Sparse Estimation for Image and Vision Processing

Julien Mairal

Inria, Grenoble

Yerevan, 2018

Outline

1. A short introduction to parsimony
2. Discovering the structure of natural images
3. Sparse models for image processing
4. Optimization for sparse estimation
Part I: A Short Introduction to Parcimony
A short introduction to parsimony

- Early thoughts
- Sparsity in the statistics literature from the 60’s and 70’s
- Wavelet thresholding in signal processing from 90’s
- The modern parsimony and the $\ell_1$-norm
- Structured sparsity
- Compressed sensing and sparse recovery

Discovering the structure of natural images

Sparse models for image processing

Optimization for sparse estimation
Early thoughts

(a) Dorothy Wrinch 1894–1980
(b) Harold Jeffreys 1891–1989

The existence of simple laws is, then, apparently, to be regarded as a quality of nature; and accordingly we may infer that it is justifiable to prefer a simple law to a more complex one that fits our observations slightly better.

[Wrinch and Jeffreys, 1921]. Philosophical Magazine Series.
Historical overview of parsimony

- 14th century: Ockham’s razor;
- 1921: Wrinch and Jeffreys’ simplicity principle;
- 1952: Markowitz’s portfolio selection;
- 60 and 70’s: best subset selection in statistics;
- 70’s: use of the $\ell_1$-norm for signal recovery in geophysics;
- 90’s: wavelet thresholding in signal processing;
- 1996: Olshausen and Field’s dictionary learning;
- 1996–1999: Lasso (statistics) and basis pursuit (signal processing);
- 2006: compressed sensing (signal processing) and Lasso consistency (statistics);
- 2006–now: applications of dictionary learning in various scientific fields such as image processing and computer vision.
Sparsity in the statistics literature from the 60’s and 70’s

Given some observed data points \( z_1, \ldots, z_n \) that are assumed to be independent samples from a statistical model with parameters \( \theta \) in \( \mathbb{R}^p \), maximum likelihood estimation (MLE) consists of minimizing

\[
\min_{\theta \in \mathbb{R}^p} \left[ \mathcal{L}(\theta) \triangleq - \sum_{i=1}^{n} \log P_{\theta}(z_i) \right].
\]

Example: ordinary least square

Observations \( z_i = (y_i, x_i) \), with \( y_i \) in \( \mathbb{R} \). Linear model: \( y_i = x_i^\top \theta + \varepsilon_i \), with \( \varepsilon_i \sim \mathcal{N}(0, 1) \).

\[
\min_{\theta \in \mathbb{R}^p} \sum_{i=1}^{n} \frac{1}{2} \left( y_i - x_i^\top \theta \right)^2.
\]
Sparsity in the statistics literature from the 60’s and 70’s

Given some observed data points $z_1, \ldots, z_n$ that are assumed to be independent samples from a statistical model with parameters $\theta$ in $\mathbb{R}^p$, maximum likelihood estimation (MLE) consists of minimizing

$$\min_{\theta \in \mathbb{R}^p} \left[ \mathcal{L}(\theta) \triangleq -\sum_{i=1}^{n} \log P_{\theta}(z_i) \right].$$

Motivation for finding a sparse solution:

- removing irrelevant variables from the model;
- obtaining an easier interpretation;
- preventing overfitting;
Sparsity in the statistics literature from the 60’s and 70’s

Given some observed data points $z_1, \ldots, z_n$ that are assumed to be independent samples from a statistical model with parameters $\theta$ in $\mathbb{R}^p$, maximum likelihood estimation (MLE) consists of minimizing

$$
\min_{\theta \in \mathbb{R}^p} \left[ \mathcal{L}(\theta) \triangleq - \sum_{i=1}^{n} \log P_{\theta}(z_i) \right].
$$

Two questions:

1. how to choose $k$?
2. how to find the best subset of $k$ variables?
Sparsity in the statistics literature from the 60’s and 70’s

**How to choose $k$?**

- Mallows’s $C_p$ statistics [Mallows, 1964, 1966];
- Akaike information criterion (AIC) [Akaike, 1973];
- Bayesian information criterion (BIC) [Schwarz, 1978];
- Minimum description length (MDL) [Rissanen, 1978].

These approaches lead to penalized problems

$$
\min_{\theta \in \mathbb{R}^p} \mathcal{L}(\theta) + \lambda \|\theta\|_0,
$$

with different choices of $\lambda$ depending on the chosen criterion.
Sparsity in the statistics literature from the 60’s and 70’s

How to solve the best $k$-subset selection problem?

Unfortunately...
...the problem is NP-hard [Natarajan, 1995].

Two strategies

- **combinatorial exploration** with branch-and-bound techniques [Furnival and Wilson, 1974] $\rightarrow$ **leaps and bounds**, exact algorithm but exponential complexity;
- **greedy approach**: forward selection [Efroymson, 1960] (originally developed for observing intermediate solutions), already contains all the ideas of matching pursuit algorithms.

Wavelet thresholding in signal processing from the 90’s

A wavelet basis represents a set of functions $\varphi_1, \varphi_2$ that are essentially dilated and shifted versions of each other [see Mallat, 2008].

Concept of parsimony with wavelets

When a signal $f$ is “smooth”, it is close to an expansion $\sum_i \alpha_i \varphi_i$ where only a few coefficients $\alpha_i$ are non-zero.
Wavelet thresholding in signal processing from the 90’s

Wavelets where the topic of a long quest for representing natural images
- 2D-Gabors [Daugman, 1985];
- steerable wavelets [Simoncelli et al., 1992];
- curvelets [Candès and Donoho, 2002];
- contourlets [Do and Vertterli, 2003];
- bandlets [Le Pennec and Mallat, 2005];
- ⋆-lets (joke).

(a) 2D Gabor filter.  (b) With shifted phase.  (c) With rotation.
Wavelet thresholding in signal processing from 90’s

The theory of wavelets is well developed for continuous signals, e.g., in $L^2(\mathbb{R})$, but also for discrete signals $x$ in $\mathbb{R}^n$. 
Wavelet thresholding in signal processing from 90’s

Given an orthogonal wavelet basis $D = [d_1, \ldots, d_n]$ in $\mathbb{R}^{n \times n}$, the wavelet decomposition of $x$ in $\mathbb{R}^n$ is simply

$$\beta = D^T x \quad \text{and we have} \quad x = D \beta.$$ 

The $k$-sparse approximation problem

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \| x - D \alpha \|_2^2 \quad \text{s.t.} \quad \| \alpha \|_0 \leq k,$$

is not NP-hard here: since $D$ is orthogonal, it is equivalent to

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \| \beta - \alpha \|_2^2 \quad \text{s.t.} \quad \| \alpha \|_0 \leq k.$$
Wavelet thresholding in signal processing from 90’s

Given an orthogonal wavelet basis $D = [d_1, \ldots, d_n]$ in $\mathbb{R}^{n \times n}$, the wavelet decomposition of $x$ in $\mathbb{R}^n$ is simply

$$\beta = D^\top x \quad \text{and we have } \ x = D \beta.$$ 

The $k$-sparse approximation problem

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \|x - D\alpha\|_2^2 \quad \text{s.t. } \|\alpha\|_0 \leq k,$$

The solution is obtained by hard-thresholding:

$$\alpha^{ht}[j] = \delta_{|\beta[j]| \geq \mu} \beta[j] = \begin{cases} \beta[j] & \text{if } |\beta[j]| \geq \mu \\ 0 & \text{otherwise} \end{cases},$$

where $\mu$ the $k$-th largest value among the set $\{|\beta[1]|, \ldots, |\beta[p]|\}$. 
Wavelet thresholding in signal processing, 90’s

Another key operator introduced by Donoho and Johnstone [1994] is the **soft-thresholding** operator:

$$\alpha_{st}[j] \triangleq \text{sign}(\beta[j]) \max(|\beta[j]| - \lambda, 0) = \begin{cases} 
\beta[j] - \lambda & \text{if } \beta[j] \geq \lambda \\
\beta[j] + \lambda & \text{if } \beta[j] \leq -\lambda \\
0 & \text{otherwise}
\end{cases},$$

where $\lambda$ is a parameter playing the same role as $\mu$ previously.

With $\beta \triangleq D^\top x$ and $D$ orthogonal, it provides the solution of the following sparse reconstruction problem:

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \|x - D\alpha\|_2^2 + \lambda \|\alpha\|_1,$$

which will be of high importance later.
Wavelet thresholding in signal processing, 90’s

(d) Soft-thresholding operator,
\[ \alpha^{st} = \text{sign}(\beta) \max(|\beta| - \lambda, 0). \]

(e) Hard-thresholding operator
\[ \alpha^{ht} = \delta_{|\beta| \geq \mu} \beta. \]

**Figure:** Soft- and hard-thresholding operators, which are commonly used for signal estimation with orthogonal wavelet basis.
Wavelet thresholding in signal processing, 90’s

Various work tried to exploit the structure of wavelet coefficients.

Figure: Illustration of a wavelet tree with four scales for one-dimensional signals. We also illustrate the zero-tree coding scheme [Shapiro, 1993].
Wavelet thresholding in signal processing, 90’s

To model spatial relations, it is possible to define some (non-overlapping) groups $G$ of wavelet coefficients, and define a group soft-thresholding operator [Hall et al., 1999, Cai, 1999]. For every group $g$ in $G$,

$$\alpha^{gt}[g] \triangleq \begin{cases} 
(1 - \frac{\lambda}{\|\beta[g]\|_2}) \beta[g] & \text{if } \|\beta[g]\|_2 \geq \lambda \\
0 & \text{otherwise}
\end{cases},$$

where $\beta[g]$ is the vector of size $|g|$ recording the entries of $\beta$ in $g$.

With $\beta \triangleq D^\top x$ and $D$ orthogonal, it is in fact the solution of

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2}\|x - D\alpha\|_2^2 + \lambda \sum_{g \in G} \|\alpha[g]\|_2,$$

which will be of interest later in the lecture.
The modern parsimony and the $\ell_1$-norm
Sparse linear models in signal processing

Let $x$ in $\mathbb{R}^n$ be a signal.

Let $D = [d_1, \ldots, d_p] \in \mathbb{R}^{n \times p}$ be a set of elementary signals.
We call it dictionary.

$D$ is “adapted” to $x$ if it can represent it with a few elements—that is, there exists a sparse vector $\alpha$ in $\mathbb{R}^p$ such that $x \approx D\alpha$. We call $\alpha$ the sparse code.

$$
\begin{pmatrix} x \end{pmatrix} \approx \begin{pmatrix} d_1 & d_2 & \cdots & d_p \end{pmatrix}
$$

$\alpha \in \mathbb{R}^p$, sparse
The modern parsimony and the $\ell_1$-norm

Sparse linear models: machine learning/statistics point of view

Let $(y_i, x_i)_{i=1}^n$ be a training set, where the vectors $x_i$ are in $\mathbb{R}^p$ and are called features. The scalars $y_i$ are in

- $\{-1, +1\}$ for **binary** classification problems.
- $\mathbb{R}$ for **regression** problems.

We assume there exists a relation $y \approx \beta^\top x$, and solve

$$
\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n L(y_i, \beta^\top x_i) + \lambda \psi(\beta).
$$

Here, $L$ is the empirical risk and $\lambda \psi(\beta)$ is the regularization term.
The modern parsimony and the $\ell_1$-norm

Sparse linear models: machine learning/statistics point of view

A few examples:

**Ridge regression:**
$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} (y_i - \beta^T x_i)^2 + \lambda \|\beta\|_2^2.$$  

**Linear SVM:**
$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - y_i \beta^T x_i) + \lambda \|\beta\|_2^2.$$  

**Logistic regression:**
$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \log \left(1 + e^{-y_i \beta^T x_i}\right) + \lambda \|\beta\|_2^2.$$
The modern parsimony and the $\ell_1$-norm

Sparse linear models: machine learning/statistics point of view

A few examples:

Ridge regression:
\[
\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} (y_i - \beta^T x_i)^2 + \lambda \|\beta\|_2^2.
\]

Linear SVM:
\[
\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - y_i \beta^T x_i) + \lambda \|\beta\|_2^2.
\]

Logistic regression:
\[
\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \log \left( 1 + e^{-y_i \beta^T x_i} \right) + \lambda \|\beta\|_2^2.
\]

The squared $\ell_2$-norm induces “smoothness” in $\beta$. When one knows in advance that $\beta$ should be sparse, one should use a sparsity-inducing regularization such as the $\ell_1$-norm. [Chen et al., 1999, Tibshirani, 1996]
The modern parsimony and the $\ell_1$-norm

Originally used to induce sparsity in geophysics [Claerbout and Muir, 1973, Taylor et al., 1979], the $\ell_1$-norm became popular in statistics with the **Lasso** [Tibshirani, 1996] and in signal processing with the **Basis pursuit** [Chen et al., 1999].

Three “equivalent” formulations

1. \[
\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \| x - D\alpha \|_2^2 + \lambda \| \alpha \|_1; 
\]

2. \[
\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \| x - D\alpha \|_2^2 \quad \text{s.t.} \quad \| \alpha \|_1 \leq \mu; 
\]

3. \[
\min_{\alpha \in \mathbb{R}^p} \| \alpha \|_1 \quad \text{s.t.} \quad \| x - D\alpha \|_2^2 \leq \varepsilon. 
\]
The modern parsimony and the $\ell_1$-norm

And some variants...

For noiseless problems

$$\min_{\alpha \in \mathbb{R}^p} \|\alpha\|_1 \quad \text{s.t.} \quad x = D\alpha.$$ 

Beyond least squares

$$\min_{\alpha \in \mathbb{R}^p} f(\alpha) + \lambda\|\alpha\|_1,$$

where $f : \mathbb{R}^p \to \mathbb{R}$ is convex.
The modern parsimony and the $\ell_1$-norm

And some variants...

For noiseless problems

$$\min_{\alpha \in \mathbb{R}^p} \| \alpha \|_1 \quad \text{s.t.} \quad x = D\alpha.$$ 

Beyond least squares

$$\min_{\alpha \in \mathbb{R}^p} f(\alpha) + \lambda \| \alpha \|_1,$$

where $f : \mathbb{R}^p \to \mathbb{R}$ is convex.

An important question remains:

**why does the $\ell_1$-norm induce sparsity?**
The modern parsimony and the $\ell_1$-norm

Why does the $\ell_1$-norm induce sparsity?

Can we get some intuition from the simplest isotropic case?

$$\hat{\alpha}(\lambda) = \arg\min_{\alpha \in \mathbb{R}^p} \frac{1}{2}\|x - \alpha\|_2^2 + \lambda\|\alpha\|_1,$$

or equivalently the Euclidean projection onto the $\ell_1$-ball?

$$\tilde{\alpha}(\mu) = \arg\min_{\alpha \in \mathbb{R}^p} \frac{1}{2}\|x - \alpha\|_2^2 \quad \text{s.t.} \quad \|\alpha\|_1 \leq \mu.$$

“equivalent” means that for all $\lambda > 0$, there exists $\mu \geq 0$ such that $\tilde{\alpha}(\mu) = \hat{\alpha}(\lambda)$. 
The modern parsimony and the $\ell_1$-norm

Why does the $\ell_1$-norm induce sparsity?

Can we get some intuition from the simplest isotropic case?

\[
\hat{\alpha}(\lambda) = \arg\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \|x - \alpha\|_2^2 + \lambda \|\alpha\|_1,
\]

or equivalently the Euclidean projection onto the $\ell_1$-ball?

\[
\tilde{\alpha}(\mu) = \arg\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \|x - \alpha\|_2^2 \quad \text{s.t.} \quad \|\alpha\|_1 \leq \mu.
\]

“equivalent” means that for all $\lambda > 0$, there exists $\mu \geq 0$ such that

$\tilde{\alpha}(\mu) = \hat{\alpha}(\lambda)$.

The relation between $\mu$ and $\lambda$ is unknown a priori.
Why does the $\ell_1$-norm induce sparsity?

Regularizing with the $\ell_1$-norm

The projection onto a convex set is “biased” towards singularities.
Why does the $\ell_1$-norm induce sparsity?

Regularizing with the $\ell_2$-norm

The $\ell_2$-norm is isotropic.
Why does the $\ell_1$-norm induce sparsity?

In 3D. (images produced by G. Obozinski)
Why does the $\ell_1$-norm induce sparsity?

Regularizing with the $\ell_\infty$-norm

The $\ell_\infty$-norm encourages $|\alpha[1]| = |\alpha[2]|$. 

$$\|\alpha\|_\infty \leq \mu$$

$\ell_\infty$-ball
Why does the $\ell_1$-norm induce sparsity?

Analytical point of view: 1D case

\[
\min_{\alpha \in \mathbb{R}} \frac{1}{2} (x - \alpha)^2 + \lambda |\alpha|
\]

Piecewise quadratic function with a kink at zero.

Derivative at $0_+$: $g_+ = -x + \lambda$ and $0_-$. $g_- = -x - \lambda$.

Optimality conditions. $\alpha$ is optimal iff:

- $|\alpha| > 0$ and $(x - \alpha) + \lambda \text{sign}(\alpha) = 0$
- $\alpha = 0$ and $g_+ \geq 0$ and $g_- \leq 0$

The solution is a **soft-thresholding**: 

\[
\alpha^* = \text{sign}(x)(|x| - \lambda)^+.
\]
Why does the $\ell_1$-norm induce sparsity?
Comparison with $\ell_2$-regularization in 1D

$$\psi(\alpha) = \alpha^2$$
$$\psi'(\alpha) = 2\alpha$$

$$\psi_-(\alpha) = -1, \quad \psi_+(\alpha) = 1$$

The gradient of the $\ell_2$-penalty vanishes when $\alpha$ get close to 0. On its differentiable part, the norm of the gradient of the $\ell_1$-norm is constant.
Why does the $\ell_1$-norm induce sparsity?

Physical illustration

$E_1 = 0$

$E_1 = 0$
Why does the \( \ell_1 \)-norm induce sparsity?

Physical illustration

\[
E_1 = \frac{k_1}{2} (x - \alpha)^2
\]

\[
E_2 = \frac{k_2}{2} \alpha^2
\]

\[
E_1 = \frac{k_1}{2} (x - \alpha)^2
\]

\[
E_2 = mg \alpha
\]
Why does the $\ell_1$-norm induce sparsity?

Physical illustration

\[ E_1 = \frac{k_1}{2} (x - \alpha)^2 \]

\[ E_2 = \frac{k_2}{2} \alpha^2 \]

\[ E_1 = \frac{k_1}{2} (x - \alpha)^2 \]

\[ E_2 = mg\alpha \]
Why does the $\ell_1$-norm induce sparsity?

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \| x - D\alpha \|_2^2 + \lambda \| \alpha \|_1.$$
Non-convex sparsity-inducing penalties

Exploiting concave functions with a kink at zero

\[ \psi(\alpha) = \sum_{j=1}^{P} \varphi(|\alpha[j]|). \]

- $\ell_q$-penalty, with $0 < q < 1$: \( \psi(\alpha) \triangleq \sum_{j=1}^{P} |\alpha[j]|^q \), [Frank and Friedman, 1993];
- log penalty, \( \psi(\alpha) \triangleq \sum_{j=1}^{P} \log(|\alpha[j]| + \varepsilon) \), [Candès et al., 2008].

\( \varphi \) is any function that looks like this:

\[ \varphi(|\alpha|) \]
Non-convex sparsity-inducing penalties

(a) $\ell_{0.5}$-ball, 2-D
(b) $\ell_1$-ball, 2-D
(c) $\ell_2$-ball, 2-D

Figure: Open balls in 2-D corresponding to several $\ell_q$-norms and pseudo-norms.
Non-convex sparsity-inducing penalties

\[ \| \alpha \|_q \leq \mu \text{ with } q < 1 \]
Elastic-net

The elastic net introduced by [Zou and Hastie, 2005]

\[ \psi(\alpha) = \|\alpha\|_1 + \gamma \|\alpha\|_2^2, \]

The penalty provides more stable (but less sparse) solutions.

(a) $\ell_1$-ball, 2-D  
(b) elastic-net, 2-D  
(c) $\ell_2$-ball, 2-D
The elastic-net vs other penalties

\[ \| \alpha \|_q \leq \mu \text{ with } q < 1 \]
The elastic-net

vs other penalties

\[ \| \alpha \|_1 \leq \mu \]
The elastic-net
vs other penalties

\[(1 - \gamma)\|\alpha\|_1 + \gamma\|\alpha\|_2^2 \leq \mu\]
The elastic-net
vs other penalties

\[ \| \alpha \|_2^2 \leq \mu \]

\[ \ell_2\text{-ball} \]

\[ \alpha[2] \]

\[ \alpha[1] \]
Total variation and fused Lasso

The anisotropic total variation [Rudin et al., 1992]

$$\psi(\alpha) = \sum_{j=1}^{p-1} |\alpha[j + 1] - \alpha[j]|$$

called fused Lasso in statistics [Tibshirani et al., 2005]. The penalty encourages piecewise constant signals (can be extended to images).

Image borrowed from a talk of J.-P. Vert, representing DNA copy numbers.
Group Lasso and mixed norms

[Turlach et al., 2005, Yuan and Lin, 2006, Zhao et al., 2009]
[Grandvalet and Canu, 1999, Bakin, 1999]

the $\ell_1/\ell_q$-norm:

$$
\psi(\alpha) = \sum_{g \in G} \|\alpha[g]\|_q.
$$

- $\mathcal{G}$ is a partition of $\{1, \ldots, p\}$;
- $q = 2$ or $q = \infty$ in practice;
- can be interpreted as the $\ell_1$-norm of $[\|\alpha[g]\|_q]_{g \in \mathcal{G}}$.

$$
\psi(\alpha) = \|\alpha[\{1, 2\}]\|_2 + |\alpha[3]|.
$$
Spectral sparsity

[Fazel et al., 2001, Srebro et al., 2005]

A natural regularization function for matrices is the rank

\[ \text{rank}(A) \triangleq |\{ j : s_j(A) \neq 0 \}| = \| s(A) \|_0, \]

where \( s_j \) is the \( j \)-th singular value and \( s \) is the spectrum of \( A \).

A successful convex relaxation of the rank is the sum of singular values

\[ \| A \|_* \triangleq \sum_{j=1}^{p} s_j(A) = \| s(A) \|_1, \]

for \( A \) in \( \mathbb{R}^{p \times k} \) with \( k \geq p \).

The resulting function is a norm, called the trace or nuclear norm.
Structured sparsity
images produced by G. Obozinski
Structured sparsity
images produced by G. Obozinski
Structured sparsity
Metabolic network of the budding yeast from Rapaport et al. [2007]
Structured sparsity

Metabolic network of the budding yeast from Rapaport et al. [2007]
Structured sparsity

Warning: Under the name “structured sparsity” appear in fact significantly different formulations!

1. non-convex
   - zero-tree wavelets [Shapiro, 1993];
   - predefined collection of sparsity patterns: [Baraniuk et al., 2010];
   - select a union of groups: [Huang et al., 2009];
   - structure via Markov random fields: [Cehver et al., 2008];

2. convex (norms)
   - tree-structure: [Zhao et al., 2009];
   - select a union of groups: [Jacob et al., 2009];
   - zero-pattern is a union of groups: [Jenatton et al., 2011a];
   - other norms: [Micchelli et al., 2013].
Structured sparsity
Group Lasso with overlapping groups [Jenatton et al., 2011a]

\[ \psi(\alpha) = \sum_{g \in G} \|\alpha[g]\|_q. \]

What happens when the groups overlap?

- the pattern of non-zero variables is an intersection of groups;
- the zero pattern is a union of groups.

\[ \psi(\alpha) = \|\alpha\|_2 + |\alpha[2]| + |\alpha[3]|. \]
Structured sparsity
Group Lasso with overlapping groups [Jenatton et al., 2011a]

Examples of set of groups $\mathcal{G}$

Selection of contiguous patterns on a sequence, $p = 6$.

- $\mathcal{G}$ is the set of blue groups.
- Any union of blue groups set to zero leads to the selection of a contiguous pattern.
Structured sparsity
Group Lasso with overlapping groups [Jenatton et al., 2011a]

Examples of set of groups $\mathcal{G}$

Selection of rectangles on a 2-D grids, $p = 25$.

- $\mathcal{G}$ is the set of blue/green groups (with their not displayed complements).
Structured sparsity

Group Lasso with overlapping groups [Jenatton et al., 2011a]

Examples of set of groups $\mathcal{G}$

Selection of diamond-shaped patterns on a 2-D grids, $p = 25$.

- It is possible to extent such settings to 3-D space, or more complex topologies.
Structured sparsity
Hierarchical norms [Zhao et al., 2009].

A node can be active only if its ancestors are active. The selected patterns are rooted subtrees.
Structured sparsity
Hierarchical norms [Zhao et al., 2009].

(d) Sparsity.
(e) Group sparsity.
(f) Hierarchical sparsity.
Structured sparsity
the non-convex penalty of Huang et al. [2009]

**Warning: different point of view than in the previous three slides**

\[
\varphi(\alpha) \triangleq \min_{\mathcal{J} \subseteq \mathcal{G}} \left\{ \sum_{g \in \mathcal{J}} \eta_g \text{ s.t. } \text{Supp}(\alpha) \subseteq \bigcup_{g \in \mathcal{J}} g \right\}.
\]

- the penalty is **non-convex**.
- is **NP-hard** to compute (set cover problem).
- The pattern of non-zeroes in \( \alpha \) is a **union** of (a few) groups.

It can be rewritten as a boolean linear program:

\[
\varphi(\alpha) = \min_{x \in \{0,1\}^{\lvert \mathcal{G} \rvert}} \left\{ \eta^\top x \text{ s.t. } \mathcal{N} x \geq \text{Supp}(\alpha) \right\}.
\]
Structured sparsity
convex relaxation and the penalty of Jacob et al. [2009]

The penalty of Huang et al. [2009]:

$$\varphi(\alpha) = \min_{x \in \{0,1\}^{|G|}} \left\{ \eta^\top x \text{ s.t. } N x \geq \text{Supp}(\alpha) \right\}.$$ 

A convex LP-relaxation:

$$\psi(\alpha) \triangleq \min_{x \in \mathbb{R}^{|G|}} \left\{ \eta^\top x \text{ s.t. } N x \geq |\alpha| \right\}.$$ 

**Lemma:** $\psi$ is the penalty of Jacob et al. [2009] with the $\ell_\infty$-norm:

$$\psi(\alpha) = \min_{(\beta_g \in \mathbb{R}^p)_{g \in \mathcal{G}}} \sum_{g \in \mathcal{G}} \eta_g \|\beta_g\|_\infty \text{ s.t. } \alpha = \sum_{g \in \mathcal{G}} \beta_g \text{ and } \forall g, \text{ Supp}(\beta_g) \subseteq g,$$
Structured sparsity

The norm of Jacob et al. [2009] in 3D

\[ \psi(\alpha) \text{ with } G = \{\{1, 2\}, \{2, 3\}, \{1, 3\}\}. \]
Sparse recovery: theoretical results for the Lasso

three upcoming slides are inspired from a lecture of G. Obozinski given at Hólar in 2010

Given some observations \((y_i, x_i)_{i=1,...,n}\), with \(y_i\) in \(\mathbb{R}\), assume that the linear model \(y_i = x_i^\top \theta + \varepsilon_i\) is valid, with \(\varepsilon_i \sim \mathcal{N}(0, \sigma^2)\).

Given an estimate \(\hat{\theta}\), three main problems:

1. **Regular consistency**: convergence of estimator \(\hat{\theta}\) to \(\theta\), i.e., \(\|\hat{\theta} - \theta\|_2\) tends to zero when \(n\) tends to \(\infty\);

2. **Model selection consistency**: convergence of the sparsity pattern of \(\hat{\theta}\) to the pattern of \(\theta\);

3. **Efficiency**: convergence of predictions with \(\hat{\theta}\) to the predictions with \(\theta\), i.e., \(\frac{1}{n} \|X\hat{\theta} - X\theta\|_2^2\) tends to zero.
Sparse recovery: theoretical results for the Lasso

Conditions on the design for success

- **Restricted Isometry Property (RIP)**
  \[ \sqrt{1 - \delta_k} \| \theta \| \leq \| X \theta \| \leq \sqrt{1 + \delta_k} \| \theta \| \quad \text{for all} \quad \| \theta \|_0 \leq k. \]

  Subsets of size $k$ of the columns of $X$ should be close to orthogonal.

- **Mutual Incoherence Property (MIP)**
  \[ \max_{i \neq j} | x_i^\top x_j | < \mu. \]

- **Irrepresentable condition (IC)**
  \[ \| Q_J^{-1} Q_J^c \text{sign}(\theta_J) \|_\infty \leq 1 - \gamma \quad \text{with} \quad Q_{JJ'} = X_J^\top X_{J'}. \]

- ** Restricted Eigenvalue condition (RE)**
  \[ \kappa^2(k) = \min_{|J| \leq k} \min_{\Delta} \frac{\Delta^\top Q \Delta}{\| \Delta \|_2^2} > 0 \]
Sparse recovery: theoretical results for the Lasso

Model selection consistency (Lasso)

- Assume $\theta$ sparse and denote $J = \{j : \theta[j] \neq 0\}$ the nonzero pattern
- **Irrepresentable Condition**($\gamma$) [Zhao and Yu, 2006, Wainwright, 2009]

\[
\|Q_{J^c J} Q_{JJ}^{-1} \text{sign}(\theta_J)\|_\infty \leq 1 - \gamma
\]

where $Q = \lim_{n \to +\infty} \frac{1}{n} \sum_{i=1}^{n} x_i x_i^\top \in \mathbb{R}^{p \times p}$ (covariance matrix).
- Note that condition depends on $\theta$ and $J$
Sparse recovery: theoretical results for the Lasso

Model selection consistency (Lasso)

- Assume $\theta$ sparse and denote $J = \{j : \theta[j] \neq 0\}$ the nonzero pattern
- **Irrepresentable Condition** ($\gamma$) [Zhao and Yu, 2006, Wainwright, 2009]

\[
\|Q_{J^c} J Q_{J}^{-1} \text{sign}(\theta_J)\|_\infty \leq 1 - \gamma
\]

where $Q = \lim_{n \to +\infty} \frac{1}{n} \sum_{i=1}^{n} x_i x_i^\top \in \mathbb{R}^{p \times p}$ (covariance matrix).
- Note that condition depends on $\theta$ and $J$

Theorem (Model selection for classical asymptotics (i.e., $p$ fixed))

*IC(0) is necessary and IC($\gamma$) for $\gamma > 0$ is sufficient for model selection consistency.*

High-dimension ($p \to +\infty$): additional requirements

- Sample size condition: $n > k \log p$
- Requires lower-bound on magnitude of nonzero $\theta[j]$

see Bühlmann and Van De Geer [2011] for a review.
Compressed sensing [Candès et al., 2006] says that

- an $s$-sparse signal $\alpha^*$ in $\mathbb{R}^p$ can be exactly recovered by observing $x = D\alpha^*$ and solving the linear program

$$\min_{\alpha \in \mathbb{R}^p} ||\alpha||_1 \text{ s.t. } x = D\alpha,$$

where $D$ satisfies the RIP assumption with $\delta_{2s} \leq \sqrt{2} - 1$. Moreover, the convex relaxation is exact.

- matrices $D$ in $\mathbb{R}^{m \times p}$ satisfying the RIP assumption can be obtained with simple random sampling schemes with

$$m = O(s \log(p/s)).$$
Compressed sensing and sparse recovery

Remarks

- The theory also admits extensions to approximately sparse signals, noisy measurements.
- Extensions where $D$ is replaced by $Z^T D$ where $Z$ is random and $D$ deterministic;
- the dictionaries we are using in this lecture do not satisfy RIP;
- sparse estimation and sparse coding is not compressed sensing.
Some thoughts from Hocking [1976]:

**The problem of selecting a subset of independent or predictor variables is usually described in an idealized setting.** That is, it is assumed that (a) the analyst has data on a large number of potential variables which include all relevant variables and appropriate functions of them plus, possibly, some other extraneous variables and variable functions and (b) the analyst has available “good” data on which to base the eventual conclusions. **In practice, the lack of satisfaction of these assumptions may make a detailed subset selection analysis a meaningless exercise.**
Conclusions from the first part

- the sparsity principle has been used for a long time, and this is not a recent idea;
- there are numerous ways of designing sparse regularization functions adapted to a particular problem. Choosing the best one is not easy and requires some domain knowledge;
- the dictionaries we will use in this literature almost never satisfy theoretical assumptions ensuring sparse recovery.

Other take-home messages:
- sparsity is not always good. If possible, try $\ell_2$ before trying $\ell_1$;
- convexity is not always good. When trying $\ell_1$, try also $\ell_0$. 
Part II: Discovering the structure of natural images
A short introduction to parsimony

Discovering the structure of natural images
  - Dictionary learning
  - Pre-processing
  - Principal component analysis
  - Clustering or vector quantization
  - Structured dictionary learning
  - Other matrix factorization methods

Sparse models for image processing

Optimization for sparse estimation
Dictionary learning

The goal of automatically learning local structures in natural images was first achieved by neuroscientists.

The model of Olshausen and Field [1996] looks for a dictionary $D$ adapted to a training set of natural image patches $x_i, i = 1, \ldots, n$:

$$
\min_{D \in C, A \in \mathbb{R}^{p \times n}} \frac{1}{n} \sum_{i=1}^{n} \|x_i - D \alpha_i\|_2^2 + \lambda \psi(\alpha_i),
$$

where $A = [\alpha_1, \ldots, \alpha_n]$ and $C \triangleq \{D \in \mathbb{R}^{m \times p} : \forall j, \|d_j\|_2 \leq 1\}$.

Typical settings

- $n \approx 100,000$;
- $m = 10 \times 10$ pixels;
- $p = 256$.  

Dictionary learning

**Figure:** with centering
Dictionary learning

Figure: with whitening
Dictionary learning

Why was it found impressive by neuroscientists?

- since Hubel and Wiesel [1968], it is known that some visual neurons are responding to particular image features, such as oriented edges.
- Later, Daugman [1985] demonstrated that fitting a linear model to neuronal responses given a visual stimuli may produce filters that can be well approximated by a two-dimensional Gabor function.
- the original motivation of Olshausen and Field [1996] was to establish a relation between the statistical structure of natural images and the properties of neurons from area V1.

The results provided some “support” for classical models of V1 based on Gabor filters.
Dictionary learning

Why was it found impressive by neuroscientists?

- since Hubel and Wiesel [1968], it is known that some visual neurons are responding to particular image features, such as oriented edges.
- Later, Daugman [1985] demonstrated that fitting a linear model to neuronal responses given a visual stimuli may produce filters that can be well approximated by a two-dimensional Gabor function.
- the original motivation of Olshausen and Field [1996] was to establish a relation between the statistical structure of natural images and the properties of neurons from area V1.

Warning

In fact, little is known about the early visual cortex [Olshausen and Field, 2005, Carandini et al., 2005].
However, [...], there remains a great deal that is still unknown about how V1 works and its role in visual system function. We believe it is quite probable that the correct theory of V1 is still far afield from the currently proposed theories.
Dictionary learning

Point of views

Matrix factorization

It is useful to see dictionary learning as a matrix factorization problem

\[
\min_{D \in \mathbb{C}, A \in \mathbb{R}^{p \times n}} \frac{1}{2n} \|X - DA\|_F^2 + \lambda \psi(A).
\]

This is simply a matter of notation:

\[
\min_{D \in \mathbb{C}, A \in \mathbb{R}^{p \times n}} \frac{1}{n} \sum_{i=1}^{n} \|x_i - D\alpha_i\|_2^2 + \lambda \psi(\alpha_i),
\]

but the matrix factorization point of view allows us to make connections with numerous other unsupervised learning techniques, such as K-means, PCA, NMF, ICA...
Dictionary learning

Point of views

Empirical risk minimization

$$\min_{D \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^{n} L(x_i, D),$$

with

$$L(x, D) \triangleq \min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \|x - D\alpha\|_2^2 + \lambda \psi(\alpha).$$

Again, this is a matter of notation, but the empirical risk minimization point of view paves the way to

- stochastic optimization [Mairal et al., 2010a];
- some theoretical analysis [Vainsencher et al., 2011, Gribonval et al., 2013].
Dictionary learning
Constrained variants

The formulations below are not equivalent

$$\min_{D \in \mathcal{C}, A \in \mathbb{R}^{p \times n}} \sum_{i=1}^{n} \frac{1}{2} \|x_i - D\alpha_i\|_2^2 \quad \text{s.t.} \quad \psi(\alpha_i) \leq \mu.$$  

or

$$\min_{D \in \mathcal{C}, A \in \mathbb{R}^{p \times n}} \sum_{i=1}^{n} \psi(\alpha_i) \quad \text{s.t.} \quad \|x_i - D\alpha_i\|_2^2 \leq \varepsilon.$$  

Using one instead of another is a matter of taste and of the problem at hand.
Pre-processing of natural image patches

Centering (also called removing the DC component)

\[ x_i \leftarrow x_i - \left( \frac{1}{m} \sum_{j=1}^{m} x_i[j] \right) 1_m, \]

(a) Without pre-processing.  (b) After centering.
Pre-processing of natural image patches

Contrast (variance) normalization

\[ x_i \leftarrow \frac{1}{\max(\|x_i\|_2, \eta)} x_i. \]

\( \eta \) can be 0.2 times the mean value of the \( \|x_i\|_2 \).

(a) After centering.  
(b) After contrast normalization.
Pre-processing of natural image patches

Whitening after centering

\[ x_i \leftarrow US^\dagger U^\top x_i, \]

where \((1/\sqrt{n})X = USV^\top\) (SVD). Sometimes, small singular values are also set to zero. The resulting covariance \((1/n)XX^\top\) is close to identity.

(a) After centering.  
(b) After whitening.
Pre-processing of natural image patches
Treatment of color image patches

Should we use RGB?
Pre-processing of natural image patches

Treatment of color image patches

Should we use RGB?

- RGB dates back to our first understanding of the nature of light: color spectrum [Newton, 1675], trichromatic vision [Young, 1845], color composition [Grassmann, 1854, Maxwell, 1860, von Helmholtz, 1852], biological photoreceptors [Nathans et al., 1986];

- other color spaces, such as CIELab, YIQ, YCrBr have less correlated color channels [Pratt, 1971, Sharma and Trussell, 1997], and provide a better perceptual distance;

- it does not mean that RGB should never be used: changing the color space will also change the nature of the noise...
Pre-processing of natural image patches

Treatment of color image patches

Should we use RGB?

- RGB dates back to our first understanding of the nature of light: color spectrum [Newton, 1675], trichromatic vision [Young, 1845], color composition [Grassmann, 1854, Maxwell, 1860, von Helmholtz, 1852], biological photoreceptors [Nathans et al., 1986];
- other color spaces, such as CIE Lab, YIQ, YCrBr have less correlated color channels [Pratt, 1971, Sharma and Trussell, 1997], and provide a better perceptual distance;
- it does not mean that RGB should never be used: changing the color space will also change the nature of the noise...

how do we perform centering, whitening and normalization?
Pre-processing of natural image patches
Treatment of color image patches

Should we use RGB?

- RGB dates back to our first understanding of the nature of light: color spectrum [Newton, 1675], trichromatic vision [Young, 1845], color composition [Grassmann, 1854, Maxwell, 1860, von Helmholtz, 1852], biological photoreceptors [Nathans et al., 1986];
- other color spaces, such as CIELab, YIQ, YCrBr have less correlated color channels [Pratt, 1971, Sharma and Trussell, 1997], and provide a better perceptual distance;
- it does not mean that RGB should never be used: changing the color space will also change the nature of the noise...

how do we perform centering, whitening and normalization?

- center each R,G,B channel independently.
Pre-processing of natural image patches

(a) Without pre-processing.

(b) After centering.

(c) Centering and normalization.

(d) After whitening.
Principal component analysis (PCA)

Also known as the Karhunen-Loève or Hotelling transform [Hotelling, 1933], it is often presented as an iterative process finding orthogonal directions maximizing variance in the data.

In fact, it can be cast as a \textit{low-rank matrix factorization} problem:

\[
\min_{U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{n \times k}} \left\| X - UV^\top \right\|_F^2 \quad \text{s.t.} \quad U^\top U = I_k,
\]

where the rows of $X$ are centered.

As a consequence of the theorem of Eckart and Young [1936], the matrix $U$ contains the principal components of $X$ corresponding to the $k$ largest singular values.
Figure: On the right, we visualize the principal components of 400,000 randomly sampled natural image patches of size $16 \times 16$. On the left, we display a discrete cosine transform (DCT) dictionary [Ahmed et al., 1974].
Principal component analysis (PCA)

Figure: Visualization of the principal components of all overlapping patches from the image tiger. Even though the image is not natural, its principal components are similar to the previous ones.
Principal component analysis (PCA)
[Bossonaier and Snyder, 1986, Simoncelli and Olshausen, 2001, Hyvärinen et al., 2009].

Warning
The sinusoids produced by PCA have nothing to do with the structure of natural images, but are due to a property of shift invariance.
Principal component analysis (PCA)

[Bosmaier and Snyder, 1986, Simoncelli and Olshausen, 2001, Hyvärinen et al., 2009].

Warning

The sinusoids produced by PCA have nothing to do with the structure of natural images, but are due to a property of shift invariance.

Consider an infinite 1D signal with covariance \( \Sigma[k, l] = \sigma(k - l) \), where \( \sigma \) is even. Then, for all \( \omega \) and \( \varphi \),

\[
\sum_l \Sigma(k, l) e^{i(\omega l + \varphi)} = \sum_l \sigma(l - k) e^{i(\omega l + \varphi)} = \left( \sum_{l'} \sigma(l') e^{i\omega l'} \right) e^{i(\omega k + \varphi)},
\]

Since the function \( \sigma \) is even, the infinite sum \( \left( \sum_{l'} \sigma(l') e^{i\omega l'} \right) \) is real, and the signals \( \left[ \sin(\omega k + \varphi) \right]_{k \in \mathbb{Z}} \) are all eigenvectors of \( \Sigma \).
Principal component analysis (PCA)
[Bossonmaier and Snyder, 1986, Simoncelli and Olshausen, 2001, Hyvärinen et al., 2009].

Warning
The sinusoids produced by PCA have **nothing to do with the structure of natural images**, but are due to a property of **shift invariance**.

Consider an infinite 1D signal with covariance $\Sigma[k, l] = \sigma(k - l)$, where $\sigma$ is even. Then, for all $\omega$ and $\varphi$,

$$
\sum_l \Sigma(k, l) e^{i(\omega l + \varphi)} = \sum_l \sigma(l - k) e^{i(\omega l + \varphi)} = \left( \sum_{l'} \sigma(l') e^{i \omega l'} \right) e^{i(\omega k + \varphi)},
$$

Since the function $\sigma$ is even, the infinite sum $\left( \sum_{l'} \sigma(l') e^{i \omega l'} \right)$ is real, and the signals $[\sin(\omega k + \varphi)]_{k \in \mathbb{Z}}$ are all eigenvectors of $\Sigma$.

Note that controlling the approximation of the PCs by the discrete Fourier transform for finite signals is non-trivial [Pearl, 1973].
Clustering or vector quantization

First used on natural image patches for compression and communication purposes [Nasrabadi and King, 1988, Gersho and Gray, 1992]. The goal is to find \( p \) clusters in the data, by minimizing the following objective:

\[
\min_{\mathbf{D} \in \mathbb{R}^{m \times p}} \sum_{i=1}^{n} \| \mathbf{x}_i - \mathbf{d}_{l_i} \|_2^2, \\
\forall i, \ l_i \in \{1, \ldots, p\}
\]

(1)

This is again a matrix factorization problem

\[
\min_{\mathbf{D} \in \mathbb{R}^{m \times p}} \min_{\mathbf{A} \in \{0,1\}^{p \times n}} \frac{1}{2n} \| \mathbf{X} - \mathbf{DA} \|_F^2 \quad \text{s.t.} \quad \forall i, \ \sum_{j=1}^{p} \alpha_i[j] = 1.
\]
Clustering or vector quantization

(a) With centering.  
(b) With whitening.

Figure: Visualization of $p = 256$ centroids computed with the algorithm K-means on $n = 400,000$ image patches of size $m = 16 \times 16$ pixels.
Dictionary learning on color image patches

Figure: Dictionaries learned on RGB patches.
Dictionary learning with structured sparsity

Formulation

\[
\min_{D \in \mathbb{C}, A \in \mathbb{R}^{p \times n}} \frac{1}{2n} \|X - DA\|_F^2 + \frac{\lambda}{n} \sum_{i=1}^{n} \sum_{g \in \mathcal{G}} \|\alpha_i[g]\|_q.
\]

Group structures

- **hierarchical**: organize the dictionary elements in a tree [Jenatton et al., 2010, 2011b];

- **topographic**: organize the elements on a 2D grid [Kavukcuoglu et al., 2009, Mairal et al., 2011]. The groups are $3 \times 3$ or $4 \times 4$ spatial neighborhoods.

The second group structure is inspired by **topographic ICA** [Hyvärinen et al., 2001].
Dictionary learning with structured sparsity

Hierarchical dictionary learning

Figure: Hierarchical dictionaries learned on natural image patches of size 16 × 16 pixels.
Dictionary learning with structured sparsity

Topographic dictionary learning

(a) With $3 \times 3$ neighborhoods.

(b) With $4 \times 4$ neighborhood.

Figure: Topographic dictionaries learned on whitened natural image patches of size $12 \times 12$ pixels.
Other matrix factorization methods

Independent component analysis (ICA)

Assume that $x$ is a random variable—here a natural image patch— and the columns of $X = [x_1, \ldots, x_n]$ are random realizations of $x$.

ICA is a principle looking for a factorization $x = D\alpha$, where $D$ is orthogonal and $\alpha$ is a random vector whose entries are statistically independent [Hérault et al., 1985, Comon, 1994, Bell and Sejnowski, 1997, Hyvärinen et al., 2009].
Other matrix factorization methods
Independent component analysis (ICA)

Assume that $\mathbf{x}$ is a random variable—here a natural image patch—and the columns of $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ are random realizations of $\mathbf{x}$.

ICA is a principle looking for a factorization $\mathbf{x} = \mathbf{D}\alpha$, where $\mathbf{D}$ is orthogonal and $\alpha$ is a random vector whose entries are statistically independent [Hérault et al., 1985, Comon, 1994, Bell and Sejnowski, 1997, Hyvärinen et al., 2009].

Warning
Because ICA is only a principle, there is not a unique ICA formulation.
Other matrix factorization methods

Independent component analysis (ICA)

Assume that $\mathbf{x}$ is a random variable—here a natural image patch— and the columns of $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ are random realizations of $\mathbf{x}$.

ICA is a **principle** looking for a factorization $\mathbf{x} = \mathbf{D} \alpha$, where $\mathbf{D}$ is orthogonal and $\alpha$ is a random vector **whose entries are statistically independent** [Hérault et al., 1985, Comon, 1994, Bell and Sejnowski, 1997, Hyvärinen et al., 2009].

How do we measure independence?

Compare $p(\alpha)$ with the product of its marginals $\prod_{j=1}^{p} p(\alpha[j])$:

$$KL \left( p(\alpha), \prod_{j=1}^{p} p(\alpha[j]) \right) \triangleq \int_{\mathbb{R}^p} p(\alpha) \log \left( \frac{p(\alpha)}{\prod_{j=1}^{p} p(\alpha[j])} \right) d\alpha,$$

which is zero iff the $\alpha[j]'s$ are independent [Cover and Thomas, 2006].
Other matrix factorization methods

Independent component analysis (ICA)

We can rewrite the Kullback-Leibler distance with entropies

$$KL \left( p(\alpha), \prod_{j=1}^{p} p(\alpha[j]) \right) = \sum_{j=1}^{p} H(\alpha[j]) - H(\alpha).$$

The entropy $H(\alpha)$ can be shown to be independent of $D$ when $D$ is orthogonal and $x$ is whitened. Minimizing $KL$ amounts to minimizing

$$\sum_{j=1}^{p} H(\alpha[j]) = \sum_{j=1}^{p} H(d_j^\top x).$$

We are close to an ICA “formulation” but not yet there. The entropy is an abstract quantity that is not computable.
Other matrix factorization methods

Independent component analysis (ICA)

Strategies leading to concrete formulations/algorithms for solving the following problem after whitening the data

\[
\min_D \sum_{j=1}^{p} H(d_j^T x) \quad \text{s.t.} \quad D^T D = I.
\]

- **parameterizing the densities** \( p(d_j^T x) \), leading to maximum likelihood estimation [see Hyvärinen et al., 2004];
- plug in **non-parametric estimators** of the entropy [Pham, 2004];
- encourage the distributions of the \( \alpha[j] \)'s to be **“non-Gaussian”** [Cardoso, 2003].

Among all probability distributions with same variance, the Gaussian ones are known to maximize entropy [Cover and Thomas, 2006].
Other matrix factorization methods

Non-negative matrix factorization [Paatero and Tapper, 1994].

\[
\min_{D \in \mathbb{R}^{m \times p}, A \in \mathbb{R}^{p \times n}} \| X - DA \|_F^2 \quad \text{s.t.} \quad D \geq 0 \quad \text{and} \quad A \geq 0.
\]
Other matrix factorization methods

Non-negative matrix factorization [Paatero and Tapper, 1994].

\[
\min_{D \in \mathbb{R}^{m \times p}, A \in \mathbb{R}^{p \times n}} \|X - DA\|_F^2 \quad \text{s.t.} \quad D \geq 0 \text{ and } A \geq 0.
\]

Archetypal analysis [Cutler and Breiman, 1994].

- for all dictionary element \( j \), \( d_j = X\beta_j \), where \( \beta_j \) is in the simplex \( \Delta_n \triangleq \{ \beta \in \mathbb{R}^n \text{ s.t. } \beta \geq 0 \text{ and } \sum_{i=1}^n \beta[i] = 1 \} \).
- for all data point \( i \), \( x_i \) is close to \( D\alpha_i \), where \( \alpha_i \) is in \( \Delta_p \);
- formulation:

\[
\min_{\alpha_i \in \Delta_p \text{ for } 1 \leq i \leq n, \beta_j \in \Delta_n \text{ for } 1 \leq j \leq p} \|X - XBA\|_F^2,
\]

where \( A = [\alpha_1, \ldots, \alpha_n] \), \( B = [\beta_1, \ldots, \beta_p] \) and the matrix of archetypes \( D \) is equal to the product \( XB \).
Illustration of archetypal analysis
[Chen, Mairal, and Harchaoui, 2014]

Classical landmarks
Illustration of archetypal analysis
[Chen, Mairal, and Harchaoui, 2014]

Unexpected archetypes
Illustration of archetypal analysis
[Chen, Mairal, and Harchaoui, 2014]

“Compositional” archetypes
Illustration of archetypal analysis
[Chen, Mairal, and Harchaoui, 2014]

Decomposition of an image onto the archetypal set

\[
\begin{align*}
30.5\% \text{ (A)} & + 25.5\% \text{ (B)} + 24.6\% \text{ (C)} \\
= 29.0\% \text{ (A)} & + 21.5\% \text{ (B)} + 20.3\% \text{ (C)} + 16.7\% \text{ (D)}
\end{align*}
\]
Combination of archetypal analysis and style transfer
[Wynen, Schmid, and Mairal, 2018]
Combination of archetypal analysis and style transfer
[Wynen, Schmid, and Mairal, 2018]

Arch. 40 = 28.42 + 27.83 + 24.19

Arch. 41 = 91.15 + 3.82 + 2.72

Arch. 42 = 21.23 + 16.54 + 13.27
Combination of archetypal analysis and style transfer

[Wynen, Schmid, and Mairal, 2018]

Arch. 16 = 51.01 + 34.89 + 14.09

Arch. 17 = 47.34 + 26.79 + 15.89

Arch. 18 = 91.88 + 8.12
Combination of archetypal analysis and style transfer
[Wynen, Schmid, and Mairal, 2018]
Combination of archetypal analysis and style transfer
[Wynen, Schmid, and Mairal, 2018]

Image 800

59.17

+ Arch. 14 = 64.85 + 20.57 + 4.92

17.81

+ Arch. 0 = 47.76 + 28.54 + 23.71

14.26

Arch. 10 = 62.07 + 18.13 + 11.55
Combination of archetypal analysis and style transfer

[Wynen, Schmid, and Mairal, 2018]
Other matrix factorization methods
Convolutional sparse coding [Zhu et al., 2005, Zeiler et al., 2010]

Main idea

Decompose directly the full image $x$ using small dictionary elements placed at all possible positions in the image.

Given a dictionary $D^{m \times p}$ where $m$ is a patch size, and an image $x$ in $\mathbb{R}^l$, the image decomposition can be written.

$$\min_{A \in \mathbb{R}^{p \times l}} \frac{1}{2} \left\| x - \sum_{k=1}^{l} R_k^T D \alpha_k \right\|_2^2 + \lambda \sum_{k=1}^{l} \| \alpha_k \|_1,$$

Model with effective applications to visual recognition [Zeiler et al., 2010, Rigamonti et al., 2013, Kavukcuoglu et al., 2010].

Then, the extension to dictionary learning is easy.
Other matrix factorization methods

Convolutional sparse coding [Zhu et al., 2005, Zeiler et al., 2010]

Figure: Visualization of $p = 100$ dictionary elements learned on 30 whitened natural images.
Conclusions from the second part

- intriguing structures naturally emerge from natural images;
- matrix factorization is an effective tool to find these structures;

Advertisement

- some matlab code will be provided upon publication of the monograph for generating most of the figures from this lecture.
Conclusions from the second part

- intriguing structures naturally emerge from natural images;
- matrix factorization is an effective tool to find these structures;

Advertisement

- some matlab code will be provided upon publication of the monograph for generating most of the figures from this lecture.

Question

Is unsupervised learning on natural image patches useful for any prediction task?
Part III: Sparse models for image processing
1. A short introduction to parsimony

2. Discovering the structure of natural images

3. Sparse models for image processing
   - Image denoising
   - Image inpainting
   - Image demosaicking
   - Video processing
   - Image up-scaling
   - Inverting nonlinear local transformations
   - Other patch modeling approaches

4. Optimization for sparse estimation
Image denoising

\[ y \text{ measurements} = x_{\text{orig}} \text{ original image} + w \text{ noise} \]
Image denoising

Classical image models

\[
\underbrace{y}_{\text{measurements}} = \underbrace{x_{\text{orig}}}_{\text{original image}} + \underbrace{w}_{\text{noise}}.
\]

Energy minimization problem - MAP estimation

\[
E(x) = \frac{1}{2} \|y - x\|_2^2 + \psi(x).
\]

Some classical priors

- Smoothness $\lambda \|Lx\|_2^2$;
- total variation $\lambda \|\nabla x\|_1^2$ [Rudin et al., 1992];
- Markov random fields [Zhu and Mumford, 1997];
- wavelet sparsity $\lambda \|Wx\|_1$. 
Image denoising

The method of Elad and Aharon [2006]

Given a fixed dictionary $\mathbf{D}$, a patch $\mathbf{y}_i$ is denoised as follows:

1. center $\mathbf{y}_i$,

$$\mathbf{y}_i^c \triangleq \mathbf{y}_i - \mu_i \mathbf{1}_m \quad \text{with} \quad \mu_i \triangleq \frac{1}{n} \mathbf{1}^T_m \mathbf{y}_i;$$

2. find a sparse linear combination of dictionary elements that approximates $\mathbf{y}_i^c$ up to the noise level:

$$\min_{\alpha_i \in \mathbb{R}^p} \|\alpha_i\|_0 \quad \text{s.t.} \quad \|\mathbf{y}_i^c - \mathbf{D} \alpha_i\|_2^2 \leq \varepsilon,$$

where $\varepsilon$ is proportional to the noise variance $\sigma^2$;

3. add back the mean component to obtain the clean estimate $\hat{\mathbf{x}}_i$:

$$\hat{\mathbf{x}}_i \triangleq \mathbf{D} \alpha_i^* + \mu_i \mathbf{1}_m,$$
Image denoising

The method of Elad and Aharon [2006]

An **adaptive** approach

1. extract all overlapping $\sqrt{m} \times \sqrt{m}$ patches $y_i$.
2. **dictionary learning**: learn $D$ on the set of centered noisy patches $[y_1^c, \ldots, y_n^c]$.
3. **final reconstruction**: find an estimate $\hat{x}_i$ for every patch using the approach of the previous slide;
4. **patch averaging**:

   $$\hat{x} = \frac{1}{m} \sum_{i=1}^{n} R_i^T \hat{x}_i,$$

**Remark**

Like other state-of-the-art denoising approaches, it is patch-based [Buades et al., 2005, Dabov et al., 2007].
Practical tricks

- use larger patches when the noise level is high;
- choose $\varepsilon = m(1.15\sigma)^2$ or take the 0.9-quantile of the $\chi^2_m$-distribution.
- always use the $l_0$ regularization for the final reconstruction;
- using $l_1$ for learning the dictionary seems to yield better results.
For removing small holes in the image, a natural extension consists in introducing a **binary mask** $M_i$ in the formulation:

$$\min_{D \in \mathcal{C}, A \in \mathbb{R}^{p \times n}} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \|M_i(y_i - D \alpha_i)\|_2^2 + \lambda \psi(\alpha_i),$$

The approach assumes that

- the noise is not structured;
- the holes are smaller than the patch size.

The problem is called inpainting [Bertalmio et al., 2000].
Image inpainting

[Mairal et al., 2008a,b]
Since 1699, when French explorers landed at the great bend of the Mississippi River and celebrated the first Mardi Gras in North America, New Orleans has brewed a fascinating melange of cultures. It was French, then Spanish, then French again, then sold to the United States. Through all these years, and even into the 1900s, others arrived from everywhere: Acadians (Cajuns), Africans, indige-
Image inpainting

[Mairal et al., 2008a,b]
Image inpainting

Inpainting a 12-Mpixel photograph [Mairal et al., 2009a]

The Salinas Valley is in Northern California. It is a long narrow valley between two ranges of mountains, and the Salinas River winds and twist up the center until it falls at last into Monterey Bay.

I remember my childhood names for grasses and sweet flowers. I remember where a toad may live and what time the birds awaken in the summer and what trees and seasons smelt the how people looked and walked and smelled each. The memory of odors is very rich.

I remember that the Salinas Mountains to the east of the valley were light gray mountains full of sun and levelness and a kind of invitation, so that you wanted to climb into their warm foothills almost as you want to climb into the rap of a beloved mother. They were beckoning mountains with a brown grass love. The Santa Lucias stood so against the sky to the west and kept the valley from the open sea, and they were dark and brooding unfriendly and dangerous. I always found in myself a quest of west and a love of east. Where I ever got such an idea I cannot say, unless it could be that the morning came over the peaks of the Salinas and the night drifted back from the ridges of the Santa Lucias. It may be that the birth and death of the day had some part in my feeling about the two ranges of mountains.

From both sides of the valley little streams slipped out of the hill canyons and fell into the bed of the Salinas River. In the winter of wet years the streams ran full-freshet, and they swelled the river until sometimes it raged and boiled, bank full, and then it was a destroyer. The river tore the edges of the farm lands and washed whole acres down; it toppled barns and houses into itself, to go floating and bobbing away. It trapped cows and pigs and sheep and drowned them in its muddy brown water and carried them to the sea. Then when the late spring came, the river drew up from its troughs and the sand banks appeared. And in the summer the river didn’t run at all above ground. Some pools would be left in the deep soil places under a high bank. The fults and grasses grew thick, and willows straightened up with the flood debris in their upper branches. The Salinas was only a real-time river. The summer sun drove it underground. It was not a full river at all, but it was the only one we had and so we boasted about it, how dangerous it was, how dangerous it was in a wet winter and how dry it was in a dry summer. You can boast about anything if it’s all you have. Maybe the less you have, the more you are required to boast.

The floor of the Salinas Valley, between the ranges and below the foothills, is level because this valley used to be the bottom of a hundred-mile inlet from the sea. The river mouth at Moss Landing was centuries ago the entrance to this long inland water. Once, fifty miles down the valley, my father built a well. The well came up first with topsoil and then with gravel and then with white sea sand full of shells and even pi...
Image inpainting

Inpainting a 12-Mpixel photograph [Mairal et al., 2009a]
Image inpainting

Inpainting a 12-Mpixel photograph [Mairal et al., 2009a]
Image inpainting

Inpainting a 12-Mpixel photograph [Mairal et al., 2009a]
Problem

The noise pattern is very structured: the previous inpainting scheme needs to be modified [Mairal et al., 2008a].
Image demoisicking

(a) Mosaicked image  (b) Demosaicked image A  (c) Demosaicked image B

**Figure:** Demosaicked image A is with the approach previously described; image B is with an extension called non-local sparse model [Mairal et al., 2009b].
Image demoisicking

(a) Zoom  (b) Zoom  (c) Zoom

Figure: Demosaicked image A is with the approach previously described; image B is with an extension called non-local sparse model [Mairal et al., 2009b].
Video processing

Extension developed by Protter and Elad [2009]:

Key ideas for video processing

- Using a 3D dictionary.
- Processing of many frames at the same time.
- Dictionary propagation.
Video processing
Inpainting, [Mairal et al., 2008b]

Figure: Inpainting results.
Video processing
Inpainting, [Mairal et al., 2008b]

Figure: Inpainting results.
Video processing

Inpainting, [Mairal et al., 2008b]

Figure: Inpainting results.
Video processing

Inpainting, [Mairal et al., 2008b]

Figure: Inpainting results.
Video processing
Inpainting, [Mairal et al., 2008b]

Figure: Inpainting results.
Video processing
Color video denoising, [Mairal et al., 2008b]

Figure: Inpainting results.
Video processing

Color video denoising, [Mairal et al., 2008b]

Figure: Inpainting results.
Video processing

Color video denoising, [Mairal et al., 2008b]

**Figure:** Inpainting results.
Video processing
Color video denoising, [Mairal et al., 2008b]

Figure: Inpainting results.
Video processing
Color video denoising, [Mairal et al., 2008b]

Figure: Inpainting results.
Image up-scaling

The main recipe of Yang et al. [2010]

The approach requires a database of pairs of training patches \((x^l_i, x^h_i)_{i=1}^n\), where \(x^l_i\) in \(\mathbb{R}^{m_l}\) is a low-resolution version of the patch \(x^h_i\) in \(\mathbb{R}^{m_h}\).

Training step:

\[
\min_{D^l \in \mathcal{C}^l, D^h \in \mathcal{C}^h} \frac{1}{n} \sum_{i=1}^n \frac{1}{2m_l} \left\| x^l_i - D^l \alpha_i \right\|_2^2 + \frac{1}{2m_h} \left\| x^h_i - D^h \alpha_i \right\|_2^2 + \lambda \left\| \alpha_i \right\|_1 ,
\]

\(D^l\) and \(D^h\) are jointly learned such that the pairs \((x^l_i, x^h_i)\) “share” the same sparse decompositions on the dictionaries.

Reconstruction step given a low-resolution image:

\[
\min_{\beta_i \in \mathbb{R}^p} \frac{1}{2m_l} \left\| y^l_i - D^l \beta_i \right\|_2^2 + \frac{1}{2m_h} \left\| z_i - D^h \beta_i \right\|_2^2 + \lambda \left\| \beta_i \right\|_1 ,
\]
Image up-scaling

Variant with regression [Zeyde et al., 2012, Couzinie-Devy et al., 2011, Yang et al., 2012]

compute $D_l$ and $A$ with a classical dictionary learning formulation,

$$\min_{D_l \in \mathcal{C}_l, A \in \mathbb{R}^{p \times n}} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \|x^l_i - D_l \alpha_i\|^2_2 + \lambda \|\alpha_i\|_1.$$ 

obtain $D_h$ by solving a multivariate regression problem:

$$\min_{D_h \in \mathbb{R}^{m_h \times p}} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \|x^h_i - D_h \alpha_i\|_2^2,$$

where the $\alpha_i$’s are fixed after the first step. See also Zeyde et al. [2012] for other variants.

Main difference with Yang et al. [2010]

- testing and training is more consistent;
- $D_h$ and $D_l$ are not learned jointly anymore.
Image up-scaling

Variant with task-driven dictionary learning [Couzinie-Devy et al., 2011, Yang et al., 2012]

Define

\[
\alpha^* (x, D) \triangleq \arg \min_{\alpha \in \mathbb{R}^p} \left[ \frac{1}{2} \| x - D \alpha \|_2^2 + \lambda \| \alpha \|_1 \right],
\]

Then, the joint dictionary learning formulation consists of minimizing

\[
\min_{D_l \in \mathcal{C}_l} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \left\| x_i^h - D_h \alpha^* (x_i^l, D_l) \right\|_2^2.
\] (3)

Pros and Cons

😊 testing and training is still consistent;

😊 \(D_h\) and \(D_l\) are learned jointly;

😢 optimization looks horribly difficult.
Image up-scaling
Scheme with regression

Pipeline:

\[
\text{Dictionary } D_h \quad \rightarrow \quad \text{Output } x^h \\
\text{Dictionary } D_I \quad \rightarrow \quad \text{Sparse codes } \alpha \\
\quad \quad \rightarrow \quad \text{Input vector } x' 
\]
Image up-scaling
Scheme with regression

First step: dictionary learning
Image up-scaling
Scheme with regression

Second step: regression
Image up-scaling

Scheme with task-driven dictionary learning

A single step: supervised (task-driven) dictionary learning

Weights $D_h$

Output $x^h$

Supervised Dictionary Learning?

Dictionary $D_l$

Sparse codes $\alpha$

Input vector $x^l$

In the neural network language, we need back-propagation [LeCun et al., 1998].
Image up-scaling
Scheme with task-driven dictionary learning [Couzinie-Devy et al., 2011]

Proposition
In the asymptotic regime, the cost function is differentiable and its gradient admits a simple form [Mairal et al., 2012].

Main recipe of the optimization
- initialize with the regression variant;
- use stochastic gradient descent.
- use classical heuristics from the neural network literature [LeCun et al., 1998].
Image up-scaling

Image from Couzinie-Devy et al. [2011]
Image up-scaling

Image from Couzinie-Devy et al. [2011]

**Figure**: Bicubic interpolation
Image up-scaling

Image from Couzinie-Devy et al. [2011]

Figure: from Couzinie-Devy et al. [2011].
Image up-scaling

Image from Couzinie-Devy et al. [2011]

Figure: Original
Image up-scaling

Image from Couzinie-Devy et al. [2011]

Figure: Bicubic interpolation
Image up-scaling

Image from Couzinie-Devy et al. [2011]

Figure: from Couzinie-Devy et al. [2011].
Image up-scaling

Image from Couzinie-Devy et al. [2011]

**Figure:** Original
Image up-scaling
Image from Couzinie-Devy et al. [2011]

Figure: Bicubic interpolation
Image up-scaling
Image from Couzinie-Devy et al. [2011]

Figure: from Couzinie-Devy et al. [2011].
Image up-scaling
Image from Couzinie-Devy et al. [2011]

Figure: Original
Image up-scaling
Image from Couzinie-Devy et al. [2011]

Figure: Bicubic interpolation
Image up-scaling

Image from Couzinie-Devy et al. [2011]

Figure: from Couzinie-Devy et al. [2011].
Inverting nonlinear local transformations

Remark

The previous up-scaling approaches are generic and can work for other types of local image transformations.

Example: inverse half-toning consists of reconstructing grayscale images from (probably old) binary ones, see, e.g., [Dabov et al., 2006]. A classical algorithm for producing binary images is the one of Floyd and Steinberg [1976].
Inverting nonlinear local transformations

Remark
The previous up-scaling approaches are generic and can work for other types of local image transformations.

Example: inverse half-toning consists of reconstructing grayscale images from (probably old) binary ones, see, e.g., [Dabov et al., 2006]. A classical algorithm for producing binary images is the one of Floyd and Steinberg [1976].

Warning
Inverse half-toning is probably not a hot topic in image processing nowadays.
Inverting nonlinear local transformations

Inverse half-toning [Mairal et al., 2012]
Inverting nonlinear local transformations

Inverse half-toning [Mairal et al., 2012]

**Figure**: Binary image
Inverting nonlinear local transformations

Inverse half-toning [Mairal et al., 2012]

Figure: Reconstructed.
Inverting nonlinear local transformations

Inverse half-toning [Mairal et al., 2012]

Figure: Original
Inverting nonlinear local transformations

Inverse half-toning [Mairal et al., 2012]

Figure: Binary image
Inverting nonlinear local transformations

Inverse half-toning [Mairal et al., 2012]

**Figure:** Reconstructed.
Inverting nonlinear local transformations

Inverse half-toning [Mairal et al., 2012]
Inverting nonlinear local transformations

Inverse half-toning [Mairal et al., 2012]
Inverting nonlinear local transformations

Inverse half-toning [Mairal et al., 2012]
Inverting nonlinear local transformations

Inverse half-toning [Mairal et al., 2012]
Inverting nonlinear local transformations

Inverse half-toning [Mairal et al., 2012]
Inverting nonlinear local transformations

Inverse half-toning [Mairal et al., 2012]
Inverting nonlinear local transformations
Inverse half-toning [Mairal et al., 2012]
Inverting nonlinear local transformations
Inverse half-toning [Mairal et al., 2012]
Inverting nonlinear local transformations

Inverse half-toning [Mairal et al., 2012]
Inverting nonlinear local transformations

Inverse half-toning [Mairal et al., 2012]
Other patch modeling approaches

Non-local means and non-parametric approaches

Image pixels are well explained by a Nadaraya-Watson estimator:

$$\hat{x}[i] = \sum_{j=1}^{n} \frac{K_h(y_i - y_j)}{\sum_{i=1}^{n} K_h(y_i - y_l)} y[j],$$

(4)

with successful application to

- texture synthesis: [Efros and Leung, 1999]
- image denoising (Non-local means): [Buades et al., 2005]
- image demosaicking: [Buades et al., 2009].
Other patch modeling approaches

BM3D

state-of-the-art image denoising approach [Dabov et al., 2007]:

- **block matching**: for each patch, find similar ones in the image;
- **3D wavelet filtering**: denoise blocks of patches with 3D-DCT;
- **patch averaging**: average estimates of overlapping patches;
- **second step with Wiener filtering**: use the first estimate to perform again and improve the previous steps.

Further refined by Dabov et al. [2009] with shape-adaptive patches and PCA filtering.
Other patch modeling approaches

Non-local sparse models [Mairal et al., 2009b]
Exploit some ideas of BM3D to combine the non-local means principle with dictionary learning.

The main idea is that similar patches should admit similar decompositions by using group sparsity:

The approach uses a block matching/clustering step, followed by group sparse coding and patch averaging.
Other patch modeling approaches

Non-local sparse image models
Other patch modeling approaches
Non-local sparse image models
Conclusions from the third part

- many inverse problems in image processing can be tackled by modeling natural image patches;
- dictionary learning is one effective way to do it, among others.
Part IV: Optimization for sparse estimation
1 A short introduction to parsimony

2 Discovering the structure of natural images

3 Sparse models for image processing

4 Optimization for sparse estimation
   - Sparse reconstruction with the $\ell_0$-penalty
   - Sparse reconstruction with the $\ell_1$-norm
   - Iterative reweighted $\ell_1$-algorithms
   - Optimization for dictionary learning
Sparse reconstruction with the $\ell_0$-penalty
Matching pursuit [Mallat and Zhang, 1993]

$\alpha = (0, 0, 0)$
Sparse reconstruction with the $\ell_0$-penalty

Matching pursuit [Mallat and Zhang, 1993]

$$\alpha = (0, 0, 0)$$
Sparse reconstruction with the $\ell_0$-penalty

Matching pursuit [Mallat and Zhang, 1993]

$\alpha = (0, 0, 0)$

$\begin{align*}
    \mathbf{r} &= \mathbf{d}_3 \\
    \mathbf{r} &= \langle \mathbf{r}, \mathbf{d}_3 \rangle \mathbf{d}_3
\end{align*}$
Sparse reconstruction with the $\ell_0$-penalty

Matching pursuit [Mallat and Zhang, 1993]

$\alpha = (0, 0, 0.75)$
Sparse reconstruction with the $\ell_0$-penalty

Matching pursuit [Mallat and Zhang, 1993]

$\alpha = (0, 0, 0.75)$
Sparse reconstruction with the $\ell_0$-penalty

Matching pursuit [Mallat and Zhang, 1993]

$\alpha = (0, 0, 0.75)$
Sparse reconstruction with the $\ell_0$-penalty

Matching pursuit [Mallat and Zhang, 1993]

$\alpha = (0, 0.24, 0.75)$
Sparse reconstruction with the $\ell_0$-penalty
Matching pursuit [Mallat and Zhang, 1993]

$\alpha = (0, 0.24, 0.75)$
Sparse reconstruction with the $\ell_0$-penalty

Matching pursuit [Mallat and Zhang, 1993]

$\alpha = (0, 0.24, 0.75)$
Sparse reconstruction with the $\ell_0$-penalty

Matching pursuit [Mallat and Zhang, 1993]

$\alpha = (0, 0.24, 0.65)$
Sparse reconstruction with the $\ell_0$-penalty

Matching pursuit [Mallat and Zhang, 1993]

$$\min_{\alpha \in \mathbb{R}^p} \left\| x - D\alpha \right\|_2^2 \quad \text{s.t.} \quad \left\| \alpha \right\|_0 \leq k.$$ 

1: $\alpha \leftarrow 0$
2: $r \leftarrow x$ (residual).
3: while $\left\| \alpha \right\|_0 < k$ do
4: Select the predictor with maximum inner-product with the residual
   $$\hat{j} \leftarrow \arg \max_{j=1,...,p} \left| d_j^\top r \right|$$
5: Update the residual and the coefficients
   $$\alpha[\hat{j}] \leftarrow \alpha[\hat{j}] + d_{\hat{j}}^\top r$$
   $$r \leftarrow r - (d_{\hat{j}}^\top r)d_{\hat{j}}$$
6: end while
Sparse reconstruction with the $\ell_0$-penalty
Matching pursuit [Mallat and Zhang, 1993]

Remarks

- Matching pursuit is a **coordinate descent** algorithm. It greedily selects one coordinate at a time and optimizes the cost function with respect to that coordinate.

\[
\alpha[\hat{j}] \leftarrow \arg \min_{\alpha \in \mathbb{R}} \left\| x - \sum_{l \neq \hat{j}} \alpha[l]d_l - \alpha d_{\hat{j}} \right\|_2^2.
\]

- Each coordinate can be selected several times during the process.

- The roots of this algorithm can be found in the statistics literature [Efroymson, 1960].
Sparse reconstruction with the $\ell_0$-penalty
Orthogonal matching pursuit [Pati et al., 1993]

$\alpha = (0, 0, 0)$
$\Gamma = \emptyset$
Sparse reconstruction with the $\ell_0$-penalty

Orthogonal matching pursuit [Pati et al., 1993]

$$\alpha = (0, 0, 0.75)$$
$$\Gamma = \{3\}$$
Sparse reconstruction with the $\ell_0$-penalty

Orthogonal matching pursuit [Pati et al., 1993]

$$\alpha = (0, 0.29, 0.63)$$

$$\Gamma = \{3, 2\}$$
Sparse reconstruction with the $\ell_0$-penalty
Orthogonal matching pursuit [Pati et al., 1993]

$$\min_{\alpha \in \mathbb{R}^p} \| \mathbf{x} - D\alpha \|_2^2 \quad \text{s.t.} \quad \| \alpha \|_0 \leq k$$

1: $\Gamma = \emptyset$.
2: for $\text{iter} = 1, \ldots, k$ do
3: Select the variable that most reduces the objective
   $$(\hat{j}, \hat{\beta}) \leftarrow \arg \min_{j \in \Gamma^\complement, \beta} \| \mathbf{x} - D_{\Gamma \cup \{j\}} \beta \|_2^2.$$  
4: Update the active set: $\Gamma \leftarrow \Gamma \cup \{\hat{j}\}$.
5: Update the coefficients:
   $$\alpha[\Gamma] \leftarrow \hat{\beta} \quad \text{and} \quad \alpha[\Gamma^\complement] \leftarrow 0.$$  
6: end for
Sparse reconstruction with the ℓ₀-penalty
Orthogonal matching pursuit [Pati et al., 1993]

Remarks

- this is an active-set algorithm.
- when a new variable is selected, the coefficients for the full set Γ are re-optimized:

\[ \alpha[\Gamma] = (D_\Gamma^T D_\Gamma)^{-1} D_\Gamma^T x, \]

and the residual is always orthogonal to the matrix \( D_\Gamma \) of previously selected dictionary elements:

\[ D_\Gamma^T (x - D\alpha) = D_\Gamma^T (x - D_\Gamma \alpha[\Gamma]) = 0. \]

- several variants of OMP exist regarding the selection rule of \( \hat{j} \). The one we use appears in Cotter et al. [1999].
Sparse reconstruction with the $\ell_0$-penalty
Orthogonal matching pursuit [Pati et al., 1993]

Keys for a fast implementation

- If available, use the Gram matrix $G = D^\top D$;
- Maintain the computation of $D^\top (x - D\alpha)$,
- Update the Cholesky decomposition of $(D_{\Gamma}^\top D_{\Gamma})^{-1}$.

The total complexity for decomposing $n k$-sparse signals of size $m$ with a dictionary of size $p$ is

$$O(p^2 m) + O(nk^3) + O(n(pm + pk^2)) = O(np(m + k^2))$$

It is also possible to use the matrix inversion lemma instead of a Cholesky decomposition.
Sparse reconstruction with the $\ell_0$-penalty
Orthogonal matching pursuit [Pati et al., 1993]

Example with the software SPAMS

```
>> I=double(imread('data/lena.eps'))/255;
>> %extract all patches of I
>> X=im2col(I,[8 8],’sliding’);
>> %load a dictionary of size 64 x 256
>> D=load(’dict.mat’);
>>
>> %set the sparsity parameter L to 10
>> param.L=10;
>> alpha=mexOMP(X,D,param);
```

On this dual-core laptop: **110000 signals processed per second!**
Sparse reconstruction with the $\ell_0$-penalty
Iterative hard-thresholding [Herrity et al., 2006, Blumensath and Davies, 2009]

**Require:** Signal $\mathbf{x}$ in $\mathbb{R}^m$, dictionary $\mathbf{D}$ in $\mathbb{R}^{m \times p}$, target sparsity $k$, gradient descent step size $\eta$, number of iterations $T$.

1: Initialize $\mathbf{\alpha} \leftarrow \mathbf{\alpha}_0$;
2: for $t = 1, \ldots, T$ do
3: perform one step of gradient descent:
   $$\mathbf{\alpha} \leftarrow \mathbf{\alpha} + \eta \mathbf{D}^\top (\mathbf{x} - \mathbf{D}\mathbf{\alpha});$$
4: choose $\tau$ to be the $k$-th largest entry of $\{|\mathbf{\alpha}[1]|, \ldots, |\mathbf{\alpha}[p]|\}$;
5: for $j = 1, \ldots, p$ do
6: hard-thresholding:
   $$\mathbf{\alpha}[j] \leftarrow \begin{cases} \mathbf{\alpha}[j] & \text{if } |\mathbf{\alpha}[j]| \geq \tau \\ 0 & \text{otherwise.} \end{cases}$$
7: end for
8: end for
9: return the sparse decomposition $\mathbf{\alpha}$ in $\mathbb{R}^p$. 
Sparse reconstruction with the \( \ell_0 \)-penalty

Iterative hard-thresholding [Herrity et al., 2006, Blumensath and Davies, 2009]

Remarks

- This is a **projected gradient algorithm**;

\[
\alpha \leftarrow \Pi_{\|\cdot\|_0 \leq k} [\alpha - \eta \nabla f(\alpha)].
\]

It performs one gradient descent step, followed by a Euclidean projection onto the non-convex set of \( k \)-sparse vectors.

- It can be easily extended to the (approximate) minimization of

\[
\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \| x - D\alpha \|^2_2 + \lambda \| \alpha \|_0.
\]

In that case, it is as a **proximal gradient algorithm**.

- It can be seen to iteratively decreases the value of the objective function from the **majorization-minimization** point of view.
Sparse reconstruction with the $\ell_0$-penalty

Majorization-minimization principle [Lange et al., 2000]

The principle for (approximately) minimizing a general cost function $f$:

$$\min_{\alpha \in \mathcal{A}} f(\alpha).$$

Figure: At each step, we update $\alpha \in \arg \min_{\alpha \in \mathcal{A}} g_\beta(\alpha)$.
Sparse reconstruction with the $\ell_0$-penalty

Majorization-minimization principle [Lange et al., 2000]

The principle for (approximately) minimizing a general cost function $f$:

$$\min_{\alpha \in A} f(\alpha).$$

Figure: At each step, we update $\alpha \in \arg \min_{\alpha \in A} g_\beta(\alpha)$

What is the surrogate for the iterative hard-thresholding algorithm?
Sparse reconstruction with the $\ell_0$-penalty

Majorization-minimization principle [Lange et al., 2000]

The principle for (approximately) minimizing a general cost function $f$:

$$\min_{\alpha \in A} f(\alpha).$$

Figure: At each step, we update $\alpha \in \arg\min_{\alpha \in A} g_\beta(\alpha)$

We need to introduce a few principles first...
Interlude
Introduction of a few optimization principles

An important inequality for smooth functions

If $\nabla f$ is $L$-Lipschitz continuous ($f$ does not need to be convex)

$$f(\alpha) \leq g(\alpha) = f(\alpha^0) + \nabla f(\alpha^0)^\top (\alpha - \alpha^0) + \frac{L}{2} \|\alpha - \alpha^0\|_2^2;$$

linear approximation
Introduction of a few optimization principles
An important inequality for smooth functions

If $\nabla f$ is $L$-Lipschitz continuous ($f$ does not need to be convex)

$$f(\alpha) \leq g(\alpha) = f(\alpha^0) + \nabla f(\alpha^0)^\top (\alpha - \alpha^0) + \frac{L}{2} \|\alpha - \alpha^0\|_2^2;$$

linear approximation

$$g(\alpha) = C_{\alpha^0} + \frac{1}{2} \|\alpha^0 - (1/L)\nabla f(\alpha^0) - \alpha\|_2^2.$$
Introduction of a few optimization principles

An important inequality for smooth functions

If $\nabla f$ is $L$-Lipschitz continuous ($f$ does not need to be convex)

\[
\begin{align*}
f(\alpha) &\leq g(\alpha) = f(\alpha^0) + \nabla f(\alpha^0)^\top (\alpha - \alpha^0) + \frac{L}{2} \|\alpha - \alpha^0\|_2^2; \\
\alpha^1 &= \alpha^0 - \frac{1}{L} \nabla f(\alpha^0). \text{(gradient descent step)}.
\end{align*}
\]
Sparse reconstruction with the $\ell_0$-penalty
iterative hard-thresholding [Herrity et al., 2006, Blumensath and Davies, 2009]

What is the surrogate $g_\beta(\alpha)$?
Sparse reconstruction with the $\ell_0$-penalty

iterative hard-thresholding [Herrity et al., 2006, Blumensath and Davies, 2009]

Simply the same as for the gradient descent algorithm:

$$g_\beta(\alpha) \triangleq f(\alpha) + \nabla f(\beta)^\top (\alpha - \beta) + \frac{L}{2}\|\beta - \alpha\|_2^2,$$

with $\beta = \alpha^\text{old}$, $L = (1/\eta)$ and $f(\alpha) = (1/2)\|x - D\alpha\|_2^2$. Indeed,

$$g_\beta(\alpha) = C_\beta + \frac{L}{2}\|\beta + \eta D^\top (x - D\beta) - \alpha\|_2^2.$$

and the update can be rewritten

$$\alpha \leftarrow \arg \min_{\alpha \in \mathbb{R}^p: \|\alpha\|_0 \leq k} g_\beta(\alpha)$$

$$= \Pi_{\|\cdot\|_0 \leq k} \left[\beta + \eta D^\top (x - D\beta)\right].$$
Sparse reconstruction with the $\ell_1$-norm

For the $\ell_0$-penalty, we have seen

1. a **coordinate descent** algorithm (matching pursuit);
2. a **gradient descent** algorithm (iterative hard-thresholding);
3. an **active-set** algorithm (orthogonal matching pursuit);

For $\ell_1$, the same three classes of methods play an important role.
Sparse reconstruction with the $\ell_1$-norm

Projected gradient descent

Suppose we want to solve

$$
\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \| x - D \alpha \|^2_2 \quad \text{s.t.} \quad \| \alpha_1 \|_1 \leq \mu.
$$
Sparse reconstruction with the $\ell_1$-norm

Projected gradient descent

Suppose we want to solve

$$
\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \|x - D\alpha\|^2_2 \quad \text{s.t.} \quad \|\alpha_1\|_1 \leq \mu.
$$

The following update with $\eta$ small enough converges to a solution

$$
\alpha \leftarrow \Pi_{\|\cdot\|_1 \leq \mu} \left[ \alpha + \eta D^\top (x - D\alpha) \right].
$$
Sparse reconstruction with the $\ell_1$-norm

Projected gradient descent

Suppose we want to solve

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \| x - D\alpha \|^2_2 \text{ s.t. } \|\alpha_1\|_1 \leq \mu.$$  

The following update with $\eta$ small enough converges to a solution

$$\alpha \leftarrow \Pi_{\|\cdot\|_1 \leq \mu} \left[ \alpha + \eta D^\top (x - D\alpha) \right].$$

Remarks

- the convergence rate is the same as the gradient descent method for smooth convex functions;
- when $L$ is unknown, efficient line-search scheme can be used.
- the principle is the same as for the iterative hard-thresholding algorithm.

Sparse reconstruction with the $\ell_1$-norm

The proximal gradient method

We consider a smooth convex function $f$ and a non-smooth regularizer $\psi$.

$$\min_{\alpha \in \mathbb{R}^p} f(\alpha) + \psi(\alpha)$$

We apply what we have seen in the optimization course for composite problems.
Sparse reconstruction with the $\ell_1$-norm

The proximal gradient method

Remarks

- also known as forward-backward algorithm;
- has similar convergence rates as the gradient descent method (proof is the same);
- there exists line search schemes to automatically tune $L$;

The case of $\ell_1$

The proximal operator of $\lambda \| . \|_1$ is the soft-thresholding operator

$$\alpha[j] = \text{sign}(\beta[j])(\|\beta[j]\| - \lambda)^+. $$

The resulting algorithm is called **iterative soft-thresholding** [Daubechies et al., 2004].
Sparse reconstruction with the $\ell_1$-norm

The proximal gradient method

The proximal operator for the group Lasso penalty

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \| \beta - \alpha \|_2^2 + \lambda \sum_{g \in \mathcal{G}} \| \alpha[g] \|_q.$$ 

For $q = 2$,

$$\alpha[g] = \frac{\beta[g]}{\| \beta[g] \|_2} (\| \beta[g] \|_2 - \lambda)^+, \quad \forall g \in \mathcal{G}.$$ 

For $q = \infty$,

$$\alpha[g] = \beta[g] - \Pi_{\| \cdot \|_1 \leq \lambda} \beta[g], \quad \forall g \in \mathcal{G}.$$ 

These formula generalize soft-thresholding to groups of variables.
Sparse reconstruction with the $\ell_1$-norm

The proximal gradient method

A few proximal operators:
- $\ell_0$-penalty: hard-thresholding;
- $\ell_1$-norm: soft-thresholding;
- group-Lasso: group soft-thresholding;
- fused-lasso (1D total variation): [Hoefling, 2010];
- **hierarchical norms**: [Jenatton et al., 2011b], $O(p)$ complexity;
- **overlapping group Lasso with $\ell_\infty$-norm**: [Mairal et al., 2010b], (link with network flow optimization);
Sparse reconstruction with the $\ell_1$-norm

Coordinate descent for the Lasso [Fu, 1998]

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \| x - D\alpha \|_2^2 + \lambda \| \alpha \|_1.$$

The coordinate descent method consists of iteratively fixing all variables and optimizing with respect to one:

$$\alpha[j] \leftarrow \arg \min_{\alpha \in \mathbb{R}} \frac{1}{2} \| x - \sum_{l \neq j} \alpha[l]d_l - \alpha d_j \|_2^2 + \lambda |\alpha|.$$

Assume the columns of $D$ to have unit $\ell_2$-norm,

$$\alpha_j \leftarrow \text{sign}(d_j^\top r)(|d_j^\top r| - \lambda)^+$$

This involves again the soft-thresholding operator.
Sparse reconstruction with the $\ell_1$-norm
Coordinate descent for the Lasso [Fu, 1998]

Remarks

- no parameter to tune!
- several strategies are possible for selecting the variable to update.
- impressive performance with five lines of code.
- coordinate descent + nonsmooth objective is not convergent in general. Here, the problem is equivalent to a convex smooth optimization problem with separable constraints

\[
\min_{\alpha_+, \alpha_-} \frac{1}{2} \| x - D_+ \alpha_+ + D_- \alpha_- \|_2^2 + \lambda \alpha_+^T 1 + \lambda \alpha_-^T 1 \quad \text{s.t.} \quad \alpha_-, \alpha_+ \geq 0.
\]

For this specific problem, the algorithm is convergent.
- can be extended to group-Lasso, or other loss functions.
- $j$ can be picked up at random, or by cycling (harder to analyze).
Sparse reconstruction with the $\ell_1$-norm

Smoothing techniques: reweighted $\ell_2$ [Daubechies et al., 2010, Bach et al., 2012]

Let us start from something simple

$$a^2 - 2ab + b^2 \geq 0.$$
Sparse reconstruction with the $\ell_1$-norm

Smoothing techniques: reweighted $\ell_2$ [Daubechies et al., 2010, Bach et al., 2012]

Let us start from something simple

$$a^2 - 2ab + b^2 \geq 0.$$  

Then

$$a \leq \frac{1}{2} \left( \frac{a^2}{b} + b \right) \text{ with equality iff } a = b$$

and

$$\|\alpha\|_1 = \min_{\eta_j \geq 0} \frac{1}{2} \sum_{j=1}^{p} \frac{\alpha[j]^2}{\eta_j} + \eta_j.$$  

The formulation becomes

$$\min_{\alpha, \eta_j \geq \varepsilon} \frac{1}{2} \|x - D\alpha\|_2^2 + \frac{\lambda}{2} \sum_{j=1}^{p} \frac{\alpha[j]^2}{\eta_j} + \eta_j.$$
Sparse reconstruction with the $\ell_1$-norm

Homotopy [Osborne et al., 2000b, Efron et al., 2004, Ritter, 1962]

Figure: The regularization path of the Lasso is piecewise linear.

\[
\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \| x - D\alpha \|_2^2 + \lambda \| \alpha \|_1.
\]

property discoved by Markowitz [1952].
Sparse reconstruction with the $\ell_1$-norm

Homotopy [Osborne et al., 2000b, Efron et al., 2004, Ritter, 1962]

**Theorem**

$\alpha$ is a solution of the Lasso if and only if

$$\begin{cases} 
|d_j^\top (x - D\alpha)| & \leq \lambda \text{ if } \alpha[j] = 0 \\
 d_j^\top (x - D\alpha) & = \lambda \text{sign}(\alpha[j]) \text{ otherwise.}
\end{cases}$$

**Consequence**

$$\alpha^*[\Gamma] = (D_\Gamma^T D_\Gamma)^{-1}(D_\Gamma^T x - \lambda \text{sign}(\alpha^*[\Gamma])) = A + \lambda B,$$

where $\Gamma = \{j \text{ s.t. } \alpha[j] \neq 0\}$. If we know $\Gamma$ and the signs of $\alpha^*$ in advance, we have a closed form solution.

Following the piecewise linear regularization path is called the **homotopy** method [Osborne et al., 2000a, Efron et al., 2004].
Sparse reconstruction with the $\ell_1$-norm

Homotopy [Osborne et al., 2000b, Efron et al., 2004, Ritter, 1962]

The regularization path $(\lambda, \alpha^*(\lambda))$ is piecewise linear.

1. Start from the trivial solution $(\lambda = \|D^T x\|_\infty, \alpha^*(\lambda) = 0)$.
2. Define $\Gamma = \{j \text{ s.t. } |d_j^T x| = \lambda\}$,
3. Follow the regularization path: $\alpha^*_\Gamma(\lambda) = A + \lambda B$, keeping $\alpha^*_\Gamma_c = 0$, decreasing the value of $\lambda$, until one of the following event occurs:
   - $\exists j \notin \Gamma$ such that $|d_j^T (x - D\alpha^*(\lambda))| = \lambda$, then $\Gamma \leftarrow \Gamma \cup \{j\}$.
   - $\exists j \in \Gamma$ such that $\alpha^*(\lambda) = 0$, then $\Gamma \leftarrow \Gamma \setminus \{j\}$.
4. Update the direction of the path and go back to 3.

Hidden assumptions

- the regularization path is unique.
- variables enter the path one at a time.

Extremely efficient for small/medium scale problems ($p \leq 10\,000$) and/or very sparse problems (when implemented correctly). Robust to correlated features. Can solve the elastic-net.
Sparse reconstruction with the $\ell_1$-norm
Homotopy [Osborne et al., 2000b, Efron et al., 2004, Ritter, 1962]

Theorem - worst case analysis [Mairal and Yu, 2012]

*In the worst-case, the regularization path of the Lasso has exactly $(3^p + 1)/2$ linear segments.*
Sparse reconstruction with the $\ell_1$-norm

Lasso empirical comparison: Lasso, small scale ($n = 200, p = 200$)
Sparse reconstruction with the $\ell_1$-norm

Empirical comparison: Lasso, medium scale ($n = 2000, p = 10000$)

- **reg: low**
  - corr: low
  - corr: high

- **reg: high**
  - corr: low
  - corr: high

---

Julien Mairal
Sparse Estimation for Image and Vision Processing 166/181
Sparse reconstruction with the $\ell_1$-norm

Empirical comparison: conclusions

Lasso

- Generic methods (subgradient descent, QP/CP solvers) are slow;
- homotopy fastest in **low dimension** and/or for **high correlation**
- Proximal methods are competitive
  - esp. larger setting and/or weak corr. and/or weak reg. and/or low precision
- Coordinate descent
  - usually dominated by LARS;
  - but much simpler to implement!

Smooth Losses and other regularization

- LARS not available $\rightarrow$ (block) coordinate descent, proximal gradient methods are good candidates.
Iterative reweighted $\ell_1$-algorithms

DC (difference of convex) - Programming

Remember? Concave functions with a kink at zero

$$\psi(\alpha) = \sum_{j=1}^{P} \varphi(|\alpha[j]|).$$

- $\ell_q$-“pseudo-norm”, with $0 < q < 1$: $\psi(w) \triangleq \sum_{j=1}^{P} (|\alpha[j]| + \varepsilon)^q$,
- log penalty, $\psi(w) \triangleq \sum_{j=1}^{P} \log(|\alpha[j]| + \varepsilon),$

$\varphi$ is any function that looks like this:

\[ \varphi(\alpha) \]
\[ f(\alpha) \]

\[ \varphi(\alpha) = \log(|\alpha| + \varepsilon) \]

\[ f(\alpha) + \varphi(|\alpha|) \]

\[ \varphi'(|\alpha^t|)|\alpha| + C \]

\[ |\alpha^t| \]

\[ |\alpha^k| \]
DC (difference of convex) - Programming

\[ \min_{\alpha \in \mathbb{R}^p} f(\alpha) + \lambda \sum_{j=1}^{p} \varphi(|\alpha[j]|). \]

This problem is non-convex. \( f \) is convex, and \( \varphi \) is concave on \( \mathbb{R}^+ \).

If \( \alpha^k \) is the current estimate at iteration \( t \), the algorithm solves

\[ \alpha^{t+1} \leftarrow \arg \min_{\alpha \in \mathbb{R}^p} \left[ f(\alpha) + \lambda \sum_{j=1}^{p} \varphi'(|\alpha^t[j]|)|\alpha[j]| \right], \]

which is a **reweighted-\( \ell_1 \)** problem [Candès et al., 2008].

**Warning:** It does not solve the non-convex problem, only provides a stationary point.

In practice, each iteration sets to zero small coefficients. After 2 – 3 iterations, the result does not change much.
Optimization for Dictionary Learning

\[ \min_{\alpha \in \mathbb{R}^{p \times n}} \sum_{i=1}^{n} \frac{1}{2} \| x_i - D \alpha_i \|_2^2 + \lambda \psi(\alpha_i) \]

\[ \mathcal{C} \triangleq \{ D \in \mathbb{R}^{m \times p} \text{ s.t. } \forall j = 1, \ldots, p, \| d_j \|_2 \leq 1 \}. \]

Classical approach

- Alternate minimization between \( D \) and \( \alpha \) (MOD with \( \psi = \ell_0 \) [Engan et al., 1999], K-SVD with \( \psi = \ell_0 \) [Aharon et al., 2006], [Lee et al., 2007] with \( \psi = \ell_1 \));

- good results, reliable, but can be slow when \( n \) is large!
Optimization for Dictionary Learning

Empirical risk minimization point of view

$$\min_{D \in C} f_n(D) = \min_{D \in C} \frac{1}{n} \sum_{i=1}^{n} L(x_i, D),$$

where

$$L(x, D) \triangleq \min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \|x - D\alpha\|_2^2 + \lambda \psi(\alpha).$$

Which formulation are we interested in?

$$\min_{D \in C} \left\{ f(D) = \mathbb{E}_x[L(x, D)] \approx \lim_{n \to +\infty} \frac{1}{n} \sum_{i=1}^{n} L(x_i, D) \right\}$$

[Bottou and Bousquet, 2008]: Online learning can

- handle potentially infinite or dynamic datasets,
- be dramatically faster than batch algorithms.
Recipe

- draw a single point $\mathbf{x}_t$ (or a mini-batch) at each iteration;
- update

\[
\mathbf{D} \leftarrow \Pi_C [\mathbf{D} - \eta_t \nabla_D L(\mathbf{x}_t, \mathbf{D})],
\]

which is equivalent (up to some assumptions) to

\[
\alpha_t \leftarrow \arg \min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \| \mathbf{x}_t - \mathbf{D}\alpha \|^2_2 + \lambda \| \alpha \|_1,
\]

\[
\mathbf{D} \leftarrow \Pi_C [\mathbf{D} + \eta_t (\mathbf{x}_t - \mathbf{D}\alpha_t) \alpha_t^\top].
\]
Optimization for Dictionary Learning

Stochastic gradient descent

Recipe

- draw a single point \( x_t \) (or a mini-batch) at each iteration;
- update

\[
D \leftarrow \Pi_C[D - \eta_t \nabla_D L(x_t, D)],
\]

which is equivalent (up to some assumptions) to

\[
\alpha_t \leftarrow \arg \min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \|x_t - D\alpha\|^2_2 + \lambda \|\alpha\|_1,
\]

\[
D \leftarrow \Pi_C[D + \eta_t (x_t - D\alpha_t)\alpha_t^\top].
\]

Remark

- historically, this is very close to the original algorithm of Olshausen and Field [1996].
Optimization for Dictionary Learning

Stochastic gradient descent

Recipe

• draw a single point $x_t$ (or a mini-batch) at each iteration;
• update

$$D \leftarrow \Pi_C[D - \eta_t \nabla_D L(x_t, D)],$$

which is equivalent (up to some assumptions) to

$$\alpha_t \leftarrow \arg \min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \|x_t - D\alpha\|_2^2 + \lambda \|\alpha\|_1,$$

$$D \leftarrow \Pi_C[D + \eta_t (x_t - D\alpha_t)\alpha_t^\top].$$

Pros and cons

😊 can be effective in practice;
😊 difficult to tune.
Optimization for Dictionary Learning
Online dictionary learning [Mairal et al., 2010a]

Recipe

- stochastic majorization-minimization algorithm;
- relies on a fast dictionary update;
- easier to tune (the implementation of SPAMS has been successfully used by others in plenty of “exotic” unexpected scenarios.)
Optimization for Dictionary Learning

Online dictionary learning [Mairal et al., 2010a]

Require: $D_0 \in \mathbb{R}^{m \times p}$ (initial dictionary); $\lambda \in \mathbb{R}$

1: $C_0 = 0$, $B_0 = 0$.
2: for $t=1,\ldots,T$ do
3: Draw $x_t$
4: Sparse Coding: $\alpha_t \leftarrow \arg \min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \|x_t - D_{t-1} \alpha\|_2^2 + \lambda \|\alpha\|_1$,
5: Aggregate sufficient statistics 
   $C_t \leftarrow C_{t-1} + \alpha_t \alpha_t^T$, $B_t \leftarrow B_{t-1} + x_t \alpha_t^T$
6: Dictionary update 
   \[ D_t \leftarrow \arg \min_{D \in \mathcal{C}} \frac{1}{t} \sum_{i=1}^{t} \left( \frac{1}{2} \|x_i - D \alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \right). \]
   \[ = \arg \min_{D \in \mathcal{C}} \frac{1}{t} \left( \frac{1}{2} \text{Tr}(D^T D C_t) - \text{Tr}(D^T B_t) \right). \]
7: end for
Optimization for Dictionary Learning

Fast dictionary update [Mairal et al., 2010a]

Require: $D_0 \in \mathcal{C}$ (input dictionary); $X \in \mathbb{R}^{m \times n}$ (dataset); $A \in \mathbb{R}^{p \times n}$ (sparse codes);

1: Initialization: $D \leftarrow D_0$; $B \leftarrow XA^\top$; $C \leftarrow AA^\top$;
2: repeat
3: for $j = 1, \ldots, p$ do
4: update the $j$-th column:

\[
\begin{align*}
  &d_j \leftarrow \frac{1}{C[j,j]}(b_j - Dc_j) + d_j, \\
  &d_j \leftarrow \frac{1}{\max(\|d_j\|_2, 1)}d_j.
\end{align*}
\]
5: end for
6: until convergence;
7: return $D$ (updated dictionary).
Minimizing with respect to one column $d_j$ when keeping the other columns fixed can be formulated as

$$d_j \leftarrow \arg \min_{d \in \mathbb{R}^m, \|d\| \leq 1} \left[ \sum_{i=1}^{n} \frac{1}{2} \left\| x_i - \sum_{l \neq j} \alpha_i[l] d_l - \alpha_i[j] d \right\|_2^2 \right].$$
Minimizing with respect to one column $d_j$ when keeping the other columns fixed can be formulated as

$$d_j \leftarrow \arg \min_{d \in \mathbb{R}^m, \|d\|_2 \leq 1} \left[ \sum_{i=1}^{n} \frac{1}{2} \left\| x_i - \sum_{l \neq j} \alpha_i[l] d_l - \alpha_i[j] d_j \right\|_2^2 \right].$$

Then, in a matrix form

$$d_j \leftarrow \arg \min_{d \in \mathbb{R}^m, \|d\|_2 \leq 1} \left[ \frac{1}{2} \left\| X - DA + d_j \alpha^j - d \alpha^j \right\|_F^2 \right],$$
Optimization for Dictionary Learning

Fast dictionary udpate [Mairal et al., 2010a]

Minimizing with respect to one column $d_j$ when keeping the other columns fixed can be formulated as

$$
d_j \leftarrow \arg \min_{d \in \mathbb{R}^m, \|d\|_2 \leq 1} \left[ \sum_{i=1}^{n} \frac{1}{2} \left\| x_i - \sum_{l \neq j} \alpha_i[l] d_l - \alpha_i[j] d_j \right\|^2 \right].
$$

After expanding the Frobenius norm and removing the constant term,

$$
d_j \leftarrow \arg \min_{d \in \mathbb{R}^m, \|d\|_2 \leq 1} \left[ -d^\top (X - DA + d_j \alpha^j) \alpha^{j\top} + \frac{1}{2} \|d \alpha^j\|_F^2 \right]
$$

$$
= \arg \min_{d \in \mathbb{R}^m, \|d\|_2 \leq 1} \left[ -d^\top (b_j - Dc_j + d_j C[j,j]) + \frac{1}{2} \|d\|_2^2 C[j,j] \right]
$$

$$
= \arg \min_{d \in \mathbb{R}^m, \|d\|_2 \leq 1} \left[ \frac{1}{2} \left\| \frac{1}{C[j,j]} (b_j - Dc_j) + d_j - d \right\|_2^2 \right],
$$
Optimization for Dictionary Learning
[Mairal et al., 2010a]

Which guarantees do we have?

Under a few reasonable assumptions,

- we build a surrogate function \( \hat{g}_t \) of the expected cost \( f \) verifying

\[
\lim_{t \to +\infty} \hat{g}_t(D_t) - f(D_t) = 0,
\]

- \( D_t \) is asymptotically close to a stationary point.

Extensions (all implemented in SPAMS)

- non-negative matrix decompositions;
- sparse PCA (sparse dictionaries);
- fused-lasso regularizations (piecewise constant dictionaries);
- non-convex regularization, structured regularization.
Optimization for Dictionary Learning
Experimental results, batch vs online

\[ m = 8 \times 8, \ p = 256 \]
Optimization for Dictionary Learning
Experimental results, batch vs online

\[ m = 12 \times 12 \times 3, \quad p = 512 \]
Conclusions from the fourth part

- there are a few algorithms for sparse estimation that are efficient and easy to implement;
- there is no algorithm that wins all the time;
- designing an evaluation benchmark that makes sense is hard.
Conclusions from the fourth part

- there are a few algorithms for sparse estimation that are efficient and easy to implement;
- there is no algorithm that wins all the time;
- designing an evaluation benchmark that makes sense is hard.

What was not covered

- stochastic optimization for sparse estimation;
- proximal splitting algorithms.

Advertisement again

- the SPAMS toolbox already contains lots of code (C++ interfaced with Matlab, Python, R) for learning dictionaries, factorizing matrices (NMF, archetypal analysis), solving sparse estimation problems, including most of the algorithms we have presented.
References I


References II


References V


References VI


References IX


References


References XI


References XII


References XIV


References XV


References XVII


References XVIII


References XIX


References XX


References XXI


References XXII


References XXIII


References XXV


T. Young. *A course of lectures on natural philosophy and the mechanical art*. Taylor and Watson, 1845.


References XXVIII


Appendix
Basic convex optimization tools: subgradients

\( \partial f(\alpha) \triangleq \{ \kappa \in \mathbb{R}^p \mid f(\alpha) + \kappa^\top (\alpha' - \alpha) \leq f(\alpha') \text{ for all } \alpha' \in \mathbb{R}^p \} \).

Figure: Gradients and subgradients for smooth and non-smooth functions.
Basic convex optimization tools: subgradients

Some nice properties

- $\partial f(\alpha) = \{g\}$ iff $f$ differentiable at $\alpha$ and $g = \nabla f(\alpha)$.
- many calculus rules: $\partial(\gamma f + \mu g) = \gamma \partial f + \mu \partial g$ for $\gamma, \mu > 0$.

for more details, see Boyd and Vandenberghe [2004], Bertsekas [1999], Borwein and Lewis [2006] and S. Boyd’s course at Stanford.

Optimality conditions

For $g : \mathbb{R}^p \rightarrow \mathbb{R}$ convex,

- $g$ differentiable: $\alpha^*$ minimizes $g$ iff $\nabla g(\alpha^*) = 0$.
- $g$ nondifferentiable: $\alpha^*$ minimizes $g$ iff $0 \in \partial g(\alpha^*)$.

Careful: the concept of subgradient requires a function to be above its tangents. It does only make sense for convex functions!
Basic convex optimization tools: dual-norm

**Definition**
Let $\kappa$ be in $\mathbb{R}^p$,

$$\|\kappa\|_* \triangleq \max_{\alpha \in \mathbb{R}^p : \|\alpha\| \leq 1} \alpha^\top \kappa.$$

**Exercises**

- $\|\alpha\|_{**} = \|\alpha\|$ (true in finite dimension)
- $\ell_2$ is dual to itself.
- $\ell_1$ and $\ell_\infty$ are dual to each other.
- $\ell_q$ and $\ell'_q$ are dual to each other if $\frac{1}{q} + \frac{1}{q'} = 1$.
- Similar relations for spectral norms on matrices.
- $\partial \|\alpha\| = \{ \kappa \in \mathbb{R}^p \text{ s.t. } \|\kappa\|_* \leq 1 \text{ and } \kappa^\top \alpha = \|\alpha\| \}$. 
Optimality conditions

Let $f : \mathbb{R}^p \rightarrow \mathbb{R}$ be convex differentiable and $\| \cdot \|$ be any norm.

$$\min_{\alpha \in \mathbb{R}^p} f(\alpha) + \lambda \| \alpha \|.$$  

$\alpha$ is solution if and only if

$$0 \in \partial (f(\alpha) + \lambda \| \alpha \|) = \nabla f(\alpha) + \lambda \partial \| \alpha \|$$

Since $\partial \| \alpha \| = \{ \kappa \in \mathbb{R}^p \text{ s.t. } \| \kappa \|_* \leq 1 \text{ and } \kappa^\top \alpha = \| \alpha \| \}$,

**General optimality conditions:**

$$\| \nabla f(\alpha) \|_* \leq \lambda \text{ and } - \nabla f(\alpha)^\top \alpha = \lambda \| \alpha \|. \quad \Box$$
Convex Duality

Strong Duality

Strong duality means that \[ \max_\kappa g(\kappa) = \min_\alpha f(\alpha) \]
Convex Duality

Duality Gaps

Strong duality means that \( \max_\kappa g(\kappa) = \min_\alpha f(\alpha) \)

The duality gap guarantees us that \( 0 \leq f(\tilde{\alpha}) - f(\alpha^*) \leq \delta(\tilde{\alpha}, \tilde{\kappa}) \).