

In this particular case the objective function is convex and is part of the class of quadratic programs. It is often referred to a Basis Pursuit DeNoising (BPDN) and was introduced in [Chen *et al.*, 1999]. The solution is unique and can be computed using efficient algorithms [Efron *et al.*, 2004] [Osborne *et al.*, 2000][Daubechies *et al.*, 2004][Rozell *et al.*, 2007][Friedman *et al.*, 2007][Figueiredo *et al.*, 2007].

1.4.4 Learning

The image prior in the sparse coding model is parameterized by the dictionary $\Phi \in \mathbb{R}^{n \times m}$. The goal of learning is to find the dictionary maximizing the likelihood for the

ensemble of natural images, i.e.

$$\max_{\Phi} \ \langle \log p(x \mid \Phi) \rangle_{p^*} .$$

As we cannot compute the likelihood $p(x \mid \Phi)$ analytically, the learning algorithm in [Olshausen and Field, 1996] proposes the following approximation

$$\begin{aligned} \log p(x \mid \Phi) &= \int_s p(x, s \mid \Phi) ds \\ &\approx p(x, \hat{s} \mid \Phi), \end{aligned}$$

where $\hat{s}$ is the MAP estimate computed during inference

$$\hat{s} = \arg\max_s \ p(s \mid x) = \arg\max_s \ p(x \mid s) p(s).$$

Using this approximation, the dictionary is updated via the learning rule

$$\Delta\Phi = \eta \left\langle (x - \Phi\hat{s})\hat{s}^T \right\rangle ,$$

where the average is taken over a batch of image patches, typically on the order of 100. We show in Figure 1.16 a dictionary learned from a set of whitened images. The basis functions resemble the receptive fields of neurons in primary visual cortex (V1). They tile the frequency plane in a similar manner to wavelet transforms.

1.5 Contributions

In this thesis, I propose new synthesis-based models where the prior over the coefficients is non-factorial. The standard sparse coding model supposes indeed that the coefficients are independent. However, the resulting sparse coefficients still exhibit

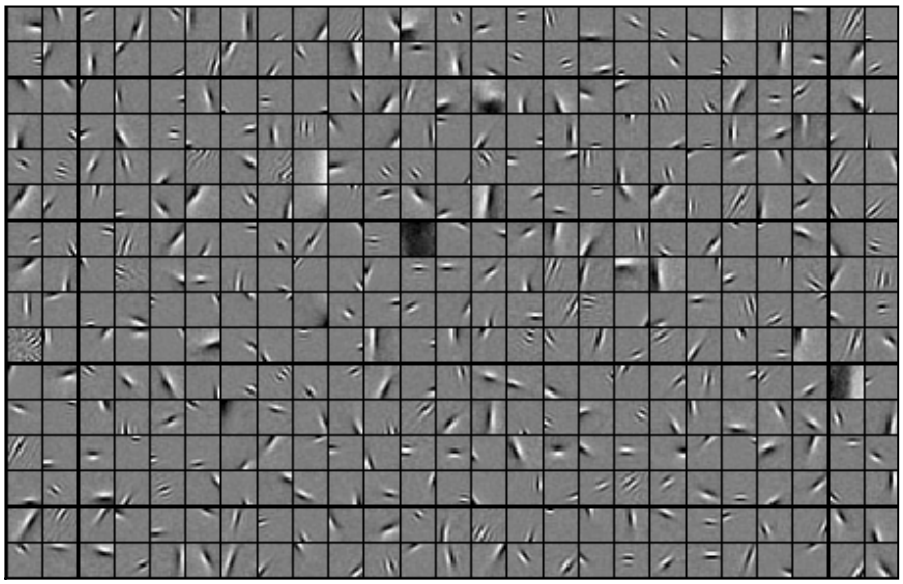


Figure 1.16: Basis functions learned from a collection of natural images in the sparse coding model.

pronounced statistical dependencies, thus violating the independence assumption of the sparse coding model with factorial prior over the coefficients. These statistical dependencies are informative and should be modeled.

I propose in Chapter 2 a model that attempts to capture the dependencies among the basis function coefficients by including a pairwise coupling term in the prior over the coefficients' activity states. When adapted to the statistics of natural images, the coupling terms converge to a solution involving a combination of facilitatory and inhibitory interactions among neighboring basis functions. These learned interactions may offer an explanation for the function of horizontal connections in V1 in terms of a prior over natural images. Part of this Chapter appeared in [Garrigues and Olshausen, 2007].

I propose in Chapter 3 a class of sparse priors called the Laplacian Scale Mixture. In this model the coefficients are distributed as continuous mixtures of Laplacian distributions, where the multiplier represents the inverse scale of the Laplacian distribution. The statistical dependencies among the coefficients are modeled via dependencies among the multiplier variables as in the Gaussian scale mixture model. I show that one can capture higher-order dependencies in natural images and leverage the efficient algorithms developed for Basis Pursuit Denoising to derive improved inference algorithms.

Up to now we have discussed the application of sparse approximation to signal and image processing, but it turns out that there is a formal relation between the sparse approximation problem and the problem of sparse linear regression in statistics. It is desired for the sake of interpretable results to have a vector of regressors that is sparse, such that the relevant features in the data are identified. A common solution is to solve the ℓ_1-regularized least-square problem that is referred to as Lasso in the statistics community [Tibshirani, 1996]. Note that it has the same cost-function as Basis Pursuit Denoising, but in most cases the least-square is overdetermined in the

regression setting. I propose in Chapter 4 RecLasso, an algorithm to solve the Lasso with online (sequential) observations. I introduce an optimization problem that allows us to compute an homotopy from the current solution to the solution after observing a new data point. I compare RecLasso to Lars [Efron *et al.*, 2004] and Coordinate Descent [Friedman *et al.*, 2007], and present an application to compressive sensing with sequential observations. The approach can easily be extended to compute an homotopy from the current solution to the solution that corresponds to removing a data point, which leads to an efficient algorithm for leave-one-out cross-validation. I also propose an algorithm to automatically update the regularization parameter after observing a new data point. Part of this Chapter appeared in [Garrigues and El Ghaoui, 2008].

Chapter 2

Learning horizontal connections

2.1 Introduction

We propose in this Chapter a linear generative model of image patches as in (1.7) that is such that the prior over the coefficients is not factorial. We introduce as in [Olshausen and Millman, 2000] a binary latent variable or "spin" for each coefficient. If the spin is equal to -1, then the corresponding coefficient is zero with probability 1. If the spin is equal to 1, then the corresponding coefficient has a Gaussian distribution. The spin variables therefore control which basis functions are being used to represent an image patch. We model these binary variables with a Boltzmann-Gibbs distribution, whose coupling weights control the dependencies among the coefficients.

Our model is motivated in part by the architecture of the visual cortex, namely the extensive network of horizontal connections among neurons in V1 [Fitzpatrick, 1996]. It has been hypothesized that they facilitate contour integration [Ben-Shahar and Zucker, 2004] and are involved in computing border ownership [Zhaoping, 2005]. In both of these models the connections are set *a priori* based on geometrical properties of the receptive fields. We propose here to learn the connection weights in an

unsupervised fashion. We hope with our model to gain insight into the the computations performed by this extensive collateral system and compare our findings to known physiological properties of these horizontal connections. Furthermore, a recent trend in neuroscience is to model networks of neurons using Ising models, and it has been shown to predict remarkably well the statistics of groups of neurons in the retina [Schneidman *et al.*, 2006]. Our model gives a prediction for what is expected if one fits an Ising model to future multi-unit recordings in V1. We also propose in Section 2.A another functional model for the horizontal connections in V1. We show that one can use such a network to decrease the number of connections in a linear dynamical system, which might explain how a neuron with a large receptive field computes its response.

2.2 A non-factorial sparse coding model

We begin with the following generative model, as described previously (1.7)

$$x = \Phi s + \nu = \sum_{i=1}^{m} s_i \varphi_i + \nu,$$

where $\Phi = [\varphi_1 \dots \varphi_m] \in \mathbb{R}^{n \times m}$ is an overcomplete transform or basis set, and the columns φ_i are its basis functions. $\nu \sim \mathcal{N}(0, \epsilon^2 I_n)$ is small Gaussian noise. Each coefficient s_i can be decomposed as follows

$$s_i = \frac{h_i + 1}{2} u_i,$$

where u_i is a Gaussian random variable and h_i is a binary random variable whose values are ± 1. The distribution of the coefficients is thus a special case of Gaussian Scale Mixture (GSM) composed of two discrete states of the multiplier variable. We

model the multiplier h with an Ising model, i.e. $h \in \{-1, 1\}^m$ has a Boltzmann-Gibbs distribution

$$p(h) = \frac{1}{Z} e^{\frac{1}{2} h^T W h + b^T h},$$

where Z is the normalization constant. If the spin h_i is down ($h_i = -1$), then $s_i = 0$ and the basis function φ_i is silent. If the spin h_i is up ($h_i = 1$), then the basis function is active and the analog value of the coefficient s_i is drawn from a Gaussian distribution with $u_i \sim \mathcal{N}(0, \sigma_i^2)$. The prior on s can thus be described as a "hard-sparse" prior as it is a mixture of a point mass at zero and a Gaussian.

The corresponding graphical model is shown in Figure 2.1. It is a chain graph since it contains both undirected and directed edges. It bears similarities to [Hinton *et al.*, 2005], which however does not have the intermediate layer s and is not a sparse coding model. To sample from this generative model, one first obtains a sample h from the Ising model, then samples coefficients s according to $p(s \mid h)$, and then x according to $p(x \mid s) \sim \mathcal{N}(\Phi s, \epsilon^2 I_n)$.

The parameters of the model to be learned from data are $\theta = (\Phi, (\sigma_i^2)_{i=1..m}, W, b)$. This model does not make any assumption about which linear code Φ should be used, and about which units should exhibit dependencies. The matrix W of the interaction weights in the Ising model describes these dependencies. $W_{ij} > 0$ favors positive correlations and thus corresponds to an excitatory connection, whereas $W_{ij} < 0$ corresponds to an inhibitory connection. A local magnetic field $b_i < 0$ favors the spin h_i to be down, which in turn makes the basis function φ_i mostly silent.

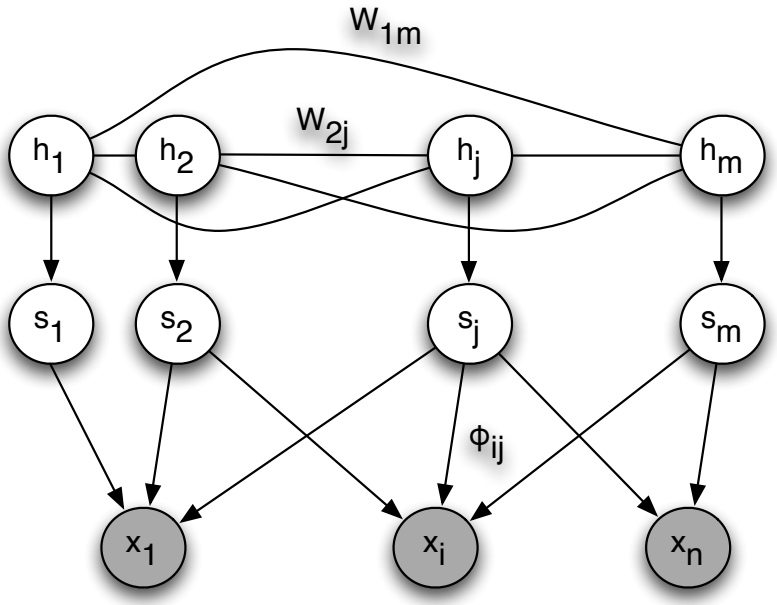


Figure 2.1: Proposed graphical model

2.3 Inference and learning

2.3.1 Coefficient estimation

We describe here how to infer the representation s of an image patch x in our model. To do so, we first compute the maximum a posteriori (MAP) multiplier h (see Section 2.3.2). Indeed, a GSM model reduces to a linear-Gaussian model conditioned on the multiplier s, and therefore the estimation of s is easy once h is known.

Given $h = \hat{h}$, let $\Gamma = \{i \ : \ \hat{h}_i = 1\}$ be the set of active basis functions. We know that $\forall i \notin \Gamma, \ s_i = 0$. Hence, we have $x = \Phi_\Gamma s_\Gamma + \nu$, where $s_\Gamma = (s_i)_{i\in\Gamma}$ and $\Phi_\Gamma = [(\varphi_i)_{i\in\Gamma}]$. The model reduces thus to linear-Gaussian, where $s_\Gamma \sim \mathcal{N}(0, H = \mathrm{diag}((\sigma_i^2)_{i\in\Gamma}))$. We have $s_\Gamma \mid x, \hat{h} \sim \mathcal{N}(\mu, K)$, where $K = (\epsilon^{-2}\Phi_\Gamma\Phi_\Gamma^T + H^{-1})^{-1}$ and $\mu = \epsilon^{-2}K\Phi_\Gamma^T x$. Hence, conditioned on x and $\hat{h}$, the Bayes Least-Square (BLS) and maximum a posteriori (MAP) estimators of s_Γ are the same and given by μ.

2.3.2 Multiplier estimation

The MAP estimate of h given x is given by $\hat{h} = \arg\max_h p(h \mid x)$. Given h, x has a Gaussian distribution $\mathcal{N}(0, \Sigma)$, where

$$\Sigma = \epsilon^2 I_n + \sum_{i \,:\, h_i=1} \sigma_i^2 \varphi_i \varphi_i^T.$$

Using Bayes' rule, we can write $p(h \mid x) \propto p(x \mid h)p(h) \propto e^{-E_x(h)}$, where

$$E_x(h) = \frac{1}{2} x^T \Sigma^{-1} x + \frac{1}{2} \log\det\Sigma - \frac{1}{2} h^T W h - b^T h.$$

We can thus compute the MAP estimate using Gibbs sampling and simulated annealing. In the Gibbs sampling procedure, the probability that node i changes its value from h_i to $\bar{h_i}$ given x, all the other nodes $h_{\neg i}$ and at temperature T is given by

$$p(h_i \to \bar{h_i} | h_{\neg i}, x) = \left(1 + \exp\left(-\frac{\Delta E_x}{T}\right)\right)^{-1},$$

where $\Delta E_x = E_x(h_i, h_{\neg i}) - E_x(\bar{h_i}, h_{\neg i})$. Note that computing E_x requires the inverse and the determinant of Σ, which is expensive. Let $\bar{\Sigma}$ and Σ be the covariance matrices corresponding to the proposed state $(\bar{h_i}, h_{\neg i})$ and current state $(h_i, h_{\neg i})$ respectively. They differ only by a rank 1 matrix, i.e. $\bar{\Sigma} = \Sigma + \alpha \varphi_i \varphi_i^T$, where $\alpha = \frac{1}{2}(\bar{h_i} - h_i)\sigma_i^2$. Therefore, to compute ΔE_x we can take advantage of the Sherman-Morrison formula

$$\bar{\Sigma}^{-1} = \Sigma^{-1} - \alpha \Sigma^{-1} \varphi_i (1 + \alpha \varphi_i^T \Sigma^{-1} \varphi_i)^{-1} \varphi_i^T \Sigma^{-1} \tag{2.1}$$

and of a similar formula for the $\log\det$ term

$$\log\det\bar{\Sigma} = \log\det\Sigma + \log\left(1 + \alpha \varphi_i^T \Sigma^{-1} \varphi_i\right). \tag{2.2}$$

Using (2.1) and (2.2) ΔE_x can be written as

$$\Delta E_x = \frac{1}{2}\frac{\alpha(x^T\Sigma^{-1}\varphi_i)^2}{1+\alpha\varphi_i^T\Sigma^{-1}\varphi_i} - \frac{1}{2}\log\left(1+\alpha\varphi_i^T\Sigma^{-1}\varphi_i\right) + (\bar{h_i}-h_i)\left(\sum_{j\neq i} W_{ij}h_j + b_i\right).$$

The transition probabilities can thus be computed efficiently, and if a new state is accepted we update Σ and Σ^{-1} using (2.1).

2.3.3 Model estimation

Given a dataset $\mathcal{D} = \{x^{(1)}, \ldots, x^{(N)}\}$ of image patches, we want to learn the parameters $\theta = (\Phi, (\sigma_i^2)_{i=1..m}, W, b)$ that offer the best explanation of the data. Let $p^*(x) = \frac{1}{N}\sum_{i=1}^{N}\delta(x - x^{(i)})$ be the empirical distribution. Since in our model the variables s and h are latent, we use a variational expectation maximization algorithm [Jordan *et al.*, 1999] to optimize θ, which amounts to maximizing a lower bound on the log-likelihood derived using Jensen's inequality

$$\log p(x \mid \theta) \;\geq\; \sum_h \int_s q(s,h \mid x)\log\frac{p(x,s,h \mid \theta)}{q(s,h \mid x)}ds \triangleq \mathcal{L}(\theta, q),$$

where $q(s,h \mid x)$ is a probability distribution. We restrict ourselves to the family of point mass distributions $\mathcal{Q} = \{q(s,h \mid x) = \delta(s-\hat{s})\delta(h-\hat{h})\}$, and with this choice the lower bound on the log-likelihood of $\mathcal{D}$ can be written as

$$\begin{aligned}\mathcal{L}(\theta, q) &= \mathbb{E}_{p^*}[\log p(x, \hat{s}, \hat{h} \mid \theta)] \qquad (2.3)\\ &= \underbrace{\mathbb{E}_{p^*}[\log p(x \mid \hat{s}, \Phi)]}_{\mathcal{L}_\Phi} + \underbrace{\mathbb{E}_{p^*}[\log p(\hat{s} \mid \hat{h}, (\sigma_i^2)_{i=1..m})]}_{\mathcal{L}_\sigma} + \underbrace{\mathbb{E}_{p^*}[\log p(\hat{h} \mid W, b)]}_{\mathcal{L}_{W,b}}.\end{aligned}$$

We perform coordinate ascent in the objective function $\mathcal{L}(\theta, q)$.

2.3.3.1 Maximization with respect to q

We want to solve $\max_{q \in \mathcal{Q}} \mathcal{L}(\theta, q)$, which amounts to finding $\arg\max_{s,h} \log p(x, s, h)$ for every $x \in \mathcal{D}$. This is computationally expensive since h is discrete. Hence, we introduce two phases in the algorithm.

In the first phase, we infer the coefficients in the usual sparse coding model where the prior over s is factorial, i.e. $p(s) = \prod_i p(s_i) \propto \prod_i \exp\{-\lambda S(s_i)\}$. In this setting, we have

$$\hat{s} = \arg\max_s p(x|s) \prod_i e^{-\lambda S(s_i)} = \arg\min_s \frac{1}{2\epsilon^2} \|x - \Phi s\|_2^2 + \lambda \sum_i S(s_i). \tag{2.4}$$

With $S(s_i) = |s_i|$, (2.4) is known as basis pursuit denoising (BPDN) whose solution has been shown to be such that many coefficient of $\hat{s}$ are exactly zero [Chen *et al.*, 1999]. This allows us to recover the sparsity pattern $\hat{h}$, where $\hat{h}_i = 2\mathbf{1}[\hat{s}_i \neq 0] - 1 \ \forall i$. BPDN can be solved efficiently using a competitive algorithm [Rozell *et al.*, 2008]. Another possible choice is $S(s_i) = \mathbf{1}[s_i \neq 0]$ ($p(s_i)$ is not a proper prior though), where (2.4) is combinatorial and can be solved approximately using orthogonal matching pursuits (OMP) [Tropp, 2004].

After several iterations of coordinate ascent and convergence of θ using the above approximation, we enter the second phase of the algorithm and refine θ by using the GSM inference described in Section 2.3.1 where $\hat{h} = \arg\max p(h|x)$ and $\hat{s} = \mathbb{E}[s \mid \hat{h}, x]$.

2.3.3.2 Maximization with respect to θ

We want to solve $\max_\theta \mathcal{L}(\theta, q)$. Our choice of variational posterior allowed us to write the objective function as the sum of the three terms $\mathcal{L}_\Phi$, $\mathcal{L}_\sigma$ and $\mathcal{L}_{W,b}$ (2.3), and hence to decouple the variables Φ, $(\sigma_i^2)_{i=1..m}$ and (W, b) of our optimization problem.

Maximization of $\mathcal{L}_\Phi$. Note that $\mathcal{L}_\Phi$ is the same objective function as in the standard sparse coding problem when the coefficients a are fixed. Let $\{\hat{s}^{(i)}, \hat{h}^{(i)}\}$ be the coefficients and multipliers corresponding to $x^{(i)}$. We have

$$\mathcal{L}_\Phi = -\frac{1}{2\epsilon^2}\sum_{i=1}^{N} \|x^{(i)} - \Phi\hat{s}^{(i)}\|_2^2 - \frac{Nn}{2}\log 2\pi\epsilon^2.$$

We add the constraint that $\|\varphi_i\|_2 \leq 1$ to avoid the spurious solution where the norm of the basis functions grows and the coefficients tend to 0. We solve this ℓ_2 constrained least-square problem using the Lagrange dual as in [Lee *et al.*, 2007].

Maximization of $\mathcal{L}_\sigma$. The problem of estimating σ_i^2 is a standard variance estimation problem for a 0-mean Gaussian random variable, where we only consider the samples $\hat{s}_i$ such that the spin $\hat{h}_i$ is equal to 1, i.e.

$$\sigma_i^2 = \frac{1}{\text{card}\{k \ : \ \hat{h}_i^{(k)} = 1\}} \sum_{k \ : \ \hat{h}_i^{(k)}=1} (\hat{s}_i^{(k)})^2.$$

Maximization of $\mathcal{L}_{W,b}$. This problem is tantamount to estimating the parameters of a fully visible Boltzmann machine [Ackley *et al.*, 1985] which is a convex optimization problem. We do gradient ascent in $\mathcal{L}_{W,b}$, where the gradients are given by

$$\begin{cases} \frac{\partial \mathcal{L}_{W,b}}{\partial W_{ij}} = -\mathbb{E}_{p^*}[h_i h_j] + \mathbb{E}_p[h_i h_j] \\ \frac{\partial \mathcal{L}_{W,b}}{\partial b_i} = -\mathbb{E}_{p^*}[h_i] + \mathbb{E}_p[h_i] \end{cases}$$

We use Gibbs sampling to obtain estimates of $\mathbb{E}_p[h_i h_j]$ and $\mathbb{E}_p[h_i]$.

Note that since computing the parameters $(\hat{s}, \hat{h})$ of the variational posterior in phase 1 only depends on Φ, we first perform several steps of coordinate ascent in (Φ, q) until Φ has converged, which is the same as in the usual sparse coding algorithm. We then maximize $\mathcal{L}_\sigma$ and $L_{W,b}$, and after that we enter the second phase of the algorithm.

2.4 Recovery of the model parameters

Although the learning algorithm relies on a method where the family of variational posteriors $q(s, h \mid x)$ is quite limited, we argue here that if data $\mathcal{D} = \{x^{(1)}, \ldots, x^{(N)}\}$ is being sampled according to parameters θ_0 that obey certain conditions that we describe now, then our proposed learning algorithm is able to recover θ_0 with good accuracy using phase 1 only.

Let η be the coherence parameter of the basis set which equals the maximum absolute inner product between two distinct basis functions. It has been shown that given a signal that is a sparse linear combination of p basis functions, BP and OMP will identify the optimal basis functions and their coefficients provided that $p < \frac{1}{2}(\eta^{-1} + 1)$, and the sparsest representation of the signal is unique [Tropp, 2004]. Similar results can be derived when noise is present ($\epsilon > 0$) [Tropp, 2006], but we restrict ourselves to the noiseless case for simplicity. Let $\|h\|_\uparrow$ be the number of spins that are up. We require (W_0, b_0) to be such that $Pr\left(\|h\|_\uparrow < \frac{1}{2}(\eta^{-1} + 1)\right) \approx 1$, which can be enforced by imposing strong negative biases. A data point $x^{(i)} \in \mathcal{D}$ thus has a high probability of yielding a unique sparse representation in the basis set Φ. Provided that we have a good estimate of Φ we can recover its sparse representation using OMP or BP, and therefore identify $h^{(i)}$ that was used to originally sample $x^{(i)}$. That is we recover with high probability all the samples from the Ising model used to generate $\mathcal{D}$, which allows us to recover (W_0, b_0).

We provide for illustration a simple example of model recovery where $n = 7$ and $m = 8$. Let $(e_1, \ldots, e_7)$ be an orthonormal basis in $\mathbb{R}^7$. We let $\Phi_0 = [e_1, \ldots e_7, \frac{1}{\sqrt{7}} \sum_i e_i]$. We fix the biases b_0 at -1.2 such that the model is sufficiently sparse as shown by the histogram of $\|h\|_\uparrow$ in Figure 2.2, and the weights W_0 are sampled according to a Gaussian distribution. The variance parameters σ_0 are fixed to 1. We then generate synthetic data by sampling 100000 data from this model using θ_0. We then estimate

θ from this synthetic data using the variational method described in Section 2.3 using OMP and phase 1 only. We found that the basis functions are recovered exactly (not shown), and that the parameters of the Ising model are recovered with high accuracy as shown in Figure 2.2.

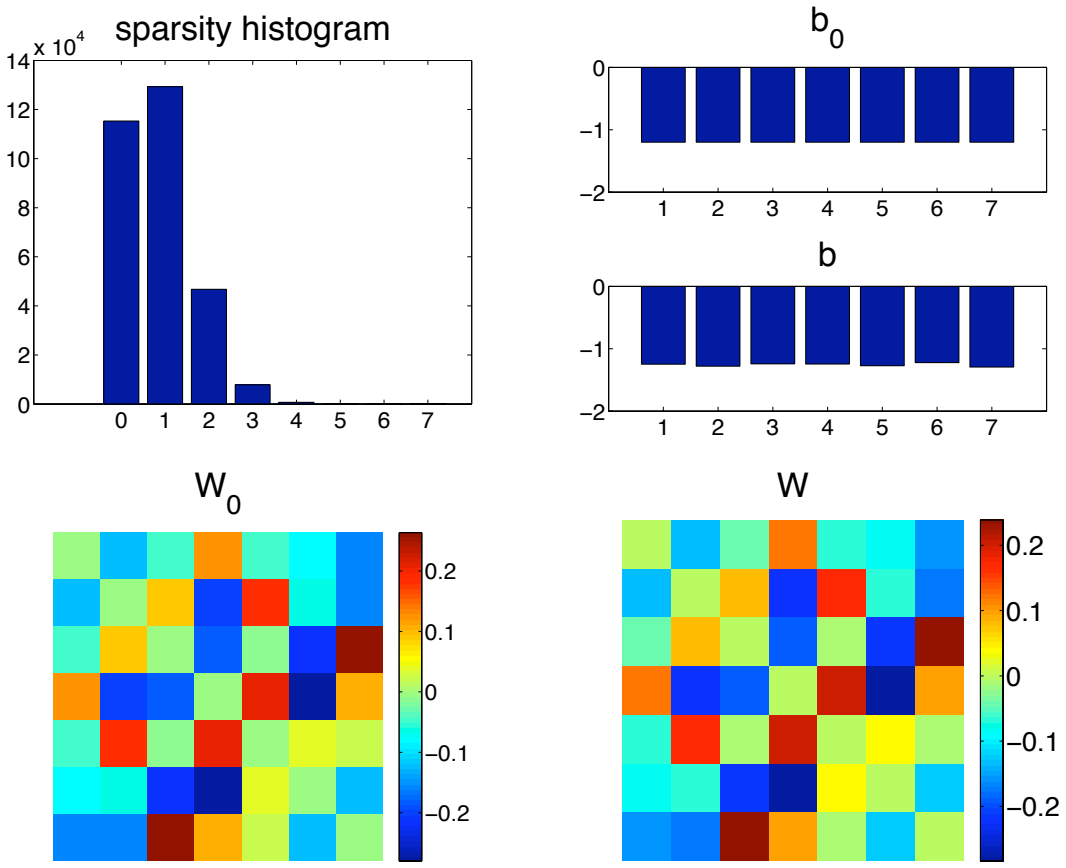


Figure 2.2: Recovery of the model. The histogram of $\|s\|_\uparrow$ is such that the model is sparse. The parameters (W, b) learned from synthetic data are close to the parameters (W_0, b_0) from which this data was generated.

2.5 Results for natural images

We build our training set by randomly selecting 16×16 image patches from a standard set of 10 512×512 whitened images as in [Olshausen and Field, 1996]. It has been shown that change of luminance or contrast have little influence on the structure of natural scenes [Wang *et al.*, 2005]. As our goal is to uncover this structure, we subtract from each patch its own mean and divide it by its standard deviation such that our dataset is contrast normalized (we do not consider the patches whose variance is below a small threshold). We fix the number of basis functions to 256. In the second phase of the algorithm we only update (W, b), and we have found that the basis functions do not change dramatically after the first phase.

Figure 2.3 shows the learned parameters Φ, σ and b. The basis functions resemble Gabor filters at a variety of orientations, positions and scales. We show the weights W in Figure 2.5 and Figure 2.6 according to the spatial properties (position, orientation, length) of the basis functions that are linked together by them. Each basis function is denoted by a bar that indicates its position, orientation, and length within the 16×16 patch.

We observe that the connections are mainly local and connect basis functions at a variety of orientations. The histogram of the weights (see Figure 2.7) shows a long positive tail corresponding to a bias toward facilitatory connections. We can see in Figure 2.4 that the 10 most "positive" pairs have similar orientations, whereas the majority of the 10 most "negative" pairs have dissimilar orientations. We compute for a basis function the average number of basis functions sharing with it a weight larger than 0.01 as a function of their orientation difference in four bins, which we refer to as the "orientation profile" in Figure 2.7. The error bars are a standard deviation. The resulting orientation profile is consistent with what has been observed in physiological experiments [Malach *et al.*, 1993; Bosking *et al.*, 1997].

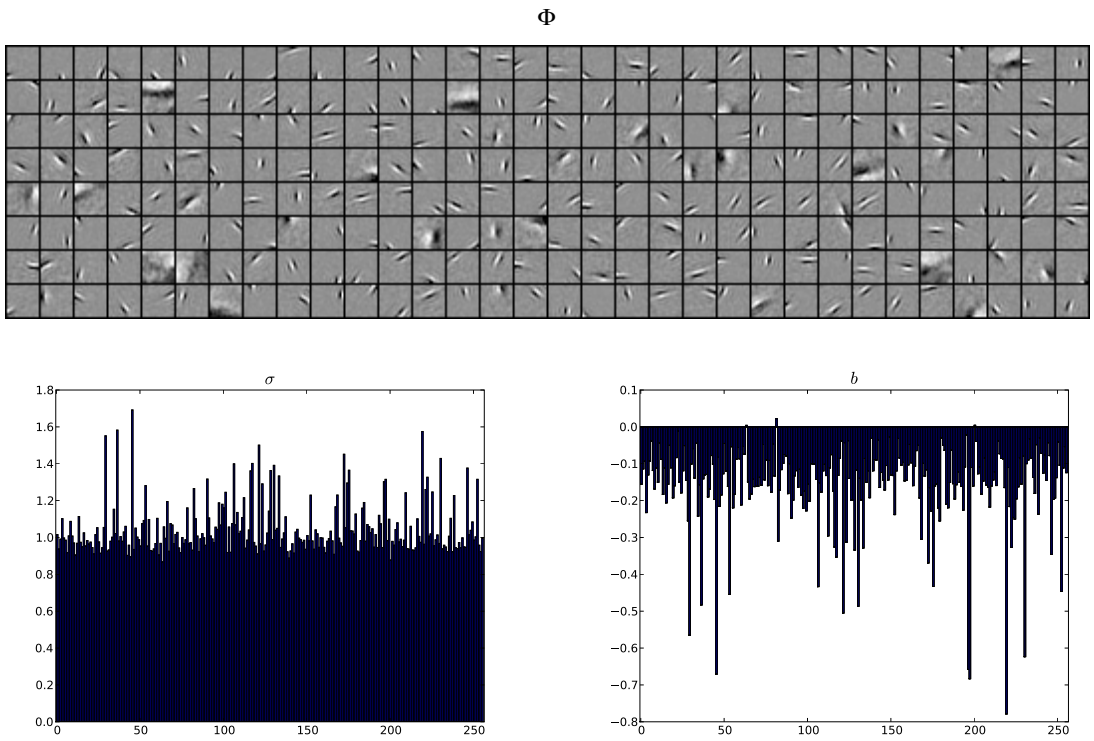
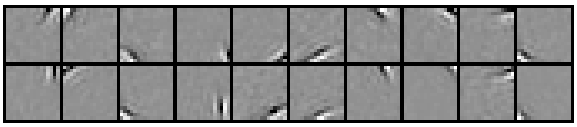
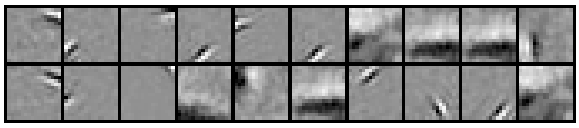


Figure 2.3: **Top** Set of basis functions Φ learned on natural images. **Bottom left** Learned variances $(\sigma_i^2)_{i=1..m}$. **Bottom right** Learned biases b in the Ising model.

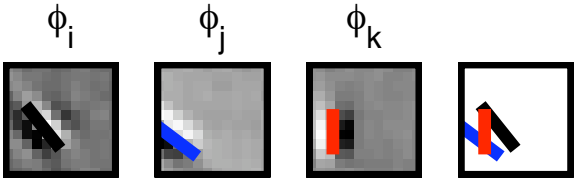


(a) 10 most positive weights

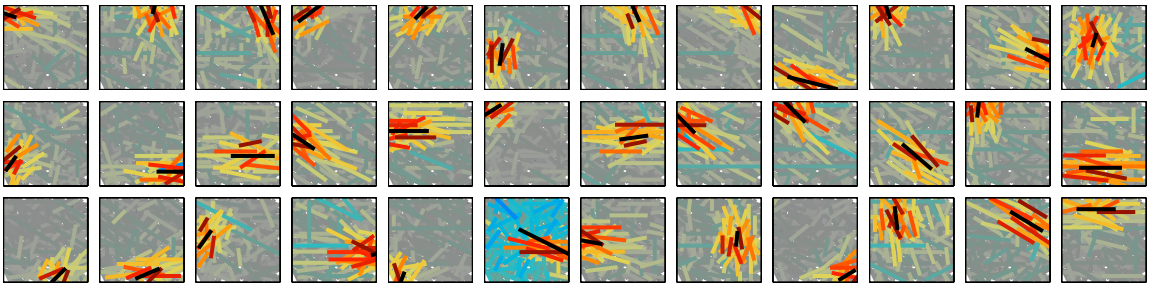


(b) 10 most negative weights

Figure 2.4: (a) (resp. (b)) shows the basis function pairs (columnwise) that share the strongest positive (resp. negative) weights ordered from left to right.



(a) Weights visualization



(b) Association fields

Figure 2.5: Each subplot in (b) shows the association field for a basis function φ_i whose position and orientation are denoted by the black bar. The horizontal connections $(W_{ij})_{j\neq i}$ are displayed by a set of colored bars whose orientation and position denote those of the basis functions φ_j to which they correspond, and the color denotes the connection strength, where red is positive and blue is negative (see (a), $W_{ij} < 0$ and $W_{ik} > 0$). We show a random selection of 36 association fields.

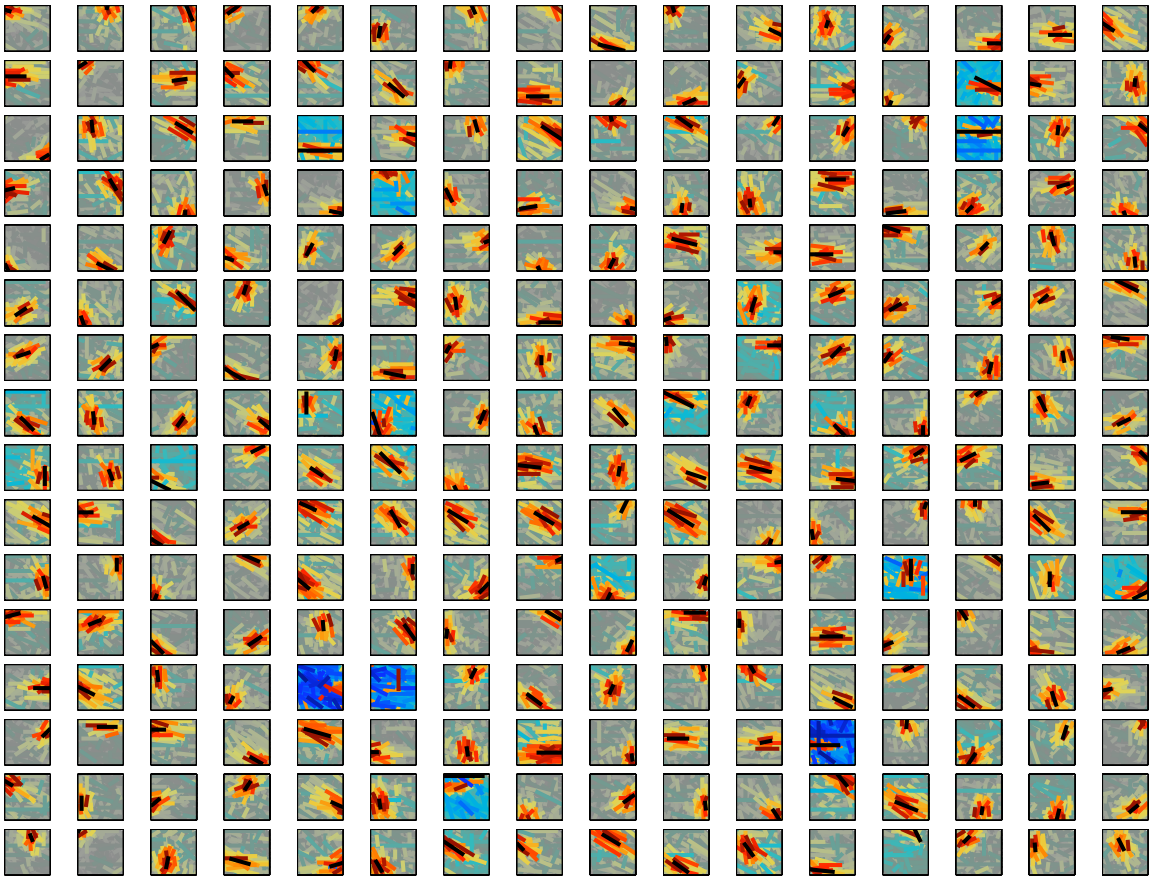


Figure 2.6: Entire set of learned association fields.

We also show in Figure 2.7 the tradeoff between the signal to noise ratio (SNR) of an image patch x and its reconstruction $\Phi\hat{s}$, and the ℓ_0 norm of the representation $\|\hat{s}\|_0$. We consider $\hat{s}$ inferred using both the Laplacian prior and our proposed prior. We vary λ (see Equation (2.4)) and ϵ respectively, and average over 1000 patches to obtain the two tradeoff curves. We see that at similar SNR the representations inferred by our model are more sparse by about a factor of 2, which bodes well for compression. We have also compared our prior for tasks such as denoising and filling-in, and have found its performance to be similar to the factorial Laplacian prior even though it does not exploit the dependencies of the code. One possible explanation is that the greater sparsity of our inferred representations makes them less robust to noise.

To assess how well the pairwise model captures the actual joint distribution of coefficients we compare the model's probability vs. the actual probability of occurrence for a select group of 10 basis function coefficients sharing strong weights. Let Λ denote the indices for this group. Given a collection of image patches that we sparsify using (2.4), we obtain a number of spins $(\hat{h}_i)_{i\in\Lambda}$ from which we can estimate the empirical distribution p_{emp}, the Boltzmann-Gibbs distribution p_{Ising} consistent with first and second order correlations, and the factorial distribution p_{fact} (i.e. no horizontal connections) consistent with first order correlations. We can see in Figure 2.8 that the Ising model produces better estimates of the empirical distribution, and results in better coding efficiency since $KL(p_{emp}||p_{Ising}) = .02$ whereas $KL(p_{emp}||p_{fact}) = .1$. As a comparison, we also selected a group of 10 basis functions randomly and estimated the Boltzmann-Gibbs and factorial distributions. In this case, we have $KL(p_{emp}||p_{Ising}) = .01$ whereas $KL(p_{emp}||p_{fact}) = .04$. As expected, the reduction in coding efficiency with the Ising model is not as great as in the case where the basis functions coefficients have greater statistical dependencies. We can visualize the estimated distributions in Figure 2.9.

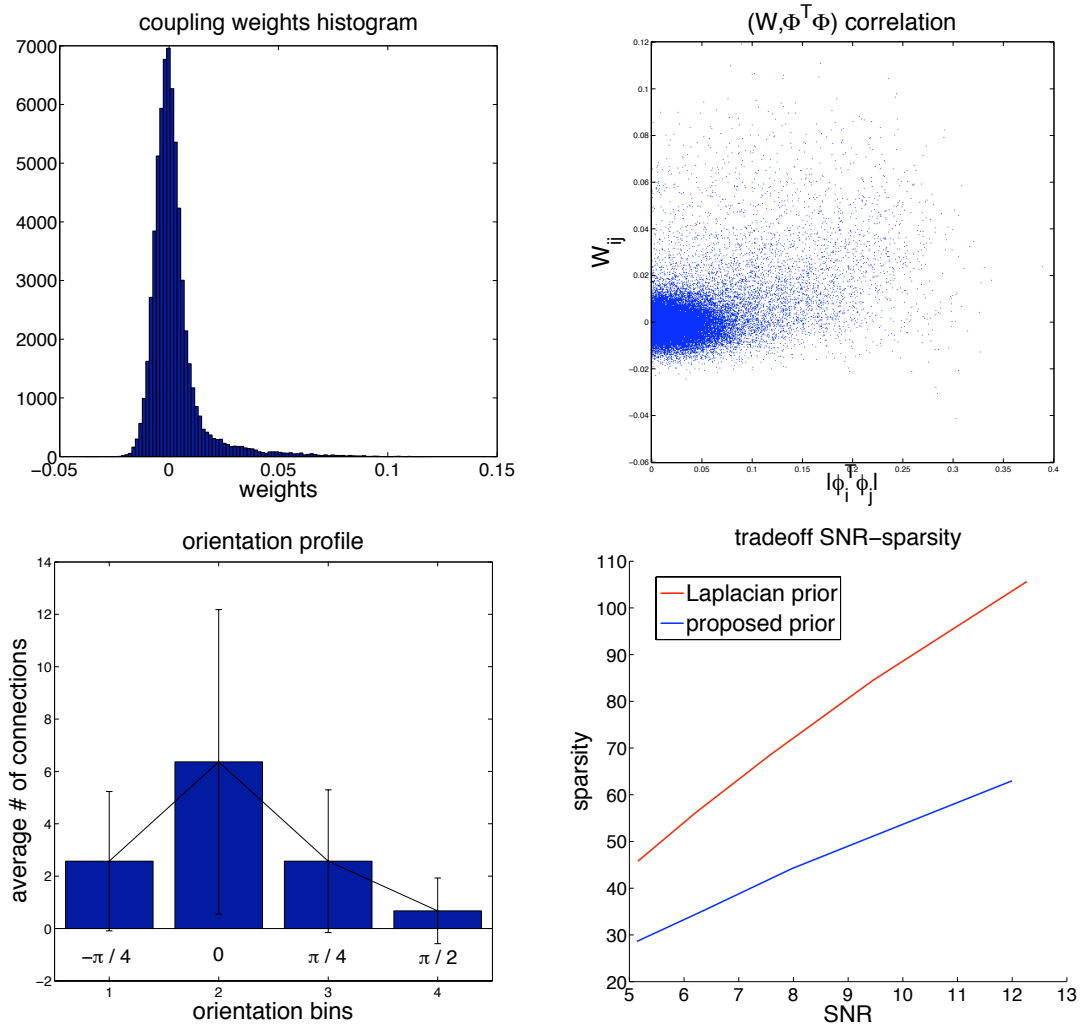


Figure 2.7: **Upper left** Histogram of the coupling weights W_{ij}. The distribution is skewed towards positive weights. **Upper right** Correlation between the coupling weights and the Gramm matrix of the basis functions. **Bottom left** Orientation profile: distribution of the angular difference between a basis function and the basis functions that it is coupled to with weights greater than $.01$. The error bars represent one standard deviation. **Bottom right** Comparison of the tradeoff curve SNR - ℓ_0 norm for the inferred coefficients using a factorial Laplacian prior and our proposed prior.

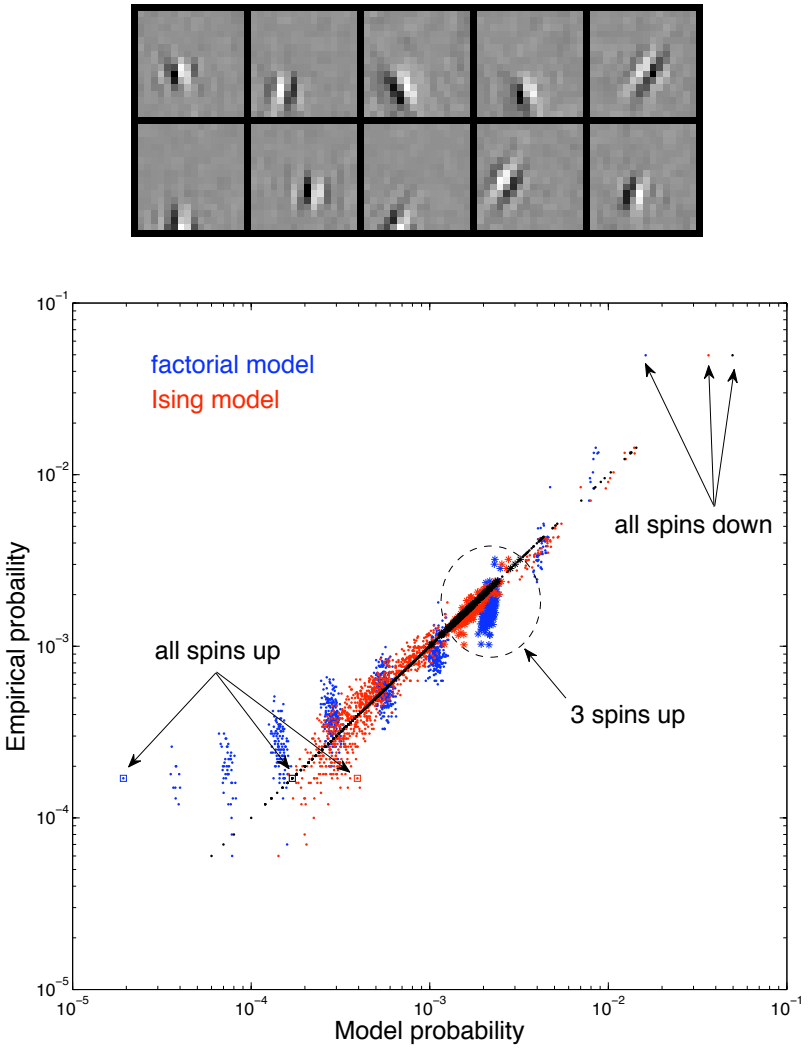


Figure 2.8: Model validation for a group of 10 basis functions sharing strong weights (top). The empirical probabilities of the 2^{10} patterns of activation are plotted against the probabilities predicted by the Ising model (red), the factorial model (blue), and their own values (black). These patterns having exactly three spins up are circled. The prediction of the Ising model is noticably better than that of the factorial model.

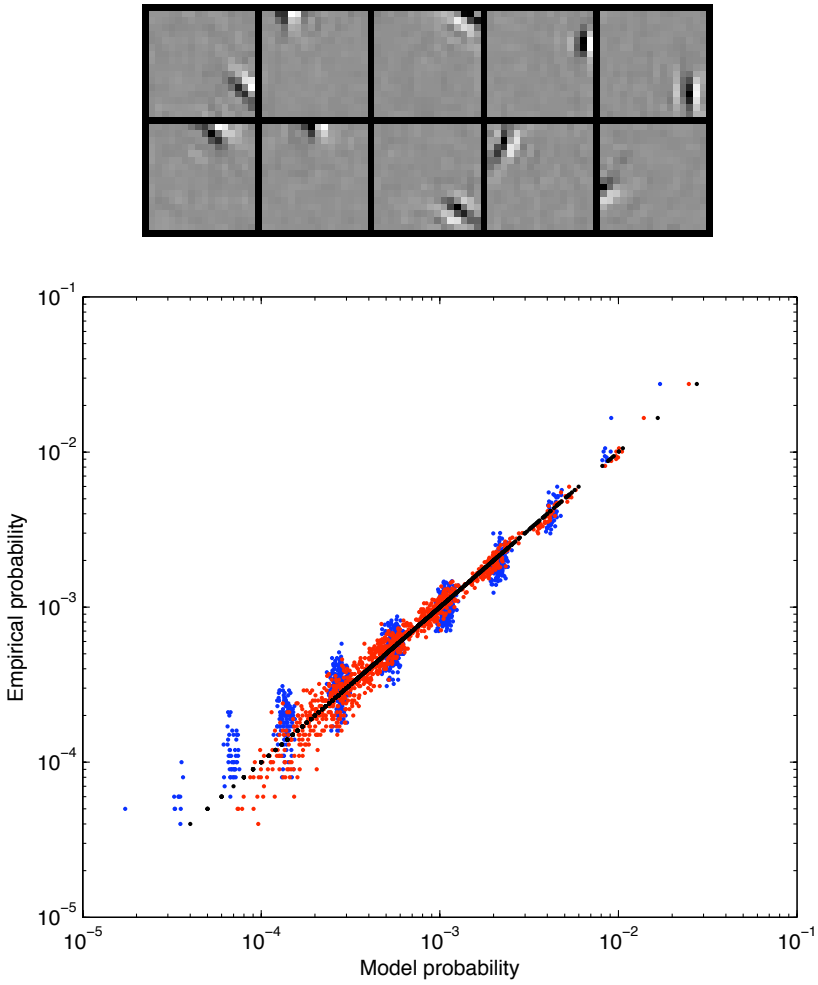


Figure 2.9: Model validation for a group of 10 basis functions selected at random (top). The empirical probabilities of the 2^{10} patterns of activation are plotted against the probabilities predicted by the Ising model (red), the factorial model (blue), and their own values (black).

2.6 Discussion

In this Chapter, we proposed a new sparse coding model where we include pairwise coupling terms among the coefficients to capture their dependencies. During inference, the hidden binary units now attempt to encourage or discourage other units to be active, similar to the likely role of horizontal connections in the cortex. We derived a new learning algorithm to adapt the parameters of the model given a data set of natural images, and we were able to discover the dependencies among the basis functions coefficients. We showed that the learned connection weights are consistent with physiological data. Furthermore, the representations inferred in our model have greater sparsity than when they are inferred using the Laplacian prior as in the standard sparse coding model. Note however that we have not found evidence that these horizontal connections facilitate contour integration, as they do not primarily connect colinear basis functions. Previous models in the literature simply assume these weights according to prior intuitions about the function of horizontal connections [Ben-Shahar and Zucker, 2004; Zhaoping, 2005]. These models are largely inspired by a line-drawing view of the visual world, and our goal here was to derive a model of pairwise interactions in a principled manner informed from the statistics of the visual world. The results are not altogether intuitive in that simple contour grouping did not emerge from the pairwise statistics. We offer several possible explanations. First, it is probable that the pairwise constraint is too weak to capture the complex image structure that includes contours, textures and boundary intersections. In addition, our model assumes that images are the *linear* superposition of features, which is not an efficient way of representing how surfaces occlude each other in natural images and give rise to contours.

Appendix 2.A Feedforward computation with horizontal connections

The receptive fields of simple cells in primary visual cortex (V1) resemble a collection of oriented Gabor filters at a variety of scales and orientations. A standard hypothesis is that these cells construct their receptive fields by taking a weighted sum of inputs from the Lateral Geniculate Nucleus (LGN) through a feedforward computation. However, the simple cells whose receptive fields' frequencies are low integrate information over a large portion of the visual field, and in a purely feedforward model they should receive information from a large spatial extent of neurons in the LGN, which is not physiologically feasible due to neuronal branching constraints. We propose here a model where a dynamical system with recurrent computations using horizontal connections allows the cells with large receptive fields to reduce the region in the LGN they need to receive inputs from while achieving the same computation as the purely feedforward system. Note that a similar strategy might be used by the retina to compute center-surround receptive fields by spreading inhibition laterally using horizontal cells. For a review of dynamical systems in neural computation, see [Eliasmith and Anderson, 2003].

To investigate this problem, we make the following abstractions. We saw in Section 1.3.2 that learning a set of filters from the statistics of natural images so as to maximize the sparsity of the outputs, an approach known as Independent Component Analysis, results in filters that resemble the receptive fields of simple cell neurons in V1 as shown in Figure 2.10. Let $x \in \mathbb{R}^n$ denote the image pixels, and $a \in \mathbb{R}^n$ denote the outputs in the mapping $a = Tx$, where T is the transform learned using ICA. The first layer x denotes the activity of our model LGN neurons, and the second layer a denotes the activity of our model V1 simple cells. In the feedforward model, the i^{th}

model unit computes its response via

$$a_i = \sum_{j=1}^{n} T_{ij} x_j.$$

Hence, the i^{th} units receives inputs from the units in the first layer $\{x_j \ : \ T_{ij} \neq 0\}$. We can see in Figure 2.10 that the low frequency units receive inputs from a large portion of the pixels in the 16×16 patch.

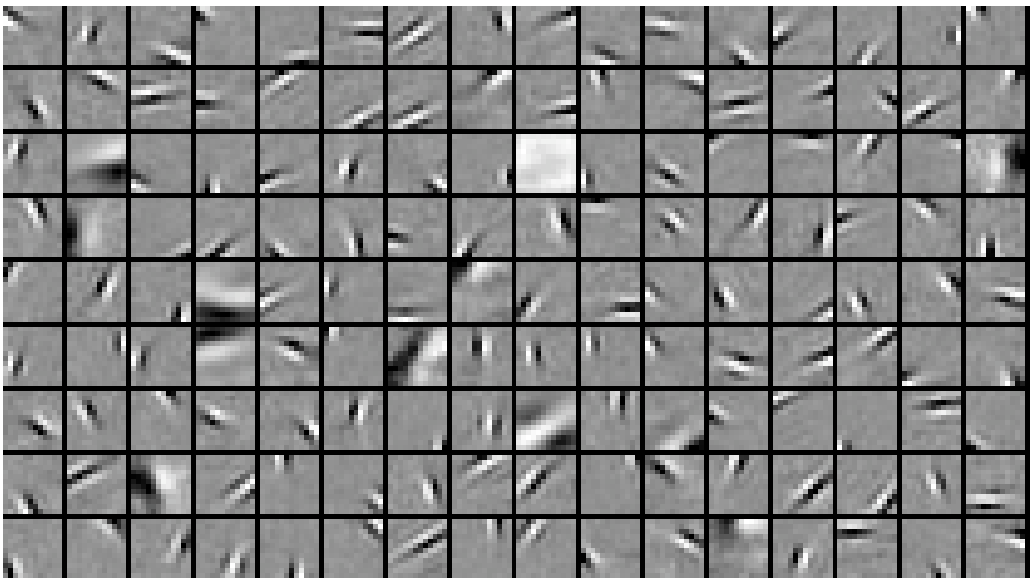


Figure 2.10: Model receptive fields learned using ICA (n = 144).

2.A.1 Dynamical system formulation

We propose the following dynamical system

$$\tau \frac{da}{dt} + a = Ma + Wx,$$

where the dynamics are illustrated in Figure 2.11. M is a matrix that defines recurrent connections among the model simple cells, and W is the matrix of feedforward weights. At the equilibrium, we have

$$a = Ma + Wx \;\Rightarrow\; a = (I - M)^{-1}Wx.$$

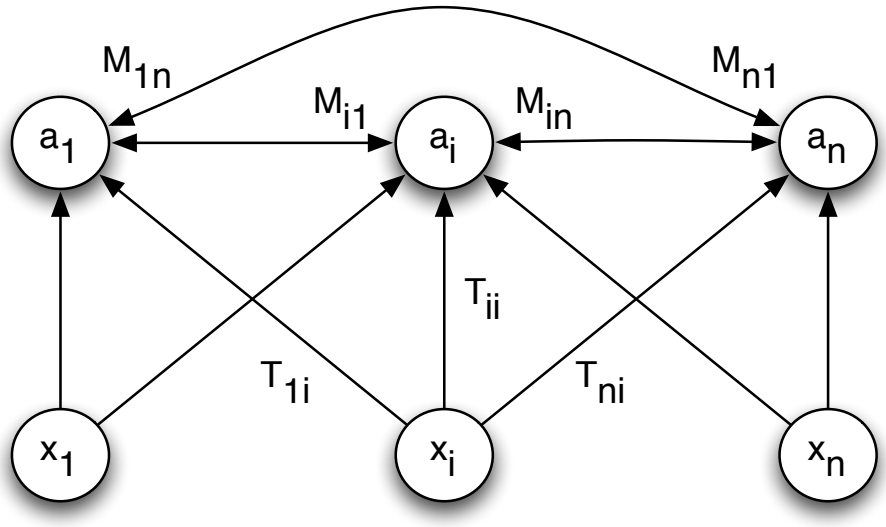


Figure 2.11: The recurrent system.

Since we want the system to compute $a = Tx$, we have the constraint $(I - M)^{-1}W = T$. Let $\|T\|_0$ be the ℓ_0 norm of T, i.e. the number of nonzero elements in T. $\|T\|_0$ corresponds to the number of connections between x and a. The question is thus whether we can find M and W such that $(I - M)^{-1}W = T$, and the number of connections in the recurrent system is smaller than a purely feedforward computation, i.e. $\|M\|_0 + \|W\|_0 < \|T\|_0$. Note that a purely feedforward system corresponds to $M = 0$ and $W = T$.

2.A.2 Optimization problem formulation

Ideally, we would like to solve

$$\min_{W,M} \|M\|_0 + \|W\|_0 \ : \ (I - M)^{-1}W = T.$$

Unfortunately, this is a combinatorial optimization problem and cannot be solved as is. A standard relaxation as seen in Section 1.4.1 is to replace the ℓ_0 norm by the ℓ_1 norm ($\|z\|_1 = \sum_i |z_i|$). Furthermore, we reformulate the constraint in the following way

$$(I - M)^{-1}W = T \ \Rightarrow \ W = T - MT \ \Rightarrow \ W + MT = T.$$

We also add the constraint that $M_{ii} = 0$ for all i, i.e. there are no self-connections. This avoids the trivial solution $W = 0$ and $M = I$, in which case $(I - M)^{-1}$ is not even defined. The optimization problem becomes

$$\min_{W,M} \|M\|_1 + \|W\|_1 \quad \text{subject to} \quad \begin{cases} W + MT = T \\ M_{ii} = 0 \quad \forall i \end{cases}$$

This can be formulated as a convex linear program, and can be thus solved efficiently using a standard interior point method.

2.A.3 Results

To compare the number of connections in the purely feedforward and the optimal recurrent system, we compute the ℓ_0 norm reduction defined by

$$100 \times \frac{\|T\|_0 - (\|M\|_0 + \|W\|_0)}{\|T\|_0}. \tag{2.5}$$

As the smallest elements of these matrices are not exactly 0 due to the interior-point method that we use, we compute the ℓ_0 norm by setting to zero the elements that are smaller than some threshold. Figure 2.12 shows the ℓ_0 norm reduction as a function of the threshold. We can see that by introducing recurrent connections we are indeed capable of reducing the overall number of connections. Note that for a conservative threshold choice of 0.01, we still have a reduction of about 15%.

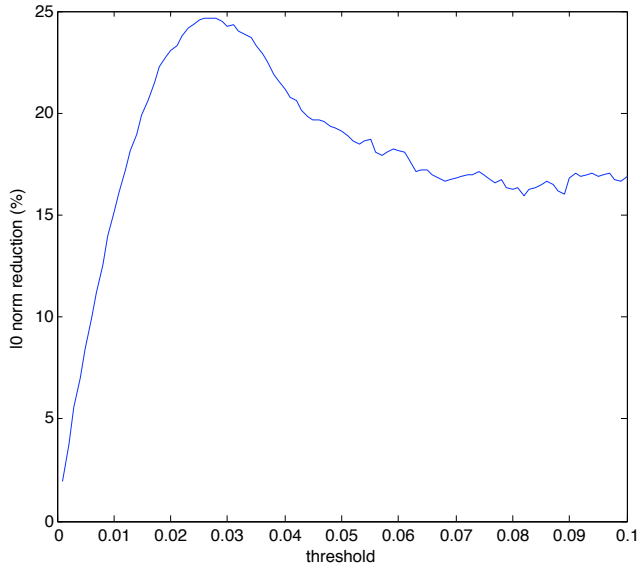


Figure 2.12: Reduction in number of connections. The reduction is not monotonic as the number of connections in the feedforward system also decreases with threshold.

It is also interesting to visualize the feedforward weights W shown in Figure 2.13 in the (M, W) recurrent system. Our primary goal was to decrease the area of the region in the x layer over which neurons in V1 receive their inputs, and it is interesting to see that it is indeed the case, even though our optimization problem does not add this constraint explicitly. Several model neurons with large receptive fields receive almost no input from the x layer in the recurrent system, and construct their receptive fields mostly by means of horizontal connections. To make this claim more quantitative,

we compute the area reduction as a function of the receptive field area as shown in Figure 2.14. We observe that the larger the receptive field, the bigger the reduction, up to 100% for 4 neurons.

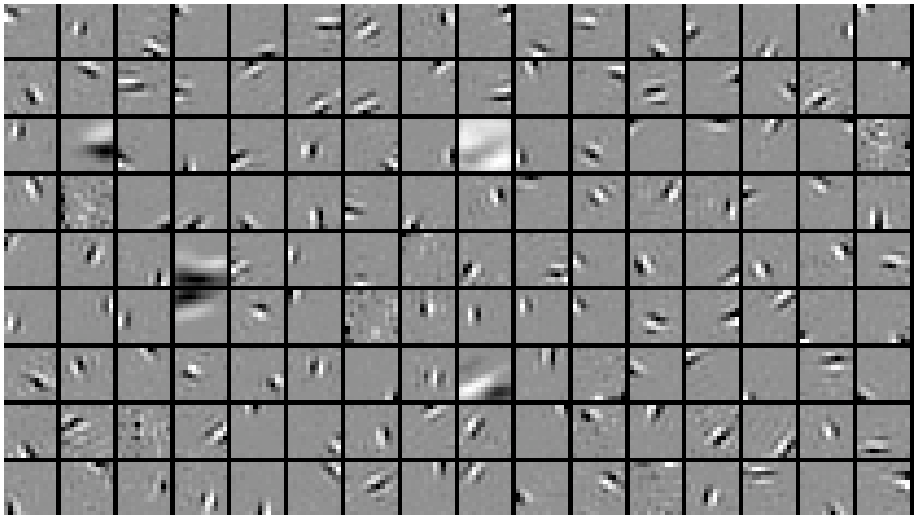


Figure 2.13: Feedforward weights in the optimal recurrent system.

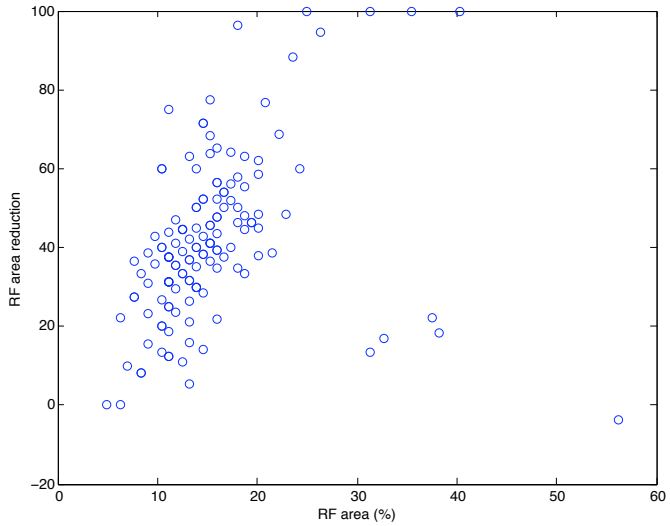


Figure 2.14: Branching reduction as a function of receptive field area. We define the area over which the i^{th} model simple cell receives inputs in the feedforward system in percent of the image patch size by $100 \times \frac{\mathsf{card}(\{j\ :\ T_{ij} \neq 0\})}{12 \times 12}$, and in the recurrent system by $100 \times \frac{\mathsf{card}(\{j\ :\ W_{ij} \neq 0\}.)}{12 \times 12}$. Each dot corresponds to a model simple cell.

Chapter 3

Laplacian Scale Mixture

3.1 Introduction

We saw in Section 1.4.3 that a popular method to compute the sparse representation of a signal consists of solving the ℓ_1-regularized least-square problem

$$\frac{1}{2\sigma^2}\|x - \Phi s\|_2^2 + \lambda\|s\|_1,$$

an approach known as Basis Pursuit Denoising (BPDN). The cost function of BPDN is convex, and many efficient algorithms have been recently developed to solve this problem [Efron *et al.*, 2004][Daubechies *et al.*, 2004][Rozell *et al.*, 2007][Friedman *et al.*, 2007][Figueiredo *et al.*, 2007][Lee *et al.*, 2007]. The ℓ_1 penalty leads to sparse solutions, which is a desirable property to achieve model selection or data compression, or for obtaining interpretable results.

We saw in Section 1.4.3 that BPDN corresponds to MAP inference in a generative model where the coefficients are independent and have Laplacian priors

$$p(s_i) = \frac{\lambda}{2}e^{-\lambda|s_i|}.$$

Hence, the signal model assumed by BPDN is linear, generative, and the basis function coefficients are independent. We saw in Chapter 1 and 2 that this is *not* a good model for real-world signals such as natural images. We were able to capture dependencies using a Gaussian Scale Mixture (GSM) prior on the coefficient where the dependencies are captured using the multiplier variables. Note that a similar prior was recently proposed in [Cevher *et al.*, 2008]. We also saw in Section 2.5 that BPDN is not always able to identify the sparsest representation for a given reconstruction error.

It has been proposed in block-ℓ_1 methods [Yuan and Lin, 2006] to account for dependencies among the coefficients by dividing them into subspaces such that dependencies within the subspaces are allowed, but not across the subspaces. This approach can produce blocking artifacts and has recently been generalized to overlapping subspaces in [Jacob *et al.*, 2009][Jenatton *et al.*, 2009]. Another approach is to only allow certain configurations of active coefficients [Baraniuk *et al.*, 2008].

We propose in this Chapter a new class of prior on the basis function coefficients that makes it possible to model their statistical dependencies, whose inferred representations are more sparse than those obtained with the factorial Laplacian prior, and for which we have efficient inference algorithms. Our approach consists of introducing for each coefficient a hyperprior on the inverse scale parameter λ_i of the Laplacian distribution. The coefficient prior is thus a mixture of Laplacian distributions which we denote "Laplacian Scale Mixture" (LSM), which is an analogy to the Gaussian scale mixture (GSM) [Wainwright *et al.*, 2001b]. The prior has higher kurtosis, and the representations are therefore more sparse. A natural way to model the statistical dependencies among the coefficients is to use a non-factorial hyperprior, i.e.

$$p(\lambda_1, \ldots, \lambda_m) \neq \prod_{i=1}^{m} p(\lambda_i).$$

In analysis-based models, such non-factorial hyperpriors on the scale parameters of

the Gaussian [Wainwright *et al.*, 2001b] or generalized Gaussian [Karklin and Lewicki, 2005] have been shown to capture higher-order dependencies in natural images. We extend this approach to a synthesis-based model. An advantage of having a mixture of Laplacian distribution as opposed to a mixture of Gaussian distribution is also computational, as we can exploit the sparsity of the solutions obtained using a the Laplacian prior and leverage efficient inference algorithms developed for BPDN. Indeed we show that inference can be solved efficiently via a sequence of reweighted ℓ_1-regularized least-square problems. Note that such optimization algorithms have been proposed for sparse coding in [Candès *et al.*, 2008][Wipf and Nagarajan, 2008]. Here we propose a Bayesian interpretation of [Candès *et al.*, 2008].

We saw in Section 1.1.3 that a natural way to compare signal models is to look at their performance in ill-posed inverse problems. We focus in this Chapter on the problem of compressive sensing recovery. Compressive sensing is an alternative to Shannon/Nyquist sampling for acquisition of sparse signals where inner-products of the signal with random vectors are observed, and the signal is subsequently recovered with a sparsity-seeking optimization algorithm such as BPDN. In the case where the signals of interest have structure beyond sparsity such as dependencies among the coefficients, it has been shown that better recovery can be achieved using an algorithm that exploits this structure [Baraniuk *et al.*, 2008][Cevher *et al.*, 2008][Cevher *et al.*, 2009]. We show that our model is also able to achieve significant improvements with signals having higher-order structure beyond sparsity.

The outline of this Chapter is as follows. We define the Laplacian scale mixture in Section 3.2, and describe the inference algorithms in the resulting sparse coding models with an LSM prior on the coefficients in Section 3.3. We present an example of a factorial LSM model in Section 3.4, and of a non-factorial LSM model in Section 3.5 that is particularly well suited to signals having the "group sparsity" property.

3.2 The Laplacian Scale Mixture distribution

3.2.1 Definition

A random variable s_i is a Laplacian scale mixture if it can be written

$$s_i = \lambda_i^{-1} u_i,$$

where u_i has a Laplacian distribution with scale 1, i.e. $p(u_i) = \frac{1}{2} e^{-|u_i|}$, and λ_i is a positive random variable with probability $p(\lambda_i)$. We also suppose that λ_i and u_i are independent. Conditioned on the parameter λ_i, the coefficient s_i has a Laplacian distribution with inverse scale λ_i, i.e.

$$p(s_i|\lambda_i) = \frac{\lambda_i}{2} e^{-\lambda_i |s_i|}.$$

We show in Figure 3.1 examples of Laplacian distributions with various inverse scales.

The distribution over s_i is therefore a continuous mixture of Laplacian distributions with different inverse scales, and it can be computed by integrating out λ_i

$$p(s_i) = \int_0^\infty p(s_i|\ \lambda_i) p(\lambda_i) d\lambda_i = \int_0^\infty \frac{\lambda_i}{2} e^{-\lambda_i |s_i|} p(\lambda_i) d\lambda_i.$$

Note that for most choices of $p(\lambda_i)$ we do not have an analytical expression for $p(s_i)$. We denote such a distribution a Laplacian Scale Mixture (LSM) as an analogy to the Gaussian Scale Mixture [Wainwright *et al.*, 2001b], and we similarly refer to λ_i as the multiplier variable.

The family of LSM defines distributions that have heavy tails. To see that, we compute the kurtosis of an LSM, and show that it is always greater than the kurtosis of the Laplacian distribution. We first note that u_i is a Laplacian distribution, and

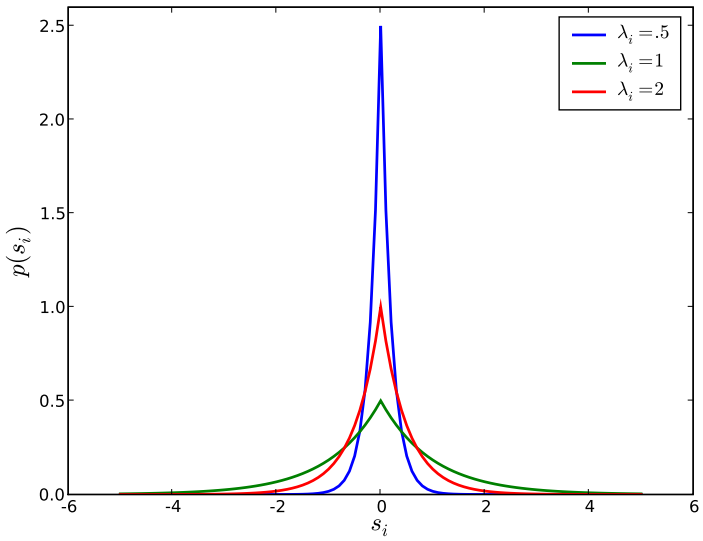


Figure 3.1: Laplacian distribution corresponding to three inverse scales λ.

its kurtosis is given by

$$\kappa(u_i) = \frac{\mathbb{E}[u_i^4]}{(\mathbb{E}[u_i^2])^2} = 6.$$

Note that the mean of an LSM is given by

$$\mathbb{E}[s_i] = \mathbb{E}[\lambda_i^{-1} u_i] = \mathbb{E}[\lambda_i^{-1}]\mathbb{E}[u_i] = 0.$$

The kurtosis of s_i is thus

$$\kappa(s_i) = \frac{\mathbb{E}[s_i^4]}{(\mathbb{E}[s_i^2])^2} = \frac{\mathbb{E}[(\lambda_i^{-1} u_i)^4]}{(\mathbb{E}[(\lambda_i^{-1} u_i)^2])^2} = \frac{\mathbb{E}[(\lambda_i^{-1})^4]}{(\mathbb{E}[(\lambda_i^{-1})^2])^2} \kappa(u_i).$$

Using the convexity of $f(x) = x^2$ it is easy to see that $\mathbb{E}[X^2] \geq (\mathbb{E}[X])^2$ for any random variable X. By applying this inequality to λ_i^{-2} we conclude that

$$\kappa(s_i) \geq \kappa(u_i) = 6.$$

Hence an LSM random variable typically has heavier tails than a Laplacian random variable.

3.2.2 Examples

The Laplacian distribution is part of the LSM family. Indeed, if the multiplier takes some value with probability 1, i.e. $p(\lambda_i) = \delta(\lambda_i - \tilde{\lambda}_i)$, then we have

$$p(s_i) = \frac{\tilde{\lambda}_i}{2} e^{-\tilde{\lambda}_i |s_i|}.$$

If the multiplier takes on discrete values $\{\tilde{\lambda}_i^j\}_{j=1..J}$ with probabilities $\{\pi_j\}_{j=1..J}$, the resulting distribution is a discrete mixture of Laplacian distributions

$$p(s_i) = \sum_{i=1}^{J} \pi_j \frac{\tilde{\lambda}_i^j}{2} e^{-\tilde{\lambda}_i^j |s_i|}.$$

In [Levin and Weiss, 2007] a discrete mixture of two Laplacian distributions is used to model the distribution of derivative filter outputs as applied to natural images.

Suppose that the multiplier has a Gamma distribution, i.e.

$$p(\lambda_i) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_i^{\alpha-1} e^{-\beta\lambda_i},$$

where α is the shape parameter and β is the inverse scale parameter. If $\alpha \in \mathbb{N}$, we have $\Gamma(\alpha) = (\alpha - 1)!$. A few examples of Gamma distributions are shown in Figure 3.2. Note that a particular case of Gamma distribution is the exponential distribution when $\alpha = 1$. With this particular choice of a prior on the multiplier, we can compute the probability distribution of s_i analytically

$$p(s_i) = \frac{\alpha\beta^\alpha}{2(\beta + |s_i|)^{\alpha+1}}. \tag{3.1}$$

We can see in Figure 3.2 that the distribution on the coefficients has heavier tails than the Laplacian distribution.

3.3 Resulting sparse coding models

3.3.1 Generative model with Laplacian scale mixture prior

We propose as we did in Chapter 2 the linear generative model

$$x = \Phi s + \nu = \sum_{i=1}^{m} s_i \varphi_i + \nu,$$

where $\Phi = [\varphi_1, \ldots, \varphi_m] \in \mathbb{R}^{n \times m}$ is an overcomplete transform or basis set, and the columns φ_i are its basis functions. $\nu \sim \mathcal{N}(0, \sigma^2 I_n)$ is small Gaussian noise. In this model the coefficients are endowed with LSM distributions. The graphical model for an LSM sparse coding model is shown in Figure 3.3. The nodes λ_i are fully connected as in general we do not make any assumptions about $p(\lambda)$.

The standard sparse coding model corresponds to the particular case where the LSM prior is the Laplacian prior. We can create richer models that capture the statistical dependencies among the coefficients by means of non-factorial priors on the multipliers, i.e.

$$p(\lambda) \neq \prod_i p(\lambda_i).$$

We propose in Section 3.4 and 3.5 various choices on the multiplier distribution $p(\lambda)$, which lead to models having different properties.

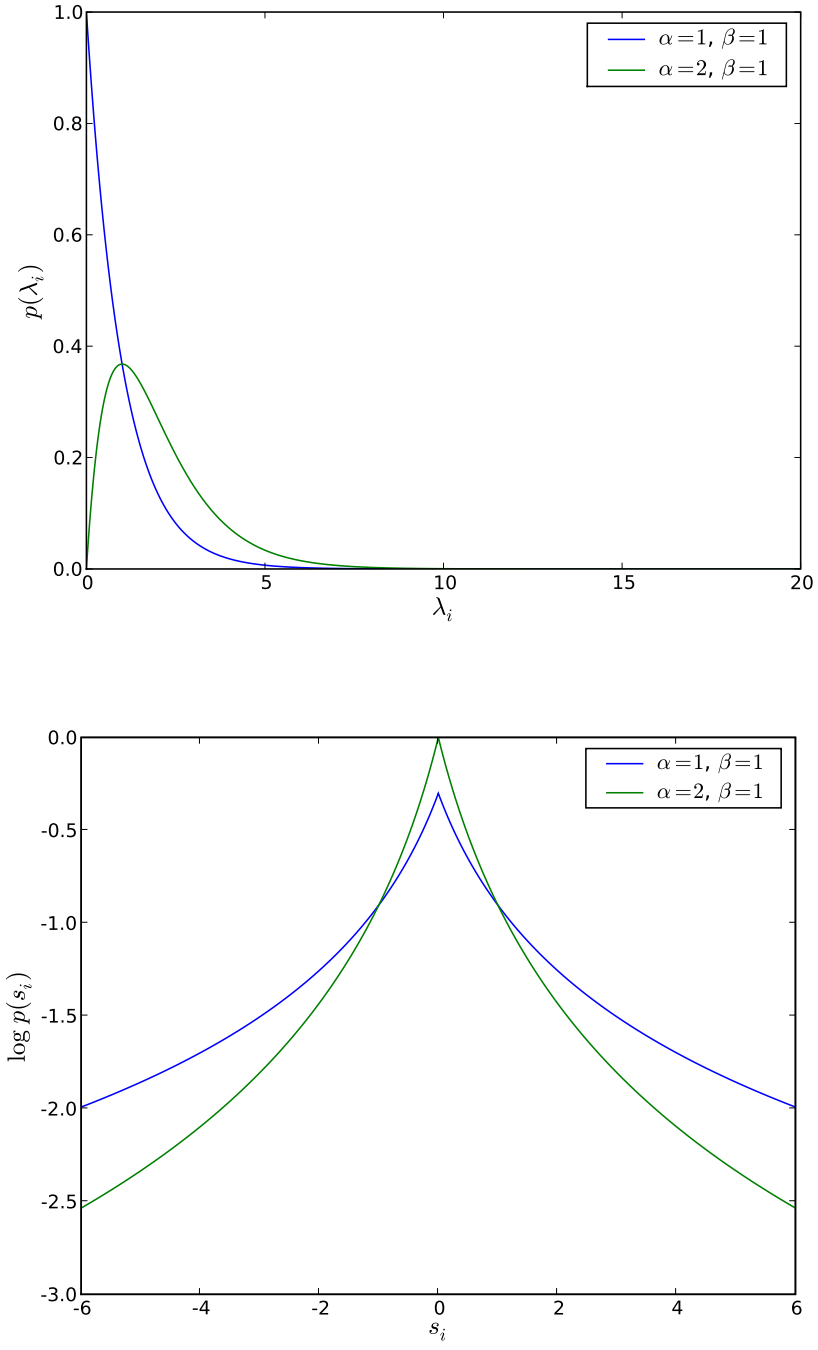


Figure 3.2: Distribution of the multipliers and coefficients in the LSM model with Gamma prior.

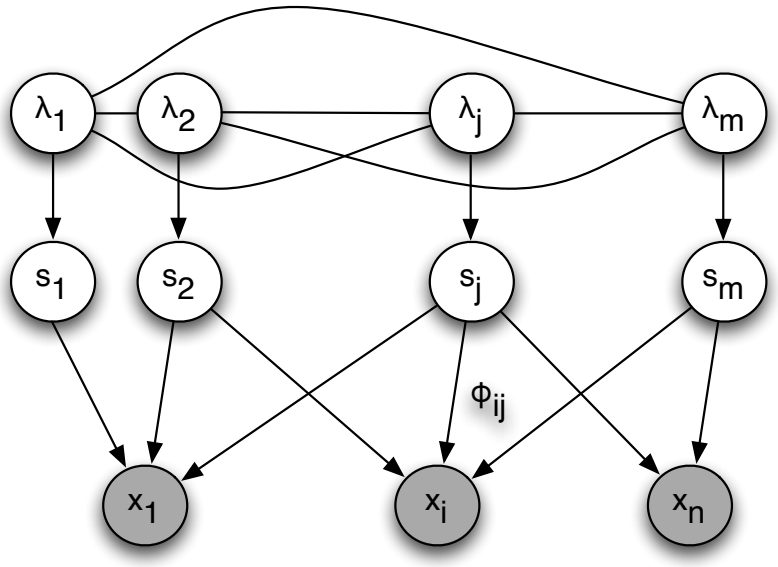


Figure 3.3: Graphical model representation of our proposed generative model with LSM prior.

3.3.2 Inference

Given a signal x, we wish to infer its sparse representation s in the dictionary Φ. We consider in this section the computation of the maximum a posteriori (MAP) estimate of the coefficients s given the input signal x. Using Bayes' rule we have $p(s \mid x) \propto p(x \mid s)p(s)$, and therefore the MAP estimate $\hat{s}$ is given by

$$\hat{s} = \arg\min_{s} \; -\log p(s \mid x) \tag{3.2}$$

$$= \arg\min_{s} \; -\log p(x \mid s) - \log p(s). \tag{3.3}$$

It is in general difficult to compute the MAP estimate with an LSM prior on s since we do not necessarily have an analytical expression for the log-likelihood $\log p(s)$.

However, we can compute the *complete* log-likelihood $\log p(s, \lambda)$ analytically

$$\begin{aligned} \log p(s, \lambda) &= \log p(s \mid \lambda) + \log p(\lambda) \\ &= -\lambda_i |s_i| + \log \frac{\lambda_i}{2} + \log p(\lambda). \end{aligned}$$

Hence, if we also observed the latent variable λ, we would have an objective function that can be maximized with respect to s. The standard approach in machine learning when confronted with such a problem is the Expectation-Maximization (EM) algorithm [Dempster *et al.*, 1977], and we derive in this Section an EM algorithm for the MAP estimation of the coefficients.

Let us first use Jensen's inequality and the concavity of the logarithm to write

$$\log p(s) \geq \int_\lambda q(\lambda) \log \frac{p(s, \lambda)}{q(\lambda)} d\lambda, \tag{3.4}$$

which is true for any probability distribution $q(\lambda)$. This gives an upper bound on the posterior likelihood

$$-\log p(s \mid x) \leq -\log p(x \mid s) - \int_\lambda q(\lambda) \log \frac{p(s, \lambda)}{q(\lambda)} d\lambda := \mathcal{L}(q, s) \tag{3.5}$$

Performing coordinate descent in the auxiliary function $\mathcal{L}(q, s)$ leads to the following updates that are usually called the E step and the M step.

$$\textbf{E Step} \qquad q^{(t+1)} = \arg\min_q \ \mathcal{L}(q, s^{(t)}) \tag{3.6}$$

$$\textbf{M Step} \qquad s^{(t+1)} = \arg\min_s \ \mathcal{L}(q^{(t+1)}, s) \tag{3.7}$$

Let $< \ . \ >_q$ denote the expectation with respect to $q(\lambda)$. We can write $\mathcal{L}(q, s)$ as

follows

$$\begin{aligned}\mathcal{L}(q,s) &= -\log p(x \mid s) + \langle -\log p(s \mid \lambda)\rangle_q + KL(q(\lambda)||p(\lambda)) \\ &= \frac{1}{2\sigma^2}\|x-\Phi s\|_2^2 + \frac{n}{2}\log 2\pi\sigma^2 + \sum_{i=1}^{m}\left(\langle\lambda_i\rangle_q |s_i| - \left\langle \log\frac{\lambda_i}{2}\right\rangle_q\right) + \dots \\ &\qquad \dots + KL(q(\lambda)||p(\lambda)),\end{aligned}$$

where we have used the conditional independence of the coefficients s given the multipliers λ, i.e. $p(s \mid \lambda) = \prod_{i=1}^{m} p(s_i \mid \lambda)$, and $KL(q(\lambda)||p(\lambda))$ represents the KL divergence between the distribution $q(\lambda)$ and $p(\lambda)$

$$KL(q(\lambda)||p(\lambda)) = \int_\lambda q(\lambda)\log\frac{q(\lambda)}{p(\lambda)}d\lambda.$$

Hence, the M Step (3.7) simplifies to

$$s^{(t+1)} = \arg\min_s \ \frac{1}{2\sigma^2}\|x-\Phi s\|_2^2 + \sum_{i=1}^{m}\langle\lambda_i\rangle_{q^{(t+1)}}|s_i|, \tag{3.8}$$

which is a least-square problem regularized by a weighted sum of the absolute values of the coefficients. It is a quadratic program very similar to BPDN, and we can use efficient algorithms that have been developed for BPDN in the M step.

We have equality in (3.4) if $q(\lambda) = p(\lambda \mid s)$. The inequality (3.5) is therefore tight for this particular choice of q, which implies that the E step reduces to $q^{(t+1)}(\lambda) = p(\lambda \mid s^{(t)})$. Note that in the M step we only need to the expectation of λ_i with respect to the maximizing distribution in the E step. Hence we only need to compute the sufficient statistics

$$\langle\lambda_i\rangle_{p(\lambda|s^{(t)})} = \int_\lambda \lambda_i p(\lambda \mid s^{(t)})d\lambda. \tag{3.9}$$

This explains why this step is usually referred to as the *expectation* step.

Note that the posterior of the multiplier given the coefficient $p(\lambda \mid s)$ might be hard to compute. We will see in Section 3.4.1 that it is tractable if the prior on λ is factorial and each λ_i has a Gamma distribution, as the Laplacian distribution and the Gamma distribution are conjugate.

3.3.3 Variational approximation

In the case where we cannot compute the sufficient statistics (3.9), we can use a variational approximation [Jordan *et al.*, 1999] whose principle is to restrict the family of distribution in the E step to a family of distribution $\mathcal{Q}$ that is simple enough such that we can compute the sufficient statistics. The variational E step is given by

$$\max_{q \in \mathcal{Q}} \ \mathcal{L}(q, s^{(t)}). \tag{3.10}$$

Note that in this case we no longer have equality in (3.4) for the maximizing distribution.

An example for $\mathcal{Q}$ is the family of point-mass distributions

$$\mathcal{Q} = \left\{ q(\lambda) = \delta(\lambda - \lambda^*),\ \lambda^* \in \mathbb{R}_+ \right\}.$$

With this choice, Jensen's inequality (3.4) has the simple form

$$\log p(s^{(t)}) \geq \log p(s^{(t)}, \tilde{\lambda}).$$

Let $q^{(t+1)}(\lambda) = \delta(\lambda - \lambda^{(t+1)})$ be the solution of (3.10). We have

$$\lambda^{(t+1)} = \arg\max_{\tilde{\lambda}} \ \log p(s^{(t)}, \tilde{\lambda}), \tag{3.11}$$

and the sufficient statistics are given by $\langle \lambda_i \rangle_{q^{(t+1)}} = \lambda_i^{(t+1)}$.

We can recast the "point-mass" variational approximation as simply computing the maximum a posteriori estimate of the latent variables s and λ. Using Bayes' rule the MAP estimate is the solution of

$$\begin{aligned}
\hat{s}, \hat{\lambda} &= \arg\max_{s,\lambda} \; p(s, \lambda \mid x) \\
&= \arg\max_{s,\lambda} \; p(x \mid s) p(s \mid \lambda) p(\lambda) \\
&= \arg\min_{s,\lambda} \; \frac{1}{2\sigma^2} \|x - \Phi s\|_2^2 + \sum_{i=1}^{m} (\lambda_i |s_i| - \log \lambda_i) - \log p(\lambda).
\end{aligned}$$

Let $E(s, \lambda)$ be the objective function that is minimized. Performing block-coordinate descent in E with respect to s and λ leads to the following algorithm

$$\textbf{Step 1} \qquad s^{(t+1)} = \arg\max_{s} \; E\left(s, \lambda^{(t)}\right) \tag{3.12}$$

$$\textbf{Step 2} \qquad \lambda^{(t+1)} = \arg\max_{\lambda} \; E\left(s^{(t+1)}, \lambda\right). \tag{3.13}$$

Suppose that the probability over λ is log-concave. In this case the objective function E is convex in s and in λ, but in general not in both variables. We are however guaranteed to decrease E by applying the block-coordinate descent in s and λ.
We can rewrite (3.12) as

$$s^{(t+1)} = \arg\min_{s} \; \frac{1}{2\sigma^2} \|x - \Phi s\|_2^2 + \sum_{i=1}^{m} \lambda_i^{(t)} |s_i|,$$

which is similar to (3.8). The second step is given by

$$\lambda^{(t+1)} = \arg\max_{\lambda} \log p(s^{(t)}, \lambda) \tag{3.14}$$

$$= \arg\min_{\lambda} \sum_{i=1}^{m} \lambda_i |s_i| - \log \lambda_i - \log p(\lambda), \tag{3.15}$$

which is the problem solved in (3.11). Hence these are in reverse order the updates proposed in the "point-mass" variational approximation. We typically initialize the multipliers with their expected value $\mathbb{E}[\lambda_i]$, and the first step is similar to inference in the standard sparse coding model. However, at the next iteration the multipliers modified according to (3.15) provide contextual feedback for the inference.

3.4 A factorial model

We propose in this Section a sparse coding model where the distribution of the multipliers is factorial, and each multiplier has a Gamma distribution with parameters α and β. The graphical model corresponding to this generative model is shown in Figure 3.4.

3.4.1 Conjugacy

The Gamma distribution and Laplacian distribution are *conjugate*, i.e. the posterior probability of λ_i given s_i is also a Gamma distribution when the prior over λ_i is a Gamma distribution and the conditional probability of s_i given λ_i is a Laplace

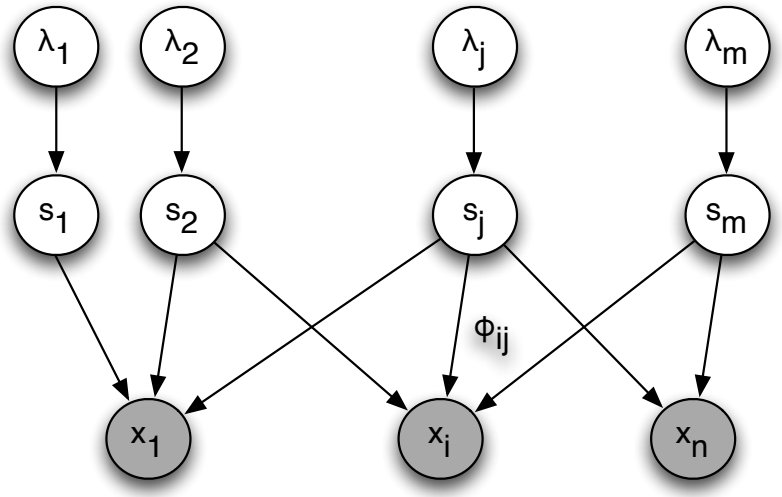


Figure 3.4: Graphical model representation of our proposed generative model where the multipliers distribution is factorial.

distribution with inverse scale λ_i. We have indeed

$$\begin{aligned} p(\lambda_i \mid s_i) &\propto p(s_i \mid \lambda_i) p(\lambda_i) \\ &\propto \lambda_i e^{-\lambda_i |s_i|} \lambda_i^{\alpha-1} e^{-\beta\lambda_i} \\ &\propto \lambda_i^{\alpha} e^{-(\beta+|s_i|)\lambda_i}. \end{aligned}$$

The posterior of λ_i given s_i is thus a Gamma distribution with parameters $\alpha+1$ and $\beta+|s_i|$.

The conjugacy is a key property that we can use in our EM algorithm proposed in Section 3.3.2. We saw that the solution of the E step is given by $q^{(t+1)}(\lambda) = p(\lambda \mid s^{(t)})$. In the factorial model shown in Figure 3.4 we have $p(\lambda \mid s) = \prod_i p(\lambda_i \mid s_i^{(t)})$. The solution of the E step is therefore a product of Gamma distributions with parameters $\alpha+1$ and $\beta+|s_i^{(t)}|$, and the sufficient statistics (3.9) are given by

$$\langle \lambda_i \rangle_{p(\lambda_i | s_i^{(t)})} = \frac{\alpha+1}{\beta+|s_i^{(t)}|}. \tag{3.16}$$

We can thus rewrite (3.8) as follows

$$s^{(t+1)} = \arg\min_{s} \ \frac{1}{2\sigma^2}\|x - \Phi s\|_2^2 + \sum_{i=1}^{m} \frac{\alpha + 1}{\beta + |s_i^{(t)}|}|s_i|. \tag{3.17}$$

Inference in the model can be solved via a sequence of reweighted ℓ_1-regularized least-square problems. The parameters λ_i are typically initialized to $\mathbb{E}[\lambda_i] = \alpha/\beta$ for all i. The first step is thus equivalent to solving BPDN. A coefficient that has a small value after t iterations but is not exactly zero will have in the next iteration a large reweighting factor $\lambda_i^{(t+1)}$, which increases the chance that it will be set to zero in the next iteration, resulting in a sparser representation. On the other hand, a coefficient having a large value after t iterations corresponds to a feature that is very salient in the signal x. It is therefore beneficial to reduce its corresponding inverse scale $\lambda_i^{(t+1)}$ such that it is not penalized and can account for as much information as possible.

We saw that with the Gamma prior we can compute the distribution of s_i analytically (see (3.1)), and therefore we can compute the gradient of $\log p(s \mid x)$ with respect to s. Hence another inference algorithm is to descend the cost function in (3.3) directly using a method such as conjugate gradient. We argue here that the EM algorithm is in fact more efficient. The solution of (3.17) indeed has typically few elements that are non-zero, and the computational complexity scales with the number of non-zero coefficients [Efron *et al.*, 2004][Daubechies *et al.*, 2004][Rozell *et al.*, 2008]. On the other hand, a gradient-based method will have a harder time identifying the support of the solution, and therefore the required computations will involve all the coefficients which is expensive.

3.4.2 A connection with reweighted ℓ_1 optimization methods

It has been proposed in [Candès *et al.*, 2008] to solve the following sequence of problems

$$s^{(t+1)} = \arg\min_{s} \sum_{i=1}^{m} \lambda_t^{(t)} |s_i| \quad \text{subject to} \quad \|x - \Phi s\|_2 \leq \delta \tag{3.18}$$

$$\lambda_i^{(t+1)} = \frac{1}{\beta + |s_i^{(t)}|}. \tag{3.19}$$

The authors show that the solutions achieved by their algorithm are more sparse than the solution of

$$\min_{s} \sum_{i=1}^{m} |s_i| \quad \text{subject to} \quad \|x - \Phi s\|_2 \leq \delta. \tag{3.20}$$

The update (3.19) is equivalent to the update we propose in (3.16). Hence our proposed probabilistic model leads to an optimization scheme that is akin to the one proposed in [Candès *et al.*, 2008] for the unconstrained problem. We provide an interpretation for their algorithm as inference in a probabilistic generative model.

It was shown in [Wipf and Nagarajan, 2008] that evidence maximization in a sparse coding model with automatic relevance determination prior can also be solved via a sequence of reweighted ℓ_1 optimization problems. The update is in this case non-factorial, i.e. $\lambda_i^{(t+1)}$ depends on $(s_1^{(t)}, \ldots, s_1^{(t)})$ as opposed to $s_i^{(t)}$ only. The authors show indeed that their algorithm is equivalent to MAP estimation in a sparse coding model with a non-factorial prior in coefficient space, where the dependencies are governed by the features and the noise. Note that this is different from the non-factorial prior proposed in Chapter 2 and the one we consider in Section 3.5 where the statistical dependencies are governed by the statistics of the signals of interest.

3.4.3 Application to image coding

We saw in Section 1.4.1 that the convex relaxation consisting of replacing the ℓ_0 norm with the ℓ_1 norm is able to identify the sparsest solution under some conditions on the dictionary of basis functions. However, these conditions are typically not verified for the dictionaries learned from the statistics of natural images using the algorithm presented in Section 1.4.4, or for the set of basis functions in the steerable pyramid [Simoncelli *et al.*, 1992]. For instance, we observed indeed in Section 2.5 that it is possible to infer sparser representations with a prior over the coefficients that is a mixture of a delta function at zero and a Gaussian distribution than with the Laplacian prior. We show that our proposed inference algorithm also leads to representations that are more sparse, as the LSM prior with Gamma hyperprior has heavier tails than the Laplacian distribution.

We selected 1000 16×16 image patches at random, and computed their sparse representations in a dictionary with 256 basis functions using a Laplacian prior and the LSM prior with factorial Gamma hyperprior. To ensure that the reconstruction error is the same in both cases, we solve the constrained version of the problem as in [Candès *et al.*, 2008], where we require that the signal to noise ratio of the reconstruction is equal to 10. We choose $\beta = 0.01$ and 5 EM iterations. We can see in Figure 3.5 that the representations using the LSM prior are indeed more sparse by a factor of about 2. Note that the computational complexity to compute those sparse representations is much lower than that of our horizontal connections model.

3.5 A non-factorial model

Many real-world signals such as sound or images have a sparse structure, but this property is not enough to fully characterize their statistics. We focus in this Section

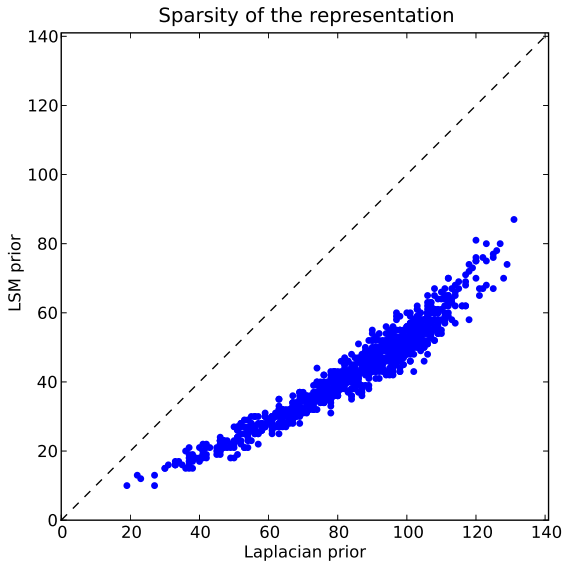


Figure 3.5: Sparsity comparison of the inferred representation with a Laplacian prior and an LSM prior with Gamma hyperprior on the coefficients. We code 1000 image patches such that the signal to noise ratio is 10, and each dot represents an image patch. On the x-axis (resp. y-axis) is the ℓ_0 norm of the representation inferred with the Laplacian prior (resp. LSM prior).

on a class of signals that has a particular type of higher-order structure where the active coefficients occur in groups. We use the LSM framework to propose an efficient inference algorithm that utilizes this property, and show that it is applicable to images.

3.5.1 Group sparsity

We consider a dictionary Φ such that the basis functions can be divided in a set of disjoint groups or neighborhoods indexed by $\mathcal{N}_k$, i.e. $\{1, \ldots, m\} = \bigcup_{k \in \Lambda} \mathcal{N}_k$, and $\mathcal{N}_i \cap \mathcal{N}_j = \emptyset$ if $i \neq j$. A signal having the group sparsity property is such that the sparse coefficients occur in groups, i.e. the indices of the nonzero coefficients are given by $\bigcup_{k \in \Gamma} \mathcal{N}_k$, where Γ is a subset of Λ.

The group sparsity structure can be captured using LSM priors on the coefficients. We propose a model where all the coefficients in a group share the same inverse scale parameter, i.e.

$$\forall i \in \mathcal{N}_k, \quad \lambda_i = \lambda_{(k)}.$$

The corresponding graphical model is shown in Figure 3.6. This addresses the case where dependencies are allowed within groups, but not across groups as in the block-ℓ_1 method [Yuan and Lin, 2006]. Note that for some types of dictionaries it is more natural to consider overlapping groups to avoid blocking artifacts. We propose in the next Section inference algorithms for both overlapping and non-overlapping cases. Note that a related notion of clustered sparsity parameterized by the number of nonzero coefficients and number of clusters was recently introduced in [Cevher *et al.*, 2009].

3.5.2 Inference

In the EM algorithm we proposed in Section 3.3.2, the sufficient statistics that are computed in the E step are $\langle \lambda_i \rangle_{p(\lambda_i | s^{(t)})}$ for all i. We suppose as in Section 3.4.1 that

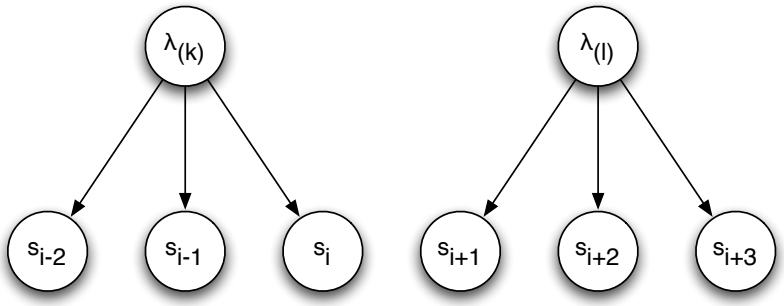


Figure 3.6: The two groups $\mathcal{N}_{(k)} = \{i-2, i-1, i\}$ and $\mathcal{N}_{(l)} = \{i+1, i+2, i+3\}$ are non-overlapping.

the prior on $\lambda_{(k)}$ is Gamma with parameters α and β. Using the structure of the dependencies in the probabilistic model shown in Figure 3.6, we have

$$\langle \lambda_i \rangle_{p(\lambda_i | s^{(t)})} = \left\langle \lambda_{(k)} \right\rangle_{p(\lambda_{(k)} | s^{(t)}_{\mathcal{N}_k})}, \tag{3.21}$$

where the index i is in the group $\mathcal{N}_k$, and $s_{\mathcal{N}_k} = (s_j)_{j \in \mathcal{N}_k}$ is the vector containing all the coefficients in the group. Using the conjugacy of the Laplacian and Gamma distributions we can compute the posterior distribution

$$\begin{aligned} p(\lambda_{(k)} \mid s_{\mathcal{N}_k}) &\propto p(s_{\mathcal{N}_k} \mid \lambda_{(k)}) p(\lambda_{(k)}) \\ &\propto \left(\prod_{j \in \mathcal{N}_k} p(s_j \mid \lambda_{(k)}) \right) p(\lambda_{(k)}) \\ &\propto \lambda_{(k)}^{\alpha + |\mathcal{N}_k| - 1} e^{-(\beta + \sum_{j \in N_k} |s_j|)\lambda_{(k)}}, \end{aligned}$$

where $|\mathcal{N}_k|$ denotes the size of the neighborhood. The distribution of $\lambda_{(k)}$ given all the coefficients in the neighborhood is therefore a Gamma distribution with parameters $\alpha + |N_k|$ and $\beta + \sum_{j \in N_k} |s_j|$. Hence (3.21) can be rewritten as follows

$$\lambda_{(k)}^{(t+1)} = \frac{\alpha + |\mathcal{N}_k|}{\beta + \sum_{j \in \mathcal{N}_k} |s_j^{(t)}|}. \tag{3.22}$$

This update is a form of divisive normalization, an operation thought to play an important role in human visual processing [Wainwright *et al.*, 2001a].

We suppose now that the coefficient neighborhoods are allowed to overlap. Let $\mathcal{N}(i)$ denote the indices of the neighborhood that is centered around s_i (see Figure 3.7 for an example). We propose to estimate the scale parameter λ_i by only considering the coefficients in $\mathcal{N}(i)$, and suppose that they all share the same multiplier λ_i. In this case the EM update is given by

$$\lambda_i^{(t+1)} = \frac{\alpha + |\mathcal{N}(i)|}{\beta + \sum_{j \in \mathcal{N}(i)} |s_j^{(t)}|}. \tag{3.23}$$

Note that we have not derived this rule from a proper probabilistic model. A coefficient is indeed a member of many neighborhoods as shown in Figure 3.7, and the structure of the dependencies implies

$$p(\lambda_i \mid s) \neq p(\lambda_i \mid s_{N(i)}).$$

However, we show experimentally that estimating the multiplier using (3.23) gives good performance.

In [Figueras and Simoncelli, 2007], the noise shaping algorithm, which bears similarities with iterative thresholding algorithm developed for BPDN [Rozell *et al.*, 2008], is modified such that the update is given by

$$\lambda_i^{(t+1)} \propto \sqrt{\beta + \sum_{j \in \mathcal{N}(i)} {s_j^{(t)}}^2}. \tag{3.24}$$

The authors show improved coding efficiency in the context of natural images. Note that our proposed update (3.23) is essentially inversely proportional to (3.24).

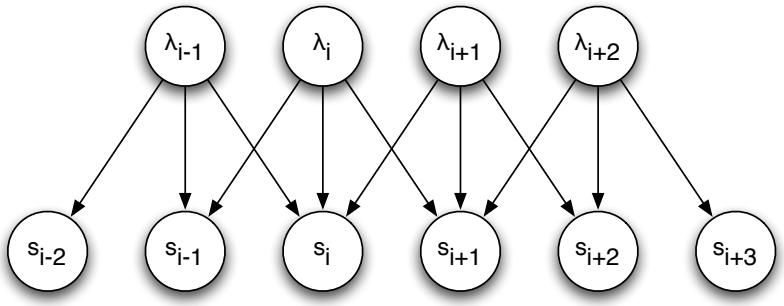


Figure 3.7: The basis functions coefficients in the neighborhood defined by $\mathcal{N}(i) = \{i-1, i, i+1\}$ share the same multiplier λ_i. The coefficient s_i is a member of the neighborhoods $\mathcal{N}(i-1)$, $\mathcal{N}(i)$ and $\mathcal{N}(i+1)$. However, to estimate λ_i only the coefficients in $\mathcal{N}(i)$ are considered.

3.5.3 Compressive sensing recovery

We saw in Section 1.1.3 that a way to compare signal priors is to look at the performance of the corresponding inference algorithms in ill-posed inverse problems. We focus here on compressive sensing recovery using synthetic data that have the overlapping group sparsity structure. We consider 50-dimensional signals that are sparse in the canonical basis and where the neighborhood size is 3. To sample such a signal $s \in \mathbb{R}^{50}$, we sample a number d of "centroids" i, and we sample three values for s_{i-1}, s_i and s_{i+1} using a normal distribution of variance 1. The groups are thus allowed to overlap. We show examples of such signals in Figure 3.8.

In the compressive sensing scenario, we observe a number n of random projections of a signal s_0 from our overlapping group sparsity class. Let $W \in \mathbb{R}^{n \times m}$ denote the measurement matrix and $y = Ws_0$. It is in principle impossible to recover s_0 from y if $n < m$. However, if s_0 has k non-zero coefficients, it has been shown in [Candès, 2006][Donoho, 2006a] that it is sufficient to use $n \propto k \log m$ such measurements. This means that we can identify the active coefficients and their values using more measurements than the sparsity of the signal, but fewer measurements than the

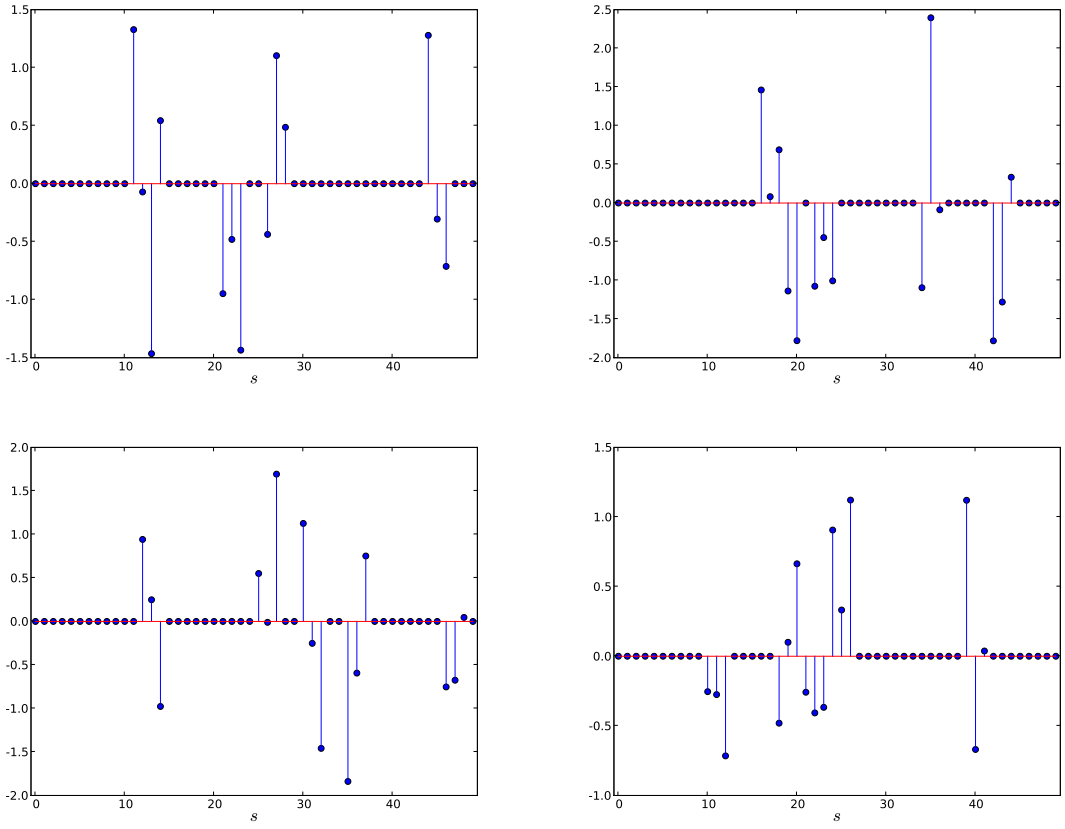


Figure 3.8: Examples of signals with overlapping group sparsity. We set $m = 50$ and the number of sampled "centroids" is $d = 5$.

dimensionality of the signal. A standard method to obtain the reconstruction is to use the solution of the Basis Pursuit (BP) problem

$$\hat{s} = \arg\min_{s} \|s\|_1 \quad \text{subject to} \quad Ws = y.$$

The performance metric is the recovery error $\|\hat{s} - s_0\|_2 / \|s_0\|_2$. Note that the solution of BP is the solution of BPDN as λ converges to zero, or $\delta = 0$ in (3.20). We compare the performance of BP with the performance of our proposed LSM inference algorithms

$$\min_{s} \sum_{i=1}^{m} \lambda_i^{(t)} |s_i| \quad \text{subject to} \quad Ws = y,$$

where

$$\lambda_i^{(t+1)} = \begin{cases} \frac{\alpha+1}{\beta+|s_i^{(t)}|}, & \text{factorial update} \\ \frac{\alpha+|\mathcal{N}(i)|}{\beta+\sum_{j\in\mathcal{N}(i)}|s_j|}, & \text{divisive normalization update.} \end{cases} \tag{3.25}$$

We denote by RWBP the algorithm with the factorial update, and RW_3BP (resp. RW_5BP) the algorithm with our proposed divisive normalization update with group size 3 (resp. 5).

A compressive sensing recovery problem is parameterized by (m, n, d). To explore the problem space we display the results using phase plots as in [Donoho and Tsaig, 2006]. We fix $m = 50$ and parameterize the phase plots using the indeterminacy of the system indexed by $\delta = n/m$, and the sparsity of the system indexed by $\rho = 3d/m$. We vary δ and ρ in the range $[.1, .9]$ using a 30 by 30 grid. For a given value (δ, ρ) on the grid, we sample 10 sparse signals using the corresponding (m, n, d) parameters. We attempt to recover the underlying sparse signal using the three algorithms and average the recovery error for each of them. The results are displayed in Figure 3.9, and we can see that our proposed algorithm with the divisive normalization update clearly has the best performance. There is a slight improvement by going from BP

to RWBP, but this improvement is rather small as compared with going from RWBP to RW_3BP and RW_5BP. This illustrates the importance of using the higher-order structure of the signals in the inference algorithm. The peformance of RW_3BP and RW_5BP is comparable, which shows that our algorithm is not very sensitive to the choice of the neighborhood size.

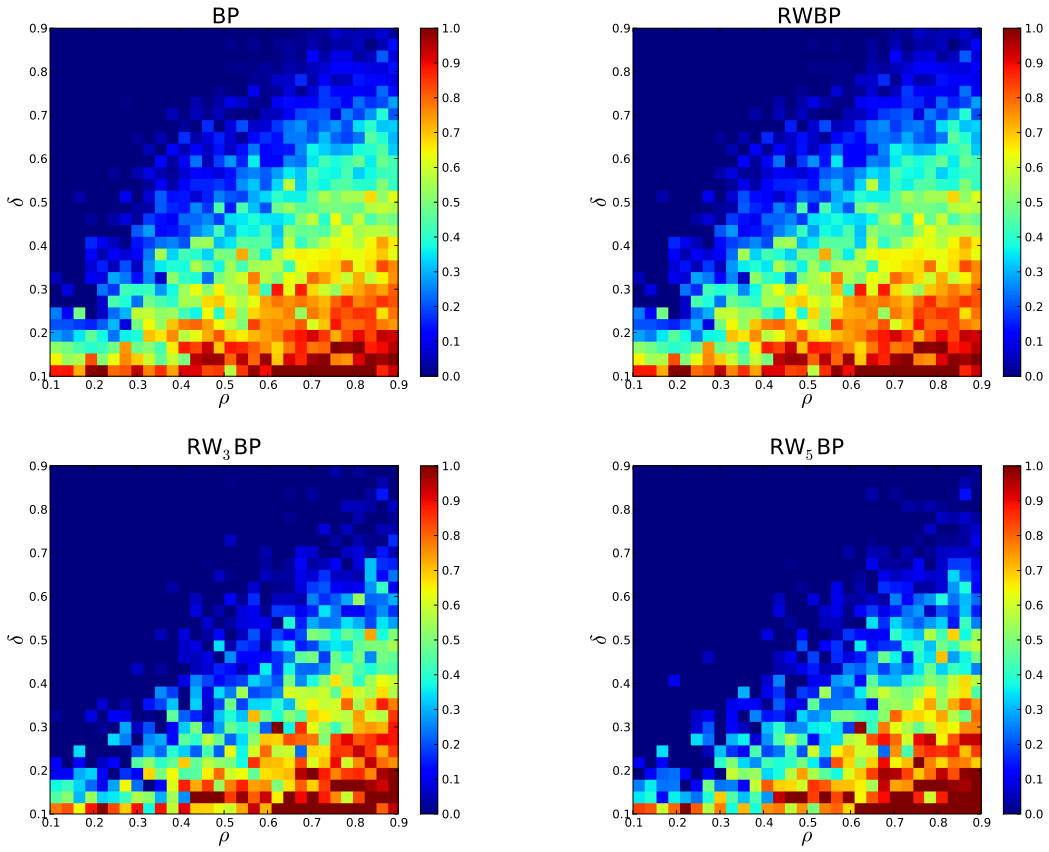


Figure 3.9: Compressive sensing recovery results using synthetic data. We show the phase plots for BP, a sequence of BP problems with the factorial update (RWBP), and a sequence of BP problems with the divisive normalization update with neighborhood size 3 (RW_3BP) and 5 (RW_5BP). On the x-axis is the sparsity of the system indexed by $\rho = 3d/m$, and on the y-axis is the indeterminacy of the system indexed by $\delta = n/m$. At each point (ρ, δ) in the phase plot, we sample 10 compressive sensing problems and display the average recovery error.

3.5.4 Application to images

We saw in Section 1.4.4 that natural images are sparse with respect to dictionaries that are composed of Gabor-like basis functions at a variety of positions, scales, and orientations. We learned in Chapter 2 a non-factorial prior on the coefficients such that the statistical dependencies are governed by a weight matrix we learned from data. We saw that a basis function coefficient exhibits statistical dependencies with the coefficients corresponding to basis functions that have a similar position, scale, and orientation. Hence, for a dictionary composed of oriented Gabor-like filters, it is natural to define a topology in terms of these parameters. Furthermore, it was shown in [Hyvärinen *et al.*, 2003] that structures in images such as edges and contours are formed by combinations of basis functions that are close in position, scale, and orientation. The authors denote a set of active coefficients used to represent such as structure a "bubble".

The overlapping group sparsity is therefore relevant to images, and we show that our proposed algorithm improves the performance in compressive sensing recovery for the reconstruction from a multi-scale subband of the Shepp-Logan phantom. This image shown in Figure 3.10 is a good example of the types of images in medical imaging and has edge and contour structures. Considering the reconstruction from a multiscale subband allows us to control the dimensionality of the problem, computational complexity, and memory requirements by limiting the size of the subband. A natural topography is also in this case particularly simple to define and we choose a grid. We consider overlapping groups of size 3×3.

Let $(\varphi_i)_{i\in\Gamma}$ denote the basis functions in the steerable pyramid [Simoncelli *et al.*, 1992]. As it is a tight frame, we have

$$x = \kappa \sum_{i\in\Gamma} a_i \varphi_i$$

Figure 3.10: The Shepp-Logan phantom.

for some constant κ, where a is the vector of analysis coefficients, i.e. $a_i = x^T\varphi_i$. Let $\Phi = (\varphi_i)_{i\in\Lambda}$ be the set of basis functions corresponding to a multi-scale oriented subband of the steerable pyramid, and m be the number of basis functions in the subband Λ. These basis functions are the translations at all possible positions of the atom shown in Figure 3.11.

The reconstruction from the coefficients in the subband is given by

$$\tilde{x} = \kappa \sum_{i\in\Lambda} a_i\varphi_i \tag{3.26}$$

and is shown in Figure 3.12. We have therefore $\tilde{x} \in \text{span}(\Phi)$, and $\tilde{x}$ has a sparse representation in this basis. Note that the analysis coefficients do not in general correspond to the representation that is the most sparse.

In the compressive sensing scenario, we observe $y = W\tilde{x}$, where $W \in \mathbb{R}^{k\times m}$ is a matrix of random projections where each element is Gaussian with unit variance.

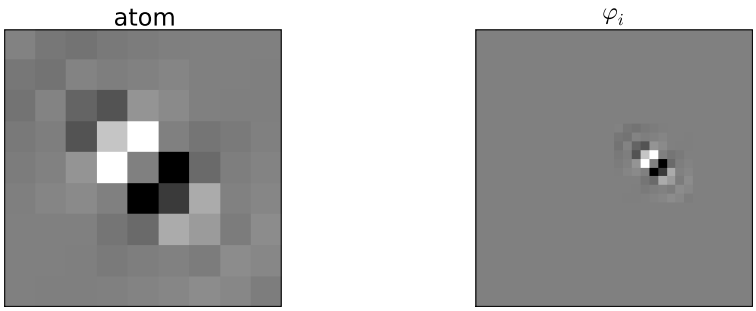


Figure 3.11: **Left** The atom of size 9×9 pixels that is used to generate all the basis functions by placing the atom at every possible position. **Right** An example of such a basis function of size 32×32 pixels.

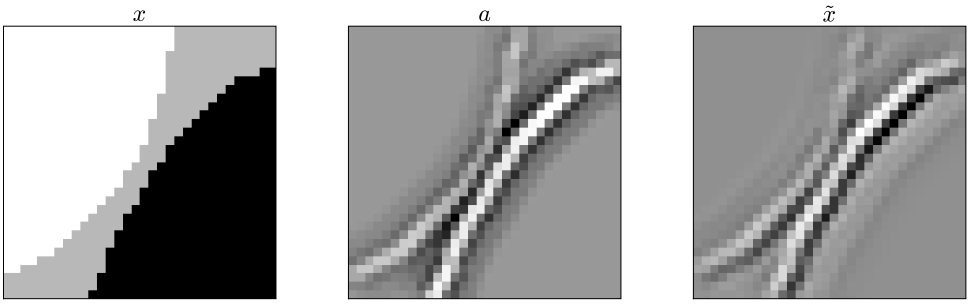


Figure 3.12: **Left** A 32×32 cutout of the Shepp-Logan phantom. **Middle** Analysis coefficients. **Right** Reconstruction using the analysis coefficients.

The reconstruction given by BP is given by $\Phi\hat{s}$, where $\hat{s}$ is the solution of

$$\min_s \|s\|_1 \quad \text{subject to} \quad W\Phi s = y.$$

The performance metric is the signal-to-noise ratio

$$20 \log_{10} \left(\frac{\|\tilde{x}\|_2}{\|\Phi\hat{s} - \tilde{x}\|_2} \right)$$

We compare the performance of BP with the performance of our proposed LSM inference algorithms

$$\min_s \sum_{i=1}^{m} \lambda_i^{(t)} |s_i| \quad \text{subject to} \quad W\Phi s = y,$$

with the same updates as in (3.25). We denote by $\text{RW}_{3\times3}$BP the algorithm with divisive normalization update and 3×3 groups. We can see in Figure 3.13 that $\text{RW}_{3\times3}$BP offers the best performance. The signal-to-noise ratio of the recovered signal is indeed superior to the other method by more than 1dB when the number of observations is between 150 and 400. When the number of observations is above 400, all methods are able to correctly recover the input image. Note that with very few observations the three methods perform equally poorly. We display in Figure 3.14 and 3.15 the coefficients inferred using the three algorithms. The coefficients inferred by $\text{RW}_{3\times3}$BP are clustered and able to identify where the important structure in the image lies, whereas the coefficients inferred using the other methods are more dispersed.

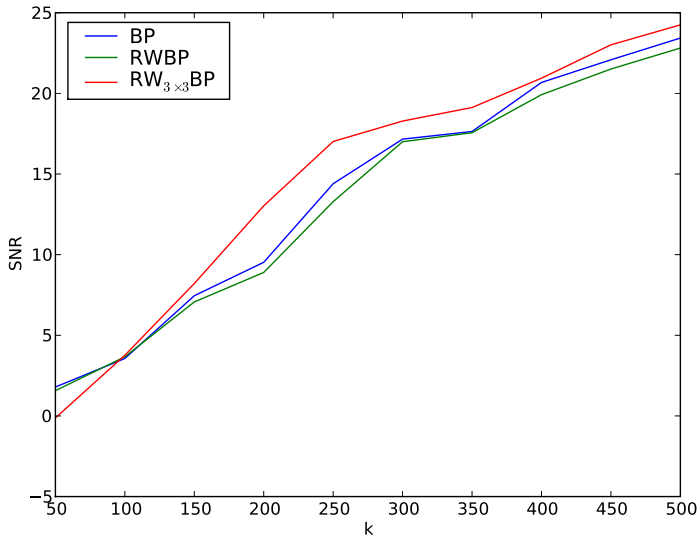


Figure 3.13: Compressive sensing recovery. On the x-axis is the number of observations k, and on the y-axis is the signal-to-noise ratio of the reconstruction. We compare the three algorithms BP, RWBP and $\text{RW}_{3\times3}$BP

3.6 Conclusion

We introduced a new class of probability densities that can be used as the coefficients prior in sparse coding models. We proposed efficient inference algorithms that consist of solving a sequence of reweighted ℓ_1 least-square problems, and can therefore leverage the algorithms developed for BPDN. Our framework also makes it possible to capture higher-order structure beyond sparsity, and we demonstrated improvements in compressive sensing recovery for signals having the group sparsity property.

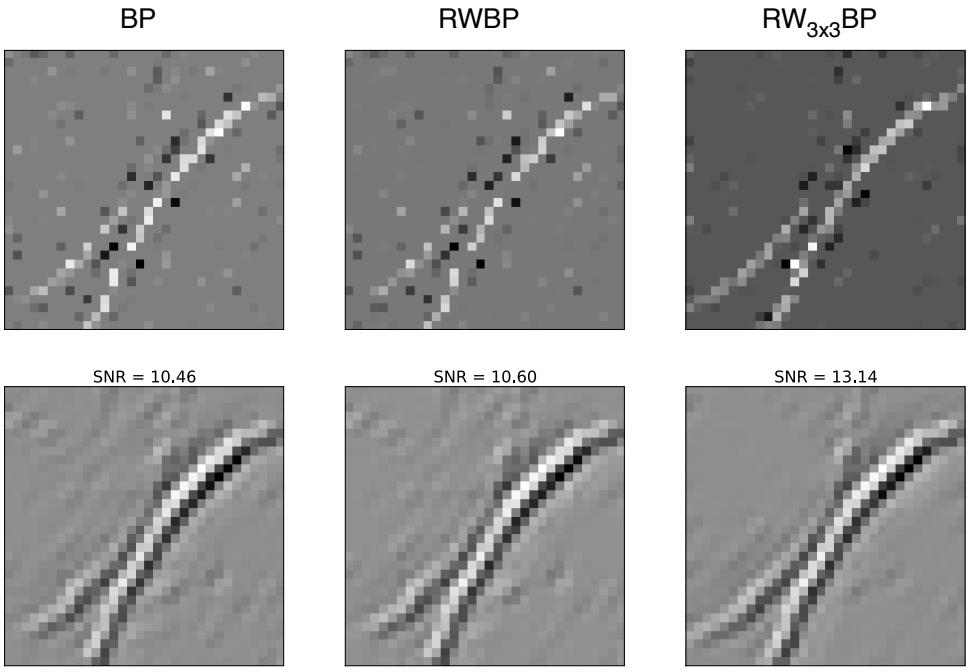


Figure 3.14: Inferred coefficients (top) and reconstructed image (bottom) with $k = 200$ observations.

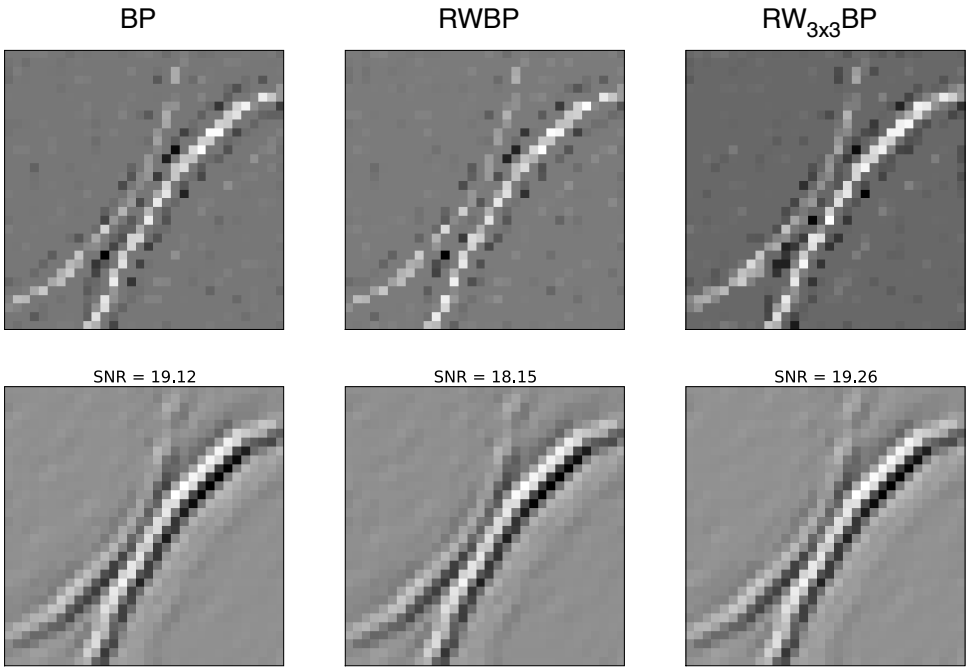


Figure 3.15: Inferred coefficients (top) and reconstructed image (bottom) with $k = 350$ observations.

Chapter 4

An homotopy algorithm with on-line observations

4.1 Introduction

Regularization using the ℓ_1-norm has attracted much interest in the statistics [Tibshirani, 1996], signal processing [Chen *et al.*, 1999], and machine learning communities. The ℓ_1 penalty indeed leads to sparse solutions, which is a desirable property to achieve model selection, data compression, or for obtaining interpretable results. In this Chapter, we focus on the problem of ℓ_1-penalized least-square regression commonly referred to as the Lasso [Tibshirani, 1996]. We have seen in Chapter 1 how this problem can be used to compute sparse approximations of signals with respect to an overcomplete dictionary. We investigate in this Chapter how this problem is used in statistics, and propose an efficient algorithm in the on-line observations settings. We are given a set of training examples or observations $(y_i, x_i) \in \mathbb{R} \times \mathbb{R}^m,\ i = 1 \dots n$. We wish to fit a linear model to predict the response y_i as a function of x_i and a feature vector $\theta \in \mathbb{R}^m$, $y_i = x_i^T \theta + \nu_i$, where ν_i represents the noise in the observation. The

Lasso optimization problem is given by

$$\min_{\theta} \ \frac{1}{2}\sum_{i=1}^{n}(x_i^T\theta - y_i)^2 + \mu_n\|\theta\|_1, \tag{4.1}$$

where μ_n is a regularization parameter. The solution of (4.1) is typically sparse, i.e. the solution θ has few entries that are non-zero, and therefore identifies which dimensions in x_i are useful to predict y_i. Note that the cost function is the sum of a squared-error term and the ℓ_1 norm, as in the Basis Pursuit Denoising problem that we have encountered in the context of image coding.

The ℓ_1-regularized least-square problem can be formulated as a convex quadratic problem (QP) with linear equality constraints. The equivalent QP can be solved using standard interior-point methods (IPM) [Boyd and Vandenberghe, 2004] which can handle medium-sized problems. A specialized IPM for large-scale problems was recently introduced in [Kim *et al.*, 2007]. Homotopy methods have also been applied to the Lasso to compute the full regularization path when λ varies [Efron *et al.*, 2004] [Osborne *et al.*, 2000][Malioutov *et al.*, 2005]. They are particularly efficient when the solution is very sparse [Drori and Donoho, 2006]. Other methods to solve (4.1) include iterative thresholding algorithms [Daubechies *et al.*, 2004][Rozell *et al.*, 2007][Friedman *et al.*, 2007], feature-sign search [Lee *et al.*, 2007], bound optimization methods [Figueiredo and Nowak, 2005] and gradient projection algorithms [Figueiredo *et al.*, 2007].

We propose an algorithm to compute the solution of the Lasso when the training examples $(y_i, x_i)_{i=1\ldots N}$ are obtained sequentially. Let $\theta^{(n)}$ be the solution of the Lasso after observing n training examples and $\theta^{(n+1)}$ the solution after observing a new data point $(y_{n+1}, x_{n+1}) \in \mathbb{R} \times \mathbb{R}^m$. We introduce an optimization problem that allows us to compute an homotopy from $\theta^{(n)}$ to $\theta^{(n+1)}$. Hence we use the previously computed solution as a "warm-start", which makes our method particularly efficient when the

supports of $\theta^{(n)}$ and $\theta^{(n+1)}$ are close. A similar algorithm appeared independently in [Asif and Romberg, 2008].

In Section 2 we review the optimality conditions of the Lasso, which we use in Section 3 to derive our algorithm. We test in Section 4 our algorithm numerically, and show applications to compressive sensing with sequential observations and leave-one-out cross-validation. We also propose an algorithm to automatically select the regularization parameter each time we observe a new data point.

4.2 Optimality conditions for the Lasso

The objective function in (4.1) is convex and non-smooth since the ℓ_1 norm is not differentiable when $\theta_i = 0$ for some i. Hence there is a global minimum at θ if and only if the subdifferential of the objective function at θ contains the 0-vector. The subdifferential of the ℓ_1-norm at θ is the following set

$$\partial\|\theta\|_1 = \left\{ v \in \mathbb{R}^m \ : \ \begin{cases} v_i = \text{sgn}(\theta_i) \text{ if } |\theta_i| > 0 \\ v_i \in [-1,1] \text{ if } \theta_i = 0 \end{cases} \right\}.$$

Let $X \in \mathbb{R}^{n \times m}$ be the matrix whose i^{th} row is equal to x_i^T, and $y = (y_1, \ldots, y_n)^T$. The optimality conditions for the Lasso are given by

$$X^T(X\theta - y) + \mu_n v = 0, \ v \in \partial\|\theta\|_1.$$

We define as the active set the indices of the elements of θ that are non-zero. To simplify notations we assume that the active set appears first, i.e. $\theta^T = (\theta_1^T, 0^T)$ and $v^T = (v_1^T, v_2^T)$, where $v_{1i} = \text{sgn}(\theta_{1i})$ for all i, and $-1 \leq v_{2j} \leq 1$ for all j. Let $X = (X_1 \ X_2)$ be the partitioning of X according to the active set. If the solution is

unique it can be shown that $X_1^T X_1$ is invertible, and we can rewrite the optimality conditions as

$$\begin{cases} \theta_1 = (X_1^T X_1)^{-1}(X_1^T y - \mu_n v_1) \\ -\mu_n v_2 = X_2^T (X_1 \theta_1 - y) \end{cases}.$$

Note that if we know the active set and the signs of the coefficients of the solution, then we can compute it in closed form.

4.3 Proposed homotopy algorithm

4.3.1 Outline of the algorithm

Suppose we have computed the solution $\theta^{(n)}$ to the Lasso with n observation and that we are given an additional observation $(y_{n+1}, x_{n+1}) \in \mathbb{R} \times \mathbb{R}^m$. Our goal is to compute the solution $\theta^{(n+1)}$ of the augmented problem. We introduce the following optimization problem

$$\theta(t, \mu) = \arg\min_{\theta} \ \frac{1}{2} \left\| \begin{pmatrix} X \\ t x_{n+1}^T \end{pmatrix} \theta - \begin{pmatrix} y \\ t y_{n+1} \end{pmatrix} \right\|_2^2 + \mu \|\theta\|_1. \tag{4.2}$$

We have $\theta^{(n)} = \theta(0, \mu_n)$ and $\theta^{(n+1)} = \theta(1, \mu_{n+1})$. We propose an algorithm that computes a path from $\theta^{(n)}$ to $\theta^{(n+1)}$ in two steps:

- **Step 1** Vary the regularization parameter from μ_n to μ_{n+1} with $t = 0$. This amounts to computing the regularization path between μ_n and μ_{n+1} as done in Lars. The solution path is piecewise linear and we do not review it in this Chapter (see [Osborne, 1992][Malioutov *et al.*, 2005][Efron *et al.*, 2004]).

- **Step 2** Vary the parameter t from 0 to 1 with $\mu = \mu_{n+1}$. We show in Section 4.3.2 how to compute this path.

4.3.2 Algorithm derivation

We show in this Section that $\theta(t, \mu)$ is a piecewise smooth function of t. To make notations lighter we write $\theta(t) := \theta(t, \mu)$. We saw in Section 4.2 that the solution to the Lasso can be easily computed once the active set and signs of the coefficients are known. This information is available at $t = 0$, and we show that the active set and signs will remain the same for t in an interval $[0, t^*)$ where the solution $\theta(t)$ is smooth. We denote such a point where the active set changes a "transition point" and show how to compute it analytically. At t^* we update the active set and signs which will remain valid until t reaches the next transition point. This process is iterated until we know the active set and signs of the solution at $t = 1$, and therefore can compute the desired solution $\theta^{(n+1)}$.

We suppose as in Section 4.2 and without loss of generality that the solution at $t = 0$ is such that $\theta(0) = (\theta_1^T, 0^T)$ and $v^T = (v_1^T, v_2^T) \in \partial\|\theta(0)\|_1$ satisfy the optimality conditions.

Lemma 1. Suppose $\theta_{1i} \neq 0$ for all i and $|v_{2j}| < 1$ for all j. There exist $t^* > 0$ such that for all $t \in [0, t^*)$, the solution of (4.2) has the same support and the same sign as $\theta(0)$.

Proof. The optimality conditions of (4.2) are given by

$$X^T (X\theta - y) + t^2 x_{n+1} \left(x_{n+1}^T \theta - y_{n+1}\right) + \mu w = 0, \tag{4.3}$$

where $w \in \partial\|\theta\|_1$. We show that there exists a solution $\theta(t)^T = (\theta_1(t)^T, 0^T)$ and $w(t)^T = (v_1^T, w_2(t)^T) \in \partial\|\theta(t)\|_1$ satisfying the optimality conditions for t sufficiently small. We partition $x_{n+1}^T = (x_{n+1,1}^T, x_{n+1,2}^T)$ according to the active set. We rewrite

the optimality conditions as

$$\begin{cases} X_1^T \left(X_1\theta_1(t) - y\right) + t^2 x_{n+1,1} \left({x_{n+1,1}}^T\theta_1(t) - y_{n+1}\right) + \mu v_1 = 0 \\ X_2^T \left(X_1\theta_1(t) - y\right) + t^2 x_{n+1,2} \left({x_{n+1,1}}^T\theta_1(t) - y_{n+1}\right) + \mu w_2(t) = 0 \end{cases}.$$

Solving for $\theta_1(t)$ using the first equation gives

$$\theta_1(t) = \left(X_1^T X_1 + t^2 x_{n+1,1}\ {x_{n+1,1}}^T\right)^{-1} \left(X_1^T y + t^2 y_{n+1}\ x_{n+1,1} - \mu v_1\right). \tag{4.4}$$

We can see that $\theta_1(t)$ is a continuous function of t. Since $\theta_1(0) = \theta_1$ and the elements of θ_1 are all strictly positive, there exists t_1^* such that for $t < t_1^*$, all elements of $\theta_1(t)$ remain positive and do not change signs. We also have

$$-\mu_{n+1} w_2(t) = X_2^T \left(X_1\theta_1(t) - y\right) + t^2 x_{n+1,2} \left({x_{n+1,1}}^T\theta_1(t) - y_{n+1}\right). \tag{4.5}$$

Similarly $w_2(t)$ is a continuous function of t, and since $w_2(0) = v_2$, there exists t_2^* such that for $t < t_2^*$ all elements of $w_2(t)$ are strictly smaller than 1 in absolute value. By taking $t^* = \min(t_1^*, t_2^*)$ we obtain the desired result. □

The solution $\theta(t)$ will therefore be smooth until t reaches a transition point where either a component of $\theta_1(t)$ becomes zero, or one of the component of $w_2(t)$ reaches one in absolute value. We now show how to compute the value of the transition point.

Let $\tilde{X} = \begin{pmatrix} X \\ {x_{n+1}}^T \end{pmatrix}$ and $\tilde{y} = \begin{pmatrix} y \\ y_{n+1} \end{pmatrix}$. We partition $\tilde{X} = \begin{pmatrix} \tilde{X}_1 & \tilde{X}_2 \end{pmatrix}$ according to the active set. We use the Sherman-Morrison formula and rewrite (4.4) as

$$\theta_1(t) = \tilde{\theta}_1 - \frac{(t^2-1)\bar{e}}{1 + \alpha(t^2-1)} u,$$

where

$$\begin{cases} \tilde{\theta}_1 = (\tilde{X}_1^T \tilde{X}_1)^{-1}(\tilde{X}_1^T \tilde{y} - \mu v_1) \\ \bar{e} = {x_{n+1,1}}^T \tilde{\theta}_1 - y_{n+1} \\ \alpha = {x_{n+1,1}}^T (\tilde{X}_1^T \tilde{X}_1)^{-1} x_{n+1,1} \\ u = (\tilde{X}_1^T \tilde{X}_1)^{-1} x_{n+1,1} \end{cases}.$$

Let t_{1i} the value of t such that $\theta_{1i}(t) = 0$. We have

$$t_{1i} = \left(1 + \left(\frac{\bar{e}u_i}{\tilde{\theta}_{1i}} - \alpha\right)^{-1}\right)^{\frac{1}{2}}.$$

We now examine the case where a component of $w_2(t)$ reaches one in absolute value. We first notice that

$$\begin{cases} {x_{n+1,1}}^T \theta_1(t) - y_{n+1} = \frac{\bar{e}}{1+\alpha(t^2-1)} \\ \tilde{X}_1 \theta_1(t) - \tilde{y} = \tilde{e} - \frac{(t^2-1)\bar{e}}{1+\alpha(t^2-1)} \tilde{X}_1 u \end{cases},$$

where $\tilde{e} = \tilde{X}_1 \tilde{\theta}_1 - \tilde{y}$. We can rewrite (4.5) as

$$-\mu w_2(t) = \tilde{X}_2^T \tilde{e} + \frac{\bar{e}(t^2-1)}{1+\alpha(t^2-1)}(x_{n+1,2} - \tilde{X}_2^T \tilde{X}_1 u).$$

Let c_j be the j^{th} column of $\tilde{X}_2$, and $x^{(j)}$ the j^{th} element of $x_{n+1,2}$. The j^{th} component of $w_2(t)$ will become 1 in absolute value as soon as

$$\left| c_j^T \tilde{e} + \frac{\bar{e}(t^2-1)}{1+\alpha(t^2-1)} \left(x^{(j)} - c_j^T \tilde{X}_1 u\right) \right| = \mu.$$

Let $t_{2\,j}^{+}$ (resp. $t_{2\,j}^{-}$) be the value such that $w_{2j}(t) = 1$ (resp. $w_{2j}(t) = -1$). We have

$$\begin{cases} t_{2\,j}^{+} = \left(1 + \left(\frac{\bar{e}(x^{(j)} - c_j^T \tilde{X}_1 u)}{-\mu - c_j^T \tilde{e}} - \alpha\right)^{-1}\right)^{\frac{1}{2}} \\ t_{2\,j}^{-} = \left(1 + \left(\frac{\bar{e}(x^{(j)} - c_j^T \tilde{X}_1 u)}{\mu - c_j^T \tilde{e}} - \alpha\right)^{-1}\right)^{\frac{1}{2}} \end{cases} .$$

Hence the transition point will be equal to $t' = \min\{\min_i t_{1i},\ \min_j t_{2\,j}^{+},\ \min_j t_{2\,j}^{-}\}$ where we restrict ourselves to the real solutions that lie between 0 and 1. We now have the necessary ingredients to derive the proposed algorithm.

Algorithm 1 RecLasso: homotopy algorithm for online Lasso

1: Compute the path from $\theta^{(n)} = \theta(0, \mu_n)$ to $\theta(0, \mu_{n+1})$.
2: Initialize the active set to the non-zero coefficients of $\theta(0, \mu_{n+1})$ and let $v = \text{sign}\,(\theta(0, \mu_{n+1}))$.
Let v_1 and $x_{n+1,1}$ be the subvectors of v and x_{n+1} corresponding to the active set, and $\tilde{X}_1$ the submatrix of $\tilde{X}$ whose columns correspond to the active set.
Initialize $\tilde{\theta}_1 = (\tilde{X}_1^T \tilde{X}_1)^{-1}(\tilde{X}_1^T \tilde{y} - \mu v_1)$.
Initialize the transition point $t' = 0$.
3: Compute the next transition point t'. If it is smaller than the previous transition point or greater than 1, go to Step 5.

Case 1 The component of $\theta_1(t')$ corresponding to the i^{th} coefficient goes to zero:
Remove i from the active set.
Update v by setting $v_i = 0$.

Case 2 The component of $w_2(t')$ corresponding to the j^{th} coefficient reaches one in absolute value:
Add j to the active set.
If the component reaches 1 (resp. -1), then set $v_j = 1$ (resp. $v_j = -1$).

4: Update v_1, $\tilde{X}_1$ and $x_{n+1,1}$ according to the updated active set.
Update $\tilde{\theta}_1 = (\tilde{X}_1^T \tilde{X}_1)^{-1}(\tilde{X}_1^T \tilde{y} - \mu v_1)$ (rank 1 update).
Go to Step 3.
5: Compute final value at $t = 1$, where the values of $\theta^{(n+1)}$ on the active set are given by $\tilde{\theta}_1$.

The initialization amounts to computing the solution of the Lasso when we have

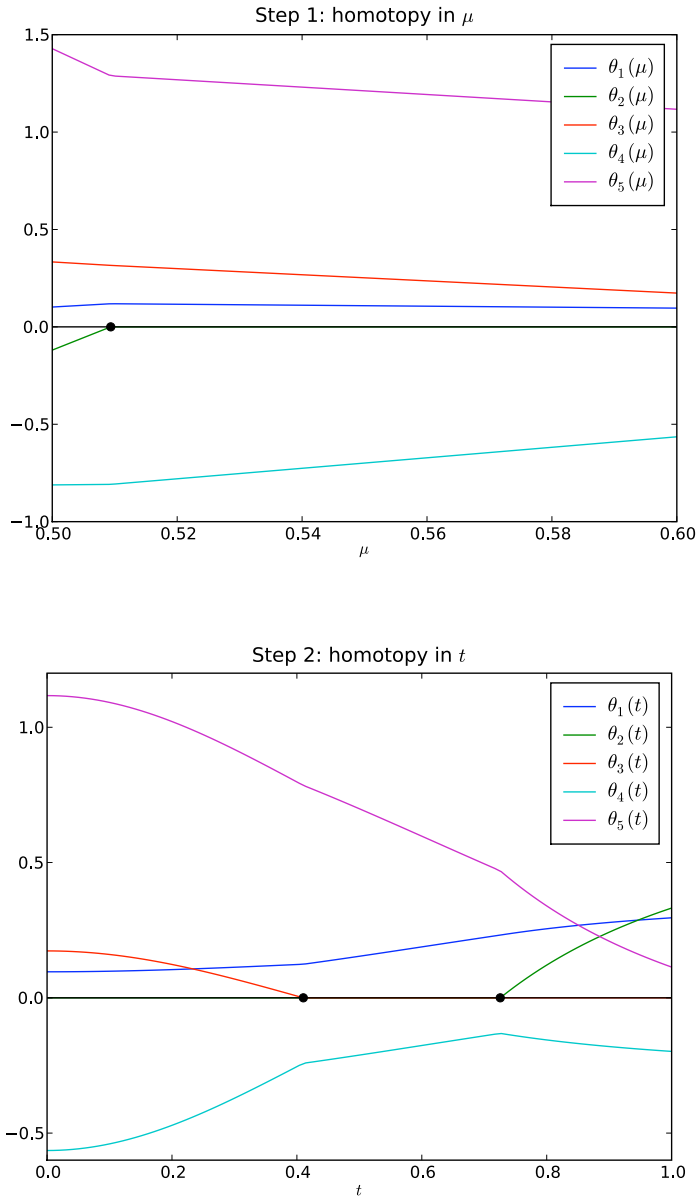


Figure 4.1: Solution path for both steps of our algorithm. We set $n = 5$, $m = 5$, $\mu_n = .1n$. All the values of X, y, x_{n+1} and y_{n+1} are drawn at random. **Top** Homotopy when the regularization parameter goes from $\mu_n = .5$ to $\mu_{n+1} = .6$. There is one transition point as θ_2 becomes inactive. **Bottom** Piecewise smooth path of $\theta(t)$ when t goes from 0 to 1. We can see that θ_3 becomes zero, θ_2 goes from being 0 to being positive, whereas θ_1, θ_4 and θ_5 remain active with their signs unchanged. The three transition points are shown as black dots.

only one data point $(y, x) \in \mathbb{R} \times \mathbb{R}^m$. In this case, the active set has at most one element. Let $i_0 = \arg\max_i |x^{(i)}|$ and $v = \text{sign}(yx^{(i_0)})$. We have

$$\begin{cases} \frac{1}{(x^{(i_0)})^2}(yx^{(i_0)} - \mu_1 v)e_{i_0}, & \text{if } |yx^{(i_0)}| > \mu_1 \\ 0, & \text{otherwise.} \end{cases}.$$

We illustrate our algorithm by showing the solution path when the regularization parameter and t are successively varied with a simple numerical example in Figure 4.1.

4.3.3 Complexity

The complexity of our algorithm is dominated by the inversion of the matrix $\tilde{X}_1^T \tilde{X}_1$ at each transition point. The size of this matrix is bounded by $q = \min(n, m)$. As the update to this matrix after a transition point is rank 1, the cost of computing the inverse is $O(q^2)$. Let k be the total number of transition points after varying the regularization parameter from μ_n to μ_{n+1} and t from 0 to 1. The complexity of our algorithm is thus $O(kq^2)$. In practice, the size of the active set d is much lower than q, and if it remains $\sim d$ throughout the homotopy, the complexity is $O(kd^2)$. It is instructive to compare it with the complexity of recursive least-square, which corresponds to $\mu_n = 0$ for all n and $n > m$. For this problem the solution typically has m non-zero elements, and therefore the cost of updating the solution after a new observation is $O(m^2)$. Hence if the solution is sparse (d small) and the active set does not change much (k small), updating the solution of the Lasso will be faster than updating the solution to the non-penalized least-square problem.

Suppose that we applied Lars directly to the problem with $n + 1$ observations without using knowledge of $\theta^{(n)}$ by varying the regularization parameter from a large value where the size of the active set is 0 to μ_{n+1}. Let k' be the number of transition

points. The complexity of this approach is $O(k'q^2)$, and we can therefore compare the efficiency of these two approaches by comparing the number of transition points.

4.4 Applications

4.4.1 Compressive sensing

Let $\theta_0 \in \mathbb{R}^m$ be an unknown vector that we wish to reconstruct. We observe n linear projections $y_i = x_i^T\theta_0 + \nu_i$, where ν_i is Gaussian noise of variance σ^2. In general one needs m such measurement to reconstruct θ_0. However, if θ_0 has a sparse representation with k non-zero coefficients, it has been shown in the noiseless case that it is sufficient to use $n \propto k \log m$ (see Section 3.5.3). The reconstruction is given by the solution of the Basis Pursuit (BP) problem

$$\min_{\theta} \|\theta\|_1 \quad \text{subject to} \quad X\theta = y.$$

If measurements are obtained sequentially, it is advantageous to start estimating the unknown sparse signal as measurements arrive, as opposed to waiting for a specified number of measurements. Algorithms to solve BP with sequential measurements have been proposed in [Sra and Tropp, 2006][Malioutov *et al.*, 2008], and it has been shown that the change in the active set gives a criterion for how many measurements are needed to recover the underlying signal [Malioutov *et al.*, 2008].

In the case where the measurements are noisy ($\sigma > 0$), a standard approach to recover θ_0 is to solve the Basis Pursuit DeNoising problem instead [Tsaig and Donoho, 2006]. Hence, our algorithm is well suited for compressive sensing with sequential and noisy measurements. We compare our proposed algorithm to Lars as applied to the entire dataset each time we receive a new measurement. We also compare our method to coordinate descent [Friedman *et al.*, 2007] with warm start: when receiving a new

measurement, we initialize coordinate descent (CD) to the actual solution.

We sample measurements of a model where $m = 100$, the vector θ_0 used to sample the data has 25 non-zero elements whose values are Bernoulli ± 1, $x_i \sim \mathcal{N}(0, I_m)$, $\sigma = 1$, and we set $\mu_n = .1n$. The reconstruction error decreases as the number of measurements grows as seen in Figure 4.2. The parameter that controls the complexity of Lars and RecLasso is the number of transition points. We see in Figure 4.3 that this quantity is consistently smaller for RecLasso, and that after 100 measurements when the support of the solution does not change much there are typically less than 5 transition points for RecLasso. We also show in Figure 4.3 timing comparison for the three algorithms that we have each implemented in Python. We observed that CD requires a lot of iterations to converge to the optimal solution when $n < m$, and we found difficult to set a stopping criterion that ensures convergence. Our algorithm is consistently faster than Lars and CD with warm-start.

We also sample measurements of a model where there is no observation noise, i.e. $\sigma = 0$. It has been shown in [Zhao and Yu, 2006] that Lasso recovers the active set of the original vector θ_0 under a simple condition on the generating covariance matrix of the observations, with a regularization schedule $\mu_n \sim \sqrt{n}$. We show in Figure 4.4 that the Hamming distance between $\theta^{(n)}$ and θ_0 indeed decreases with the number of observations. We can see in Figure 4.5 that our proposed algorithm outperforms coordinate descent and Lars in this setting as well.

4.4.2 Selection of the regularization parameter

We have supposed until now a pre-determined regularization schedule, an assumption that is not practical. The amount of regularization depends indeed on the variance of the noise present in the data which is not known a priori. It is therefore not obvious how to determine the amount of regularization. We write $\mu_n = n\lambda_n$ such that λ_n is

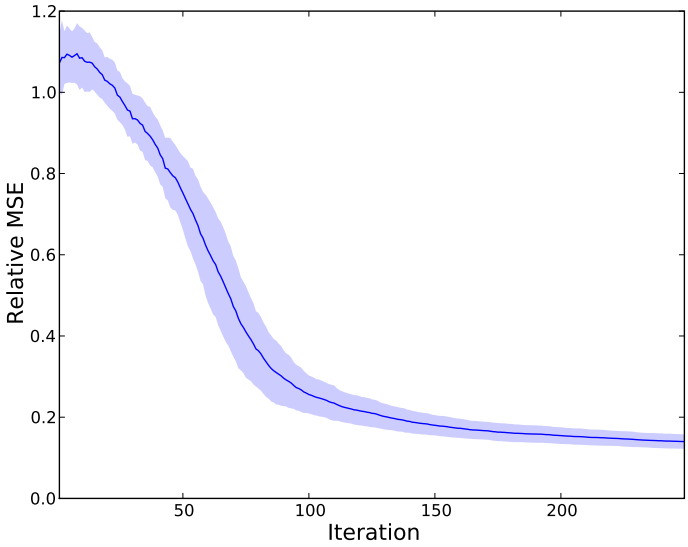


Figure 4.2: On the x-axis of the plots are the iterations of the algorithm, where at each iteration we receive a new measurement. We show the evolution of the reconstruction error $\|\theta^{(n)} - \theta_0\|_2 / \|\theta_0\|_2$. The simulation is repeated 100 times and shaded areas represent one standard deviation.

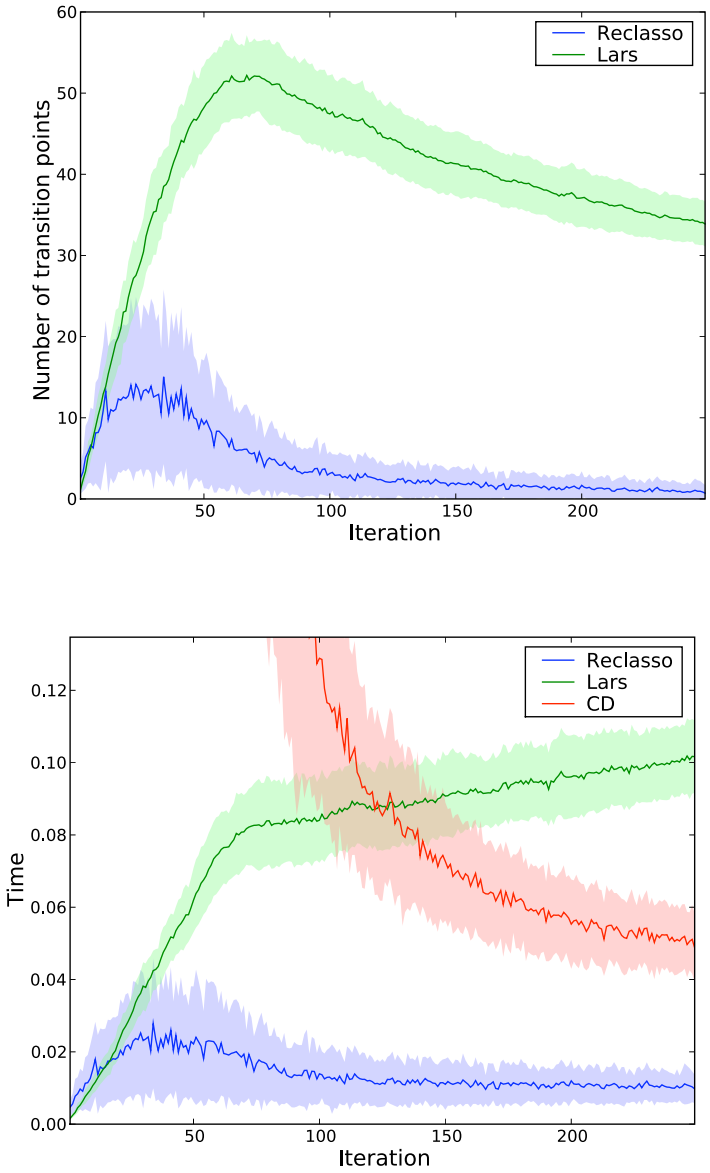


Figure 4.3: Compressive sensing results. **Top** Comparison of the number of transition points for Lars and RecLasso. **Bottom** Timing comparison for the three algorithms.

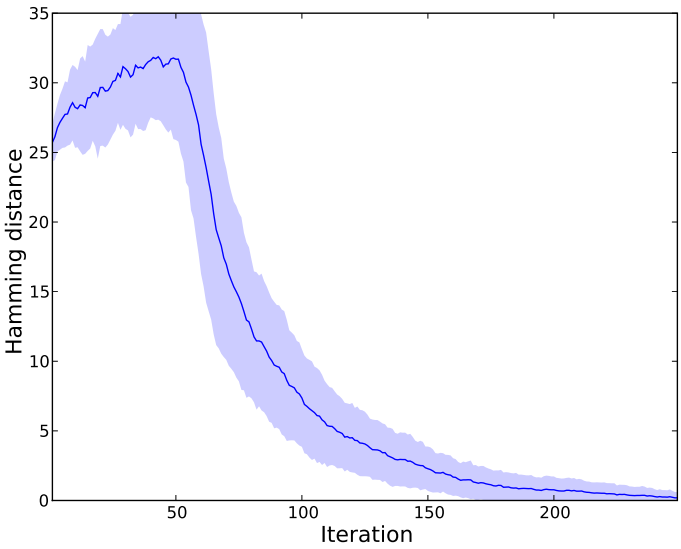


Figure 4.4: Evolution of the hamming distance $H(\theta^{(n)}, \theta_0)$.

the weighting factor between the average mean-squared error and the ℓ_1-norm. We propose an algorithm that selects λ_n in a data-driven manner. The problem with n observations is given by

$$\theta(\lambda) = \arg\min_{\theta} \ \frac{1}{2n} \sum_{i=1}^{n} (x_i^T \theta - y_i)^2 + \lambda \|\theta\|_1.$$

We have seen previously that $\theta(\lambda)$ is piecewise linear, and we can therefore compute its gradient unless λ is a transition point. Let $err(\lambda) = (x_{n+1}^T \theta(\lambda) - y_{n+1})^2$ be the error on the new observation. We propose the following update rule to select λ_{n+1}

$$\begin{aligned} \log \lambda_{n+1} &= \log \lambda_n - \eta \frac{\partial err}{\partial \log \lambda}(\lambda_n) \\ \Rightarrow \lambda_{n+1} &= \lambda_n \times exp\Big\{ 2n\lambda_n \eta x_{n+1,1}^T (X_1^T X_1)^{-1} v_1 (x_{n+1}^T \theta_1 - y_{n+1}) \Big\}, \end{aligned}$$

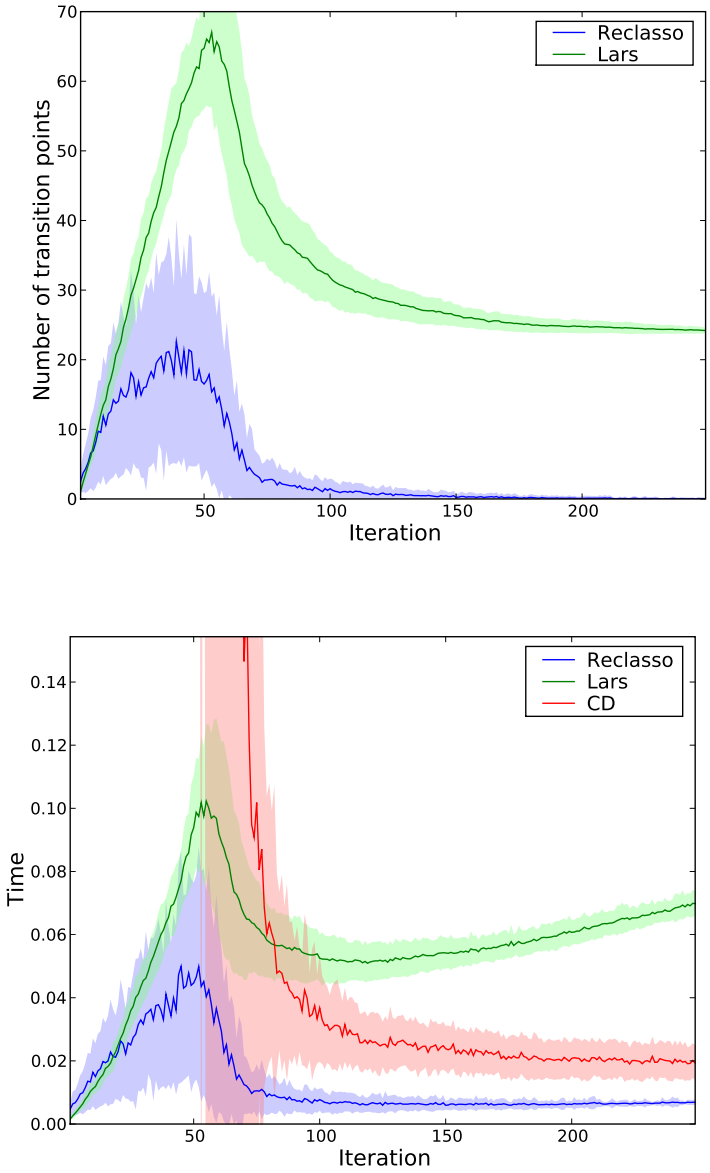


Figure 4.5: Model recovery results. **Top** Comparison of the number of transition points for Lars and RecLasso. **Bottom** Timing comparison for the three algorithms.

where the solution after n observations corresponding to the regularization parameter λ_n is given by $(\theta_1^T, 0^T)$, and $v_1 = \text{sgn}(\theta_1)$. We therefore use the new observation as a test set, which allows us to update the regularization parameter before introducing the new observation by varying t from 0 to 1. We perform the update in the log domain to ensure that λ_n is always positive. We performed simulations using the same experimental setup as in Section 4.4.1 and using $\eta = .01$. We show in Figure 4.6 a representative example where λ converges. We compared this value to the one we would obtain if we had a training and a test set with 250 observations each such that we could fit the model on the training set for various values of λ, and see which one gives the smallest prediction error on the test set. We obtain a very similar result, and understanding the convergence properties of our proposed update rule for the regularization parameter is the object of current research.

4.4.3 Leave-one-out cross-validation

We suppose in this Section that we have access to a dataset $(y_i, x_i)_{i=1\dots n}$ and that $\mu_n = n\lambda$. The parameter λ is tied to the amount of noise in the data which we do not know a priori. A standard approach to select this parameter is leave-one-out cross-validation. For a range of values of λ, we use $n-1$ data points to solve the Lasso with regularization parameter $(n-1)\lambda$ and then compute the prediction error on the data point that was left out. This is repeated n times such that each data point serves as the test set. Hence the best value for λ is the one that leads to the smallest mean prediction error.

Our proposed algorithm can be adapted to the case where we wish to update the solution of the Lasso after a data point is removed. To do so, we compute the first homotopy by varying the regularization parameter from $n\lambda$ to $(n-1)\lambda$. We then compute the second homotopy by varying t from 1 to 0 which has the effect of

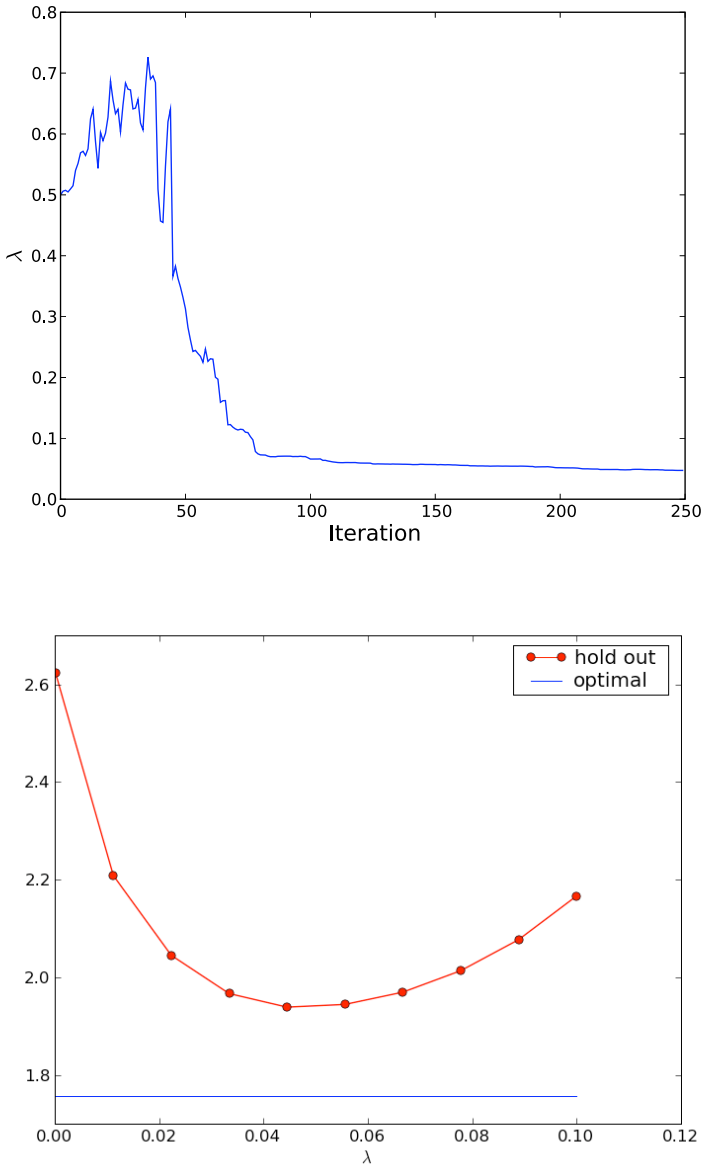


Figure 4.6: **Top** Evolution of the regularization parameter when using our proposed update rule. **Bottom** Regularization parameter selected using a hold-out set. On the x-axis are a range of λ values, and the generalization error is on the y-axis. The optimal λ is $.044$, which is very similar to the value that our learning algorithm converges to.

removing the data point that will be used for testing. As the algorithm is very similar to the one we proposed in Section 4.3.2 we omit the derivation. We sample a model with $n = 32$ and $m = 32$. The vector θ_0 used to generate the data has 8 non-zero elements. We add Gaussian noise of variance 0.2 to the observations, and select λ for a range of 10 values. We show in Figure 4.7 the histogram of the number of transition points for our algorithm when solving the Lasso with $n-1$ data points (we solve this problem $10 \times n$ times). Note that in the majority cases there are very few transition points, which makes our approach very efficient in this setting.

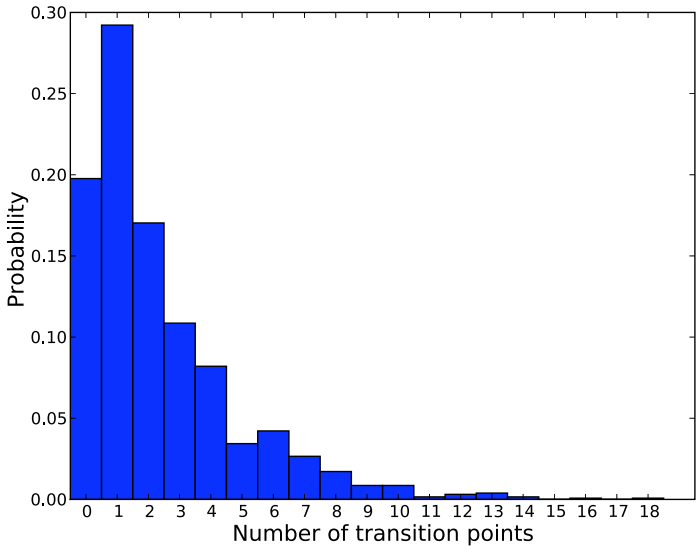


Figure 4.7: Histogram of the number of transition points when removing an observation.

4.5 Conclusion

We have presented an algorithm to solve ℓ_1-penalized least-square regression with online observations. We use the current solution as a "warm-start" and introduce an optimization problem that allows us to compute an homotopy from the current

solution to the solution after observing a new data point. The algorithm is particularly efficient if the active set does not change much, and we show a computational advantage as compared to Lars and Coordinate Descent with warm-start for applications such as compressive sensing with sequential observations and leave-one-out cross-validation. We have also proposed an algorithm to automatically select the regularization parameter where each new measurement is used as a test set.

Chapter 5

Conclusion

Exploiting the sparse structure of natural images is at the heart of most theories about the visual system and the ability to infer sparse solutions is central to solve inverse problems in image processing and computer vision. Learning algorithms seeking sparse solutions are also increasingly popular in the machine learning and statistics communities as they are easier to interpret and avoid over-fitting. We have seen that a classic example of an optimization problem used in these fields is the ℓ_1-regularized least-square problem known as Basis Pursuit Denoising or Lasso, for which many efficient algorithms have been proposed. However, natural signals that are sparse often have more structure that is not accounted for when regularizing using the ℓ_1 norm, which amounts to assuming independence among the basis functions coefficients. In this thesis, I have proposed richer models where the statistical dependencies among the basis functions coefficients are modeled.

In Chapter 2, I modeled the distribution of the basis function activation patterns using an Ising model where a pairwise coupling term captures the dependencies among the basis function coefficients. When adapted to a collection of natural images, these coupling terms converge to a solution consisting of a combination of facilitatory and

inhibitory interactions among neighboring basis functions, and are consistent with physiological data. Furthermore, the representations inferred using the proposed prior have greater sparsity than those inferred using the factorial Laplacian prior.

I introduced in Chapter 3 a class of probability distributions called the Laplace Scale Mixture. A random variable having such a distribution can be written as the product of random variable having a Laplace distribution with scale 1, and a positive random variable called the multiplier. I developed sparse coding models where the basis function coefficients have Laplace Scale Mixture priors. Inference in such models can be performed by solving a sequence of least-square problems regularized by a weighted sum of the coefficients' absolute values, where the weights are updated at each step. The updates are particularly simple when the multipliers have independent Gamma priors. In the case where the distribution of the multiplier variables is non-factorial, I proposed an update that takes a form of divisive normalization, which is thought to be an important operation performed by the human visual system. This model shows increased performance in compressive sensing recovery when applied to signals whose sparsity patterns are clustered.

Finally, I presented in Chapter 4 an efficient algorithm to solve the Lasso with on-line observations. I introduced an optimization problem that makes it possible to compute an homotopy from the current solution to the solution after observing a new data point. The algorithm is particularly efficient if the active set does not change much, and I showed a computational advantage as compared to Lars and Coordinate Descent with warm-start for applications such as compressive sensing with sequential observations and leave-one-out cross-validation. I also proposed an algorithm to automatically select the regularization parameter where each new measurement is used as a test set.

It is essential to have a good signal model when solving inverse problems with real-world signals such as natural images. The image priors I have proposed do not

capture all the structure in natural images, and it is important to continue developing richer models. A key challenge with generative models is the problem of inference and learning. In the algorithms I proposed, the computational complexity of learning is governed by the computational complexity of inference. Hence, learning is not tractable if the inference algorithm is not efficient. In Section 2.3.3, I proposed a variational approximation relying on the MAP estimate for learning the parameters of the model. Though I showed empirically in Section 2.4 that this variational approximation is able to recover the parameters of the model under some conditions, it may not be adequate when applied to learning in more complex models. Also, the generative models I have considered assume that the signals are formed by a *linear* superposition of features. This assumption is clearly wrong for natural images where occlusion plays a crucial role. It is therefore challenging to develop models that are rich enough to capture the structure of real-world signals such as natural images, in which inference is efficient, and learning is tractable.

Bibliography

[Ackley *et al.*, 1985] D.H. Ackley, G.E. Hinton, and T.J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

[Asif and Romberg, 2008] M.S. Asif and J. Romberg. Streaming measurements in compressive sensing: L1 filtering. In *Proc. of 42nd Asilomar Conference on Signals, Systems and Computers*, October 2008.

[Attneave, 1954] F. Attneave. Some informational aspects of visual perception. *Psychol Rev*, 61(3):183–193, May 1954.

[Bach and Jordan, 2004] F.R. Bach and M.I. Jordan. Beyond independent components: trees and clusters. *J. Mach. Learn. Res.*, 4(7-8):1205–1233, 2004.

[Baraniuk *et al.*, 2008] R.G. Baraniuk, V. Cevher, M.F. Duarte, and C. Hegde. Model-based compressive sensing. *Preprint*, August 2008.

[Barlow, 1961] H.B. Barlow. Possible principles underlying the transformation of sensory messages. *Sensory Communications*, pages 217–234, 1961.

[Bell and Sejnowski, 1997] A.J. Bell and T.J. Sejnowski. The 'independent components' of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997.

[Ben-Shahar and Zucker, 2004] O. Ben-Shahar and S. Zucker. Geometrical computations explain projection patterns of long-range horizontal connections in visual cortex. *Neural Comput*, 16(3):445–476, March 2004.

[Bosking *et al.*, 1997] W. Bosking, Y. Zhang, B. Schofield, and D. Fitzpatrick. Orientation selectivity and the arrangement of horizontal connections in the tree shrew striate cortex. *J. Neuroscience*, 17(6):2112–2127, 1997.

[Boyd and Vandenberghe, 2004] S. Boyd and L. Vandenberghe. Convex optimization. *Cambridge Univ. Press*, 2004.

[Candès *et al.*, 2008] E.J. Candès, M.B. Wakin, and S.P. Boyd. Enhancing sparsity by reweighted l1 minimization. *J. Fourier Anal. Appl.*, to appear, 2008.

[Candès, 2006] E. Candès. Compressive sampling. *Proceedings of the International Congress of Mathematicians*, 2006.

[Cevher *et al.*, 2008] V. Cevher, , M. F. Duarte, C. Hegde, and R. G. Baraniuk. Sparse signal recovery using markov random fields. In *Advances in Neural Computation Systems (NIPS)*, Vancouver, B.C., Canada, 2008.

[Cevher *et al.*, 2009] V. Cevher, P. Indyk, C. Hedge, and R.G. Baraniuk. Recovery of clustered sparse signals from compressive measurements. 2009.

[Chen *et al.*, 1999] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.

[Daubechies *et al.*, 2004] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57:1413–1541, 2004.

[Dempster *et al.*, 1977] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[Doi *et al.*, 2003] E. Doi, T. Inui, T.-W. Lee, T. Wachtler, and T.J. Sejnowski. Spatiochromatic receptive field properties derived from information-theoretic analyses of cone mosaic responses to natural scenes. *Neural Computation*, 15:397–417, 2003.

[Donoho and Tsaig, 2006] D. Donoho and Y. Tsaig. Fast solution of l 1-norm minimization problems when the solution may be sparse. *preprint*, 2006.

[Donoho, 2006a] D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[Donoho, 2006b] D.L. Donoho. For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59(6):797–829, 2006.

[Drori and Donoho, 2006] I. Drori and D.L. Donoho. Solution of ℓ_1 minimization problems by lars/homotopy methods. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

[Efron *et al.*, 2004] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

[Elad and Aharon, 2006] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, Dec. 2006.

[Eliasmith and Anderson, 2003] C. Eliasmith and C.H. Anderson. *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. Bradford Books, 2003.

[Fergus *et al.*, 2006] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, and W.T. Freeman. Removing camera shake from a single photograph. In *ACM Trans. on Graphics (Proc. SIGGRAPH 2006)*, 2006.

[Field, 1987] D.J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *J Opt Soc Am A*, 4(12):2379–2394, December 1987.

[Field, 1994] D.J. Field. What is the goal of sensory coding? *Neural computation*, 6(4):559–601, 1994.

[Figueiredo and Nowak, 2005] M. Figueiredo and R. Nowak. A bound optimization approach to wavelet-based image deconvolution. In *Proceedings of the International Conference on Image Processing (ICIP)*, Genova, Italy, September 2005.

[Figueiredo *et al.*, 2007] M. Figueiredo, R. Nowak, and S. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):586–597, 2007.

[Figueras and Simoncelli, 2007] R.M. Figueras and E.P. Simoncelli. Statistically driven sparse image representation. In *Proc 14th IEEE Int'l Conf on Image Proc*, volume 6, pages 29–32, September 2007.

[Fitzpatrick, 1996] D. Fitzpatrick. The functional organization of local circuits in visual cortex: insights from the study of tree shrew striate cortex. *Cerebral Cortex*, 6:329–41, 1996.

[Friedman *et al.*, 2007] J. Friedman, T. Hastie, H. Hoefling, and R. Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[Garrigues and El Ghaoui, 2008] P.J. Garrigues and L. El Ghaoui. An homotopy algorithm for the lasso with online observations. In *Advances in Neural Computation Systems (NIPS)*, Vancouver, Canada, 2008.

[Garrigues and Olshausen, 2007] P.J. Garrigues and B.A. Olshausen. Learning horizontal connections in a sparse coding model of natural images. In *Advances in Neural Computation Systems (NIPS)*, Vancouver, Canada, 2007.

[Hinton *et al.*, 2005] G. Hinton, S. Osindero, and K. Bao. Learning causally linked markov random fields. *Artificial Intelligence and Statistics*, Barbados, 2005.

[Hyvärinen and Hoyer, 2000] A. Hyvärinen and P.O. Hoyer. Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Comp.*, 12(7):1705–1720, July 2000.

[Hyvärinen *et al.*, 2001] A. Hyvärinen, P.O. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Computation*, 13(7):1527–1558, 2001.

[Hyvärinen *et al.*, 2003] A. Hyvärinen, J. Hurri, and J. Väyrynen. Bubbles: a unifying framework for low-level statistical properties of natural image sequences. *J. Opt. Soc. Am. A*, 20(7):1237–1252, July 2003.

[Hyvärinen, 1999] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[Jacob *et al.*, 2009] L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In *International Conference on Machine Learning (ICML)*, 2009.

[Jenatton *et al.*, 2009] R. Jenatton, J.-Y. Audibert, and F.R. Bach. Structured variable selection with sparsity-inducing norms. Research Report, 2009.

[Jordan *et al.*, 1999] M.I. Jordan, Z. Ghahramani, T. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Learning in Graphical Models*, Cambridge, MA: MIT Press, 1999.

[Karklin and Lewicki, 2005] Y. Karklin and M.S. Lewicki. A hierarchical bayesian model for learning nonlinear statistical regularities in nonstationary natural signals. *Neural Computation*, 17(2):397–423, February 2005.

[Kim *et al.*, 2007] S-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized least squares. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):606–617, 2007.

[Lee *et al.*, 2007] H. Lee, A. Battle, R. Raina, and A.Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems 19*, pages 801–808. MIT Press, Cambridge, MA, 2007.

[Levin and Weiss, 2007] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(9):1647–1654, 2007.

[Lyu and Simoncelli, 2006] S. Lyu and E. P. Simoncelli. Statistical modeling of images with fields of gaussian scale mixtures. In *Advances in Neural Computation Systems (NIPS)*, Vancouver, Canada, 2006.

[Malach *et al.*, 1993] R. Malach, Y. Amir, M. Harel, and A. Grinvald. Relationship between intrinsic connections and functional architecture revealed by optical imaging and in vivo targeted biocytin injections in primate striate cortex. *Proc. Natl. Acad. Sci. U.S.A.*, 82:935–939, 1993.

[Malioutov *et al.*, 2005] D.M. Malioutov, M. Cetin, and A.S. Willsky. Homotopy continuation for sparse signal representation. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, PA, March 2005.

[Malioutov *et al.*, 2008] D.M. Malioutov, S. Sanghavi, and A.S. Willsky. Compressed sensing with sequential observations. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, NV, March 2008.

[Mallat, 1989] S. G. Mallat. A theory for multiresolution signal decomposition - the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, July 1989.

[Olshausen and Field, 1996] B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, June 1996.

[Olshausen and Field, 1997] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: a strategy employed by v1? *Vision Res*, 37(23):3311–3325, December 1997.

[Olshausen and Millman, 2000] B.A. Olshausen and K.J. Millman. Learning sparse codes with a mixture-of-gaussians prior. *Advances in Neural Information Processing Systems, 12*, 2000.

[Osborne *et al.*, 2000] M.R. Osborne, B. Presnell, and B.A. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20:389–404, 2000.

[Osborne, 1992] M.R. Osborne. An effective method for computing regression quantiles. *IMA Journal of Numerical Analysis*, Jan 1992.

[Portilla *et al.*, 2003] J. Portilla, V. Strela, M.J Wainwright, and E.P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, 2003.

[Roth and Black, 2009] S. Roth and M.J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, April 2009.

[Rozell *et al.*, 2007] C.J. Rozell, D.H. Johnson, R.G. Baraniuk, and B.A. Olshausen. Locally competitive algorithms for sparse approximation. In *Proceedings of the International Conference on Image Processing (ICIP)*, San Antonio, TX, September 2007.

[Rozell *et al.*, 2008] C.J. Rozell, D.H Johnson, R.G. Baraniuk, and B.A. Olshausen. Sparse coding via thresholding and local competition in neural circuits. *Neural Computation*, 20(10):2526–2563, October 2008.

[Schneidman *et al.*, 2006] E. Schneidman, M.J. Berry, R. Segev, and W. Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, April 2006.

[Simoncelli and Adelson, 1996] E.P. Simoncelli and E.H. Adelson. Noise removal via bayesian wavelet coring. In *Third Int'l Conf on Image Proc*, volume I, pages 379–382, Lausanne, 1996. IEEE Sig Proc Society.

[Simoncelli *et al.*, 1992] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger. Shiftable multiscale transforms. *Information Theory, IEEE Transactions on*, 38(2):587–607, 1992.

[Sra and Tropp, 2006] S. Sra and J.A. Tropp. Row-action methods for compressed sensing. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

[Tibshirani, 1996] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996.

[Tropp, 2004] J.A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

[Tropp, 2006] J.A. Tropp. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006.

[Tsaig and Donoho, 2006] Y. Tsaig and D.L. Donoho. Extensions of compressed sensing. *Signal Processing*, 86(3):549–571, 2006.

[Wainwright *et al.*, 2001a] M.J. Wainwright, O. Schwartz, and E.P. Simoncelli. Natural image statistics and divisive normalization: Modeling nonlinearity and adaptation in cortical neurons. In R. Rao, B.A. Olshausen, and M.S. Lewicki, editors, *Statistical Theories of the Brain*. MIT Press, 2001.

[Wainwright *et al.*, 2001b] M.J. Wainwright, E.P. Simoncelli, and A.S. Willsky. Random cascades on wavelet trees and their use in modeling and analyzing natural imagery. *Applied and Computational Harmonic Analysis*, 11(1), July 2001.

[Wang *et al.*, 2005] Z. Wang, A.C. Bovik, and E.P. Simoncelli. Structural approaches to image quality assessment. In Alan Bovik, editor, *Handbook of Image and Video Processing*, chapter 8.3, pages 961–974. Academic Press, May 2005. 2nd edition.

[Wipf and Nagarajan, 2008] D. Wipf and S. Nagarajan. A new view of automatic relevance determination. In *Advances in Neural Information Processing Systems 20*, 2008.

[Yuan and Lin, 2006] 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, February 2006.

[Zetzsche *et al.*, 1999] C. Zetzsche, G. Krieger, and B. Wegmann. The atoms of vision: Cartesian or polar? *Journal of the Optical Society of America A*, 16(7):1554–1565, 1999.

[Zhao and Yu, 2006] P. Zhao and B. Yu. On model selection consistency of lasso. *The Journal of Machine Learning Research*, pages 2541–2563, 2006.

[Zhaoping, 2005] L. Zhaoping. Border ownership from intracortical interactions in visual area v2. *Neuron*, 47:143–153, 2005.