

Sparse Coding and Dictionary Learning for Image Analysis

Part III: Optimization for Sparse Coding and Dictionary Learning

Francis Bach, Julien Mairal, Jean Ponce and Guillermo Sapiro

CVPR'10 tutorial, San Francisco, 14th June 2010

The Sparse Decomposition Problem

$$\min_{\alpha \in \mathbb{R}^p} \underbrace{\frac{1}{2} \|\mathbf{x} - \mathbf{D}\alpha\|_2^2}_{\text{data fitting term}} + \underbrace{\lambda \psi(\alpha)}_{\text{sparsity-inducing regularization}}$$

ψ induces sparsity in α . It can be

- the ℓ_0 “pseudo-norm”. $\|\alpha\|_0 \triangleq \#\{i \text{ s.t. } \alpha[i] \neq 0\}$ (NP-hard)
- the ℓ_1 norm. $\|\alpha\|_1 \triangleq \sum_{i=1}^p |\alpha[i]|$ (convex)
- ...

This is a **selection** problem.

Finding your way in the sparse coding literature. . .

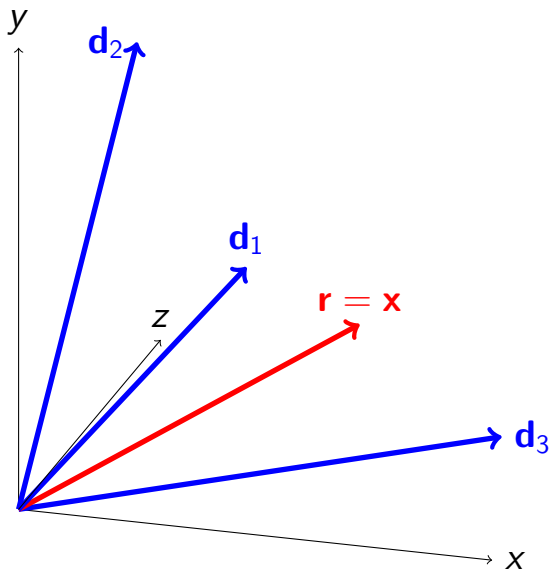
. . . is not easy. The literature is vast, redundant, sometimes confusing and many papers are claiming victory. . .

The main class of methods are

- **greedy** procedures [Mallat and Zhang, 1993], [Weisberg, 1980]
- **homotopy** [Osborne et al., 2000], [Efron et al., 2004], [Markowitz, 1956]
- **soft-thresholding** based methods [Fu, 1998], [Daubechies et al., 2004], [Friedman et al., 2007], [Nesterov, 2007], [Beck and Teboulle, 2009], . . .
- reweighted- ℓ_2 methods [Daubechies et al., 2009], . . .
- active-set methods [Roth and Fischer, 2008].
- . . .

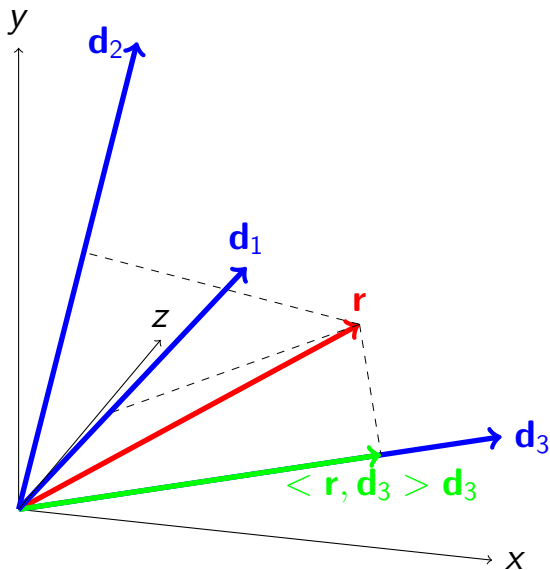
Matching Pursuit

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



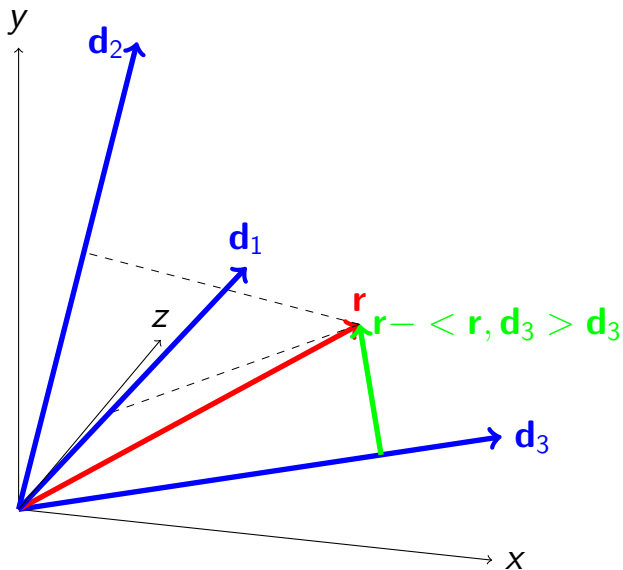
Matching Pursuit

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



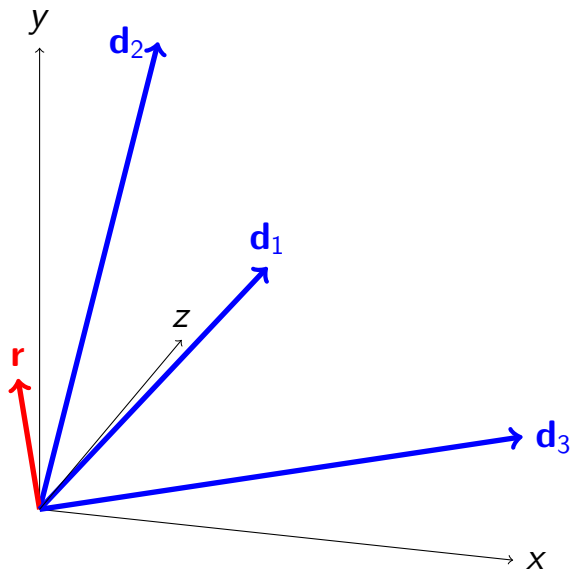
Matching Pursuit

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



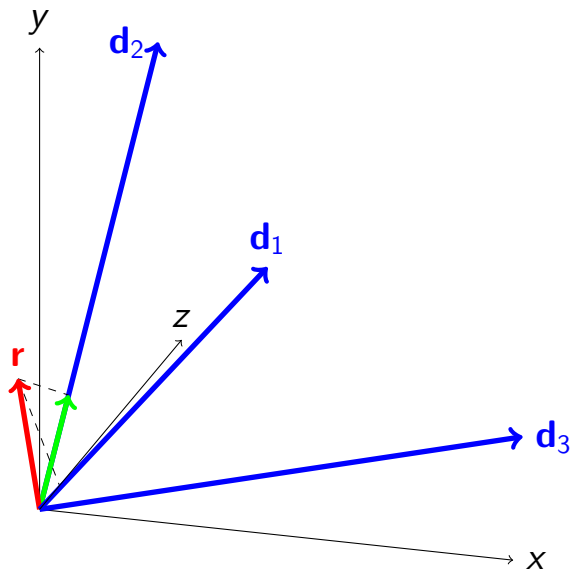
Matching Pursuit

$$\alpha = (0, 0, 0.75)$$



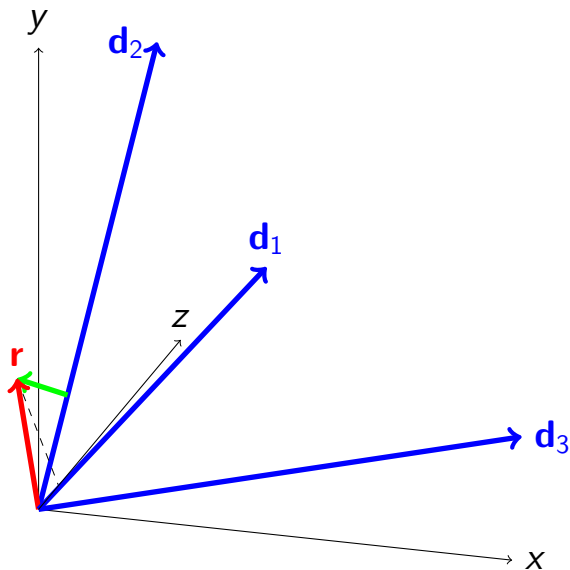
Matching Pursuit

$$\alpha = (0, 0, 0.75)$$



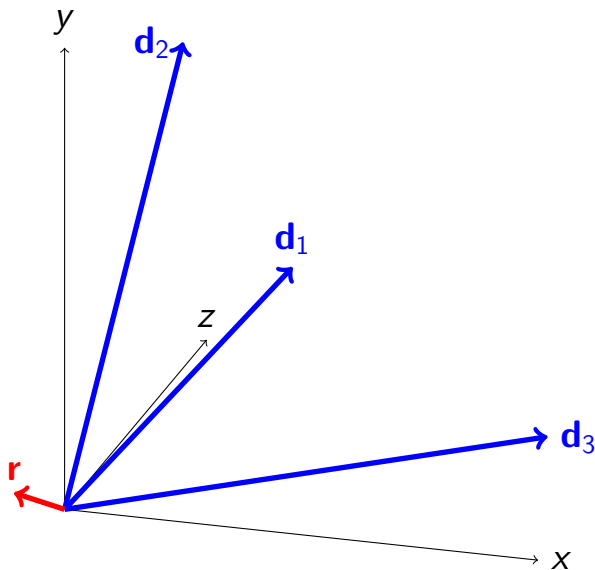
Matching Pursuit

$$\alpha = (0, 0, 0.75)$$



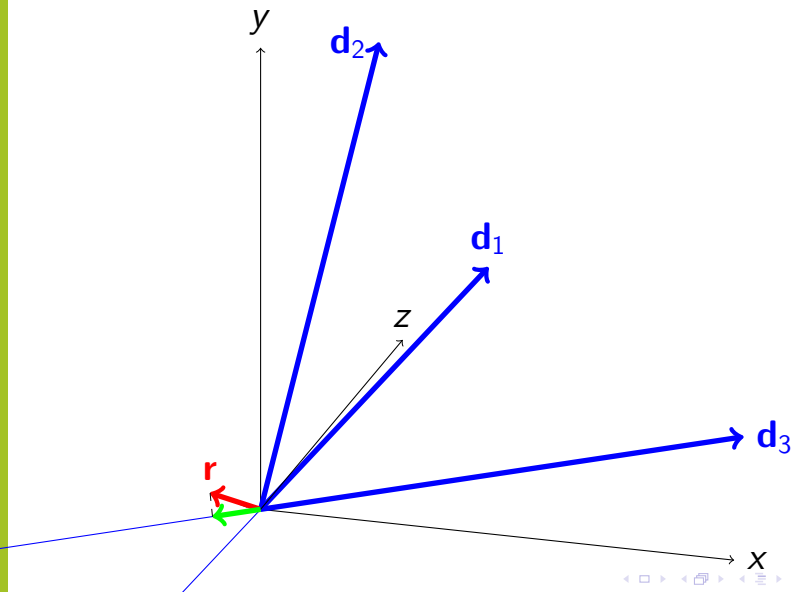
Matching Pursuit

$$\alpha = (0, 0.24, 0.75)$$



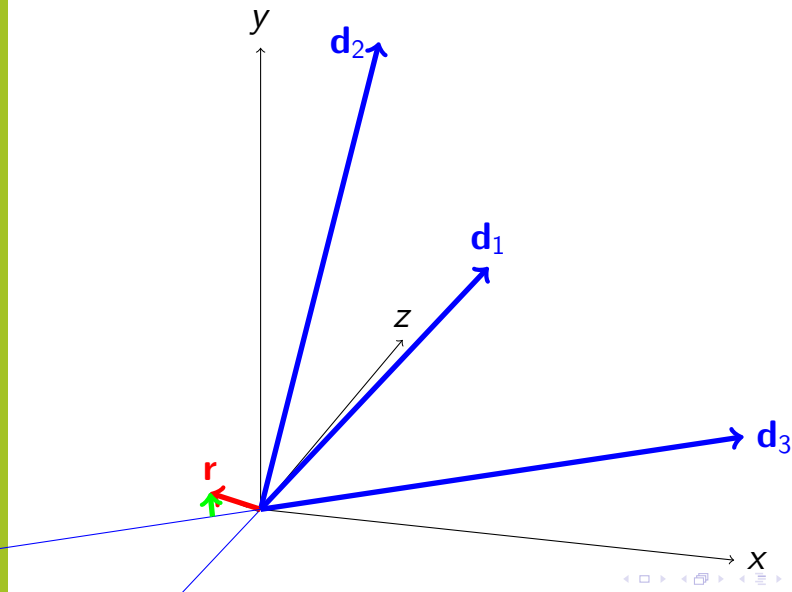
Matching Pursuit

$$\alpha = (0, 0.24, 0.75)$$



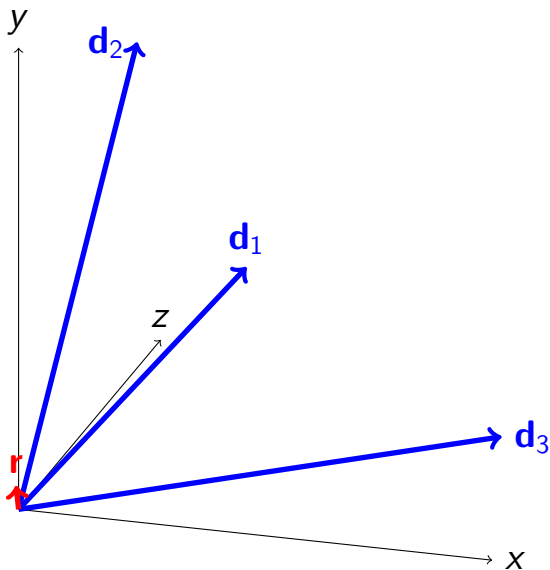
Matching Pursuit

$$\alpha = (0, 0.24, 0.75)$$



Matching Pursuit

$$\alpha = (0, 0.24, 0.65)$$



Matching Pursuit

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

- 1: $\alpha \leftarrow 0$
- 2: $\mathbf{r} \leftarrow \mathbf{x}$ (residual).
- 3: **while** $\|\alpha\|_0 < L$ **do**
- 4: Select the atom with maximum correlation with the residual

$$\hat{i} \leftarrow \arg \max_{i=1, \dots, p} |\mathbf{d}_i^T \mathbf{r}|$$

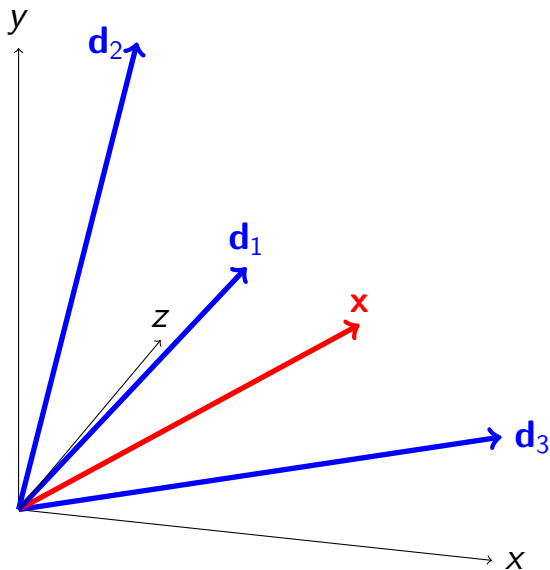
- 5: Update the residual and the coefficients

$$\begin{aligned} \alpha[\hat{i}] &\leftarrow \alpha[\hat{i}] + \mathbf{d}_{\hat{i}}^T \mathbf{r} \\ \mathbf{r} &\leftarrow \mathbf{r} - (\mathbf{d}_{\hat{i}}^T \mathbf{r}) \mathbf{d}_{\hat{i}} \end{aligned}$$

- 6: **end while**

Orthogonal Matching Pursuit

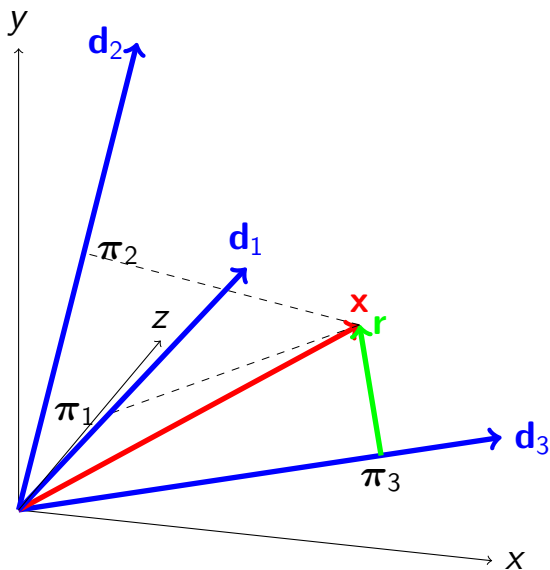
$$\alpha = (0, 0, 0)$$
$$\Gamma = \emptyset$$



Orthogonal Matching Pursuit

$$\alpha = (0, 0, 0.75)$$

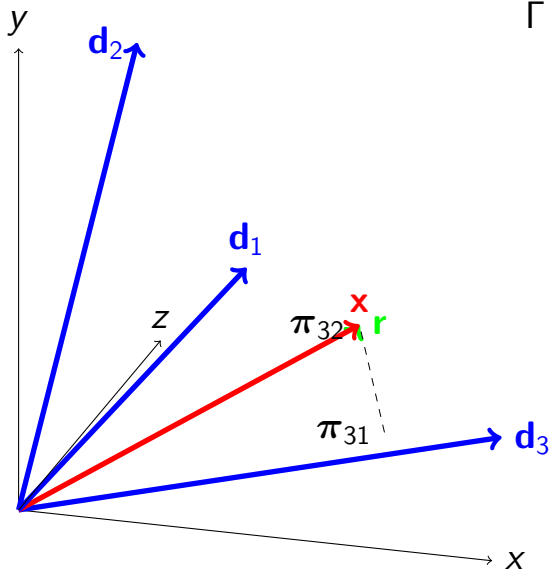
$$\Gamma = \{3\}$$



Orthogonal Matching Pursuit

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

$$\Gamma = \{3, 2\}$$



Orthogonal Matching Pursuit

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

- 1: $\Gamma = \emptyset$.
- 2: **for** $iter = 1, \dots, L$ **do**
- 3: Select the atom which most reduces the objective

$$\hat{i} \leftarrow \arg \min_{i \in \Gamma^c} \left\{ \min_{\alpha'} \|\mathbf{x} - \mathbf{D}_{\Gamma \cup \{i\}} \alpha'\|_2^2 \right\}$$

- 4: Update the active set: $\Gamma \leftarrow \Gamma \cup \{\hat{i}\}$.
- 5: Update the residual (orthogonal projection)

$$\mathbf{r} \leftarrow (\mathbf{I} - \mathbf{D}_\Gamma (\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma)^{-1} \mathbf{D}_\Gamma^T) \mathbf{x}.$$

- 6: Update the coefficients

$$\alpha_\Gamma \leftarrow (\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma)^{-1} \mathbf{D}_\Gamma^T \mathbf{x}.$$

- 7: **end for**

Orthogonal Matching Pursuit

Contrary to MP, an atom can only be selected one time with OMP. It is, however, more difficult to implement efficiently. The keys for a good implementation in the case of a large number of signals are

- Precompute the Gram matrix $\mathbf{G} = \mathbf{D}^T \mathbf{D}$ once in for all,
- Maintain the computation of $\mathbf{D}^T \mathbf{r}$ for each signal,
- Maintain a Cholesky decomposition of $(\mathbf{D}_r^T \mathbf{D}_r)^{-1}$ for each signal.

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

$$\underbrace{O(p^2 m)}_{\text{Gram matrix}} + \underbrace{O(nL^3)}_{\text{Cholesky}} + \underbrace{O(n(pm + pL^2))}_{\mathbf{D}^T \mathbf{r}} = O(np(m + L^2))$$

It is also possible to use the matrix inversion lemma instead of a Cholesky decomposition (same complexity, but less numerical stability)

Example with the software SPAMS

Software available at <http://www.di.ens.fr/willow/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 a 8-cores 2.83Ghz machine: **23000 signals processed per second!**

Optimality conditions of the Lasso

Nonsmooth optimization

Directional derivatives and subgradients are useful tools for studying ℓ_1 -decomposition problems:

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

In this tutorial, we use the **directional derivatives** to derive simple optimality conditions of the Lasso.

For more information on convex analysis and nonsmooth optimization, see the following books: [Boyd and Vandenberghe, 2004], [Nocedal and Wright, 2006], [Borwein and Lewis, 2006], [Bonnans et al., 2006], [Bertsekas, 1999].

Optimality conditions of the Lasso

Directional derivatives

- **Directional derivative** in the direction \mathbf{u} at α :

$$\nabla f(\alpha, \mathbf{u}) = \lim_{t \rightarrow 0^+} \frac{f(\alpha + t\mathbf{u}) - f(\alpha)}{t}$$

- Main idea: in non smooth situations, one may need to look at all directions \mathbf{u} and not simply p independent ones!
- **Proposition 1:** if f is differentiable in α , $\nabla f(\alpha, \mathbf{u}) = \nabla f(\alpha)^T \mathbf{u}$.
- **Proposition 2:** α is optimal iff for all \mathbf{u} in \mathbb{R}^p , $\nabla f(\alpha, \mathbf{u}) \geq 0$.

Optimality conditions of the Lasso

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

α^* is optimal iff for all \mathbf{u} in \mathbb{R}^p , $\nabla f(\alpha, \mathbf{u}) \geq 0$ —that is,

$$-\mathbf{u}^T \mathbf{D}^T (\mathbf{x} - \mathbf{D}\alpha^*) + \lambda \sum_{i, \alpha^*[i] \neq 0} \text{sign}(\alpha^*[i]) \mathbf{u}[i] + \lambda \sum_{i, \alpha^*[i] = 0} |\mathbf{u}[i]| \geq 0,$$

which is equivalent to the following conditions:

$$\forall i = 1, \dots, p, \quad \begin{cases} |\mathbf{d}_i^T (\mathbf{x} - \mathbf{D}\alpha^*)| \leq \lambda & \text{if } \alpha^*[i] = 0 \\ \mathbf{d}_i^T (\mathbf{x} - \mathbf{D}\alpha^*) = \lambda \text{sign}(\alpha^*[i]) & \text{if } \alpha^*[i] \neq 0 \end{cases}$$

Homotopy

- A homotopy method provides a set of solutions indexed by a parameter.
- The regularization path $(\lambda, \alpha^*(\lambda))$ for instance!!
- It can be useful when the path has some “nice” properties (piecewise linear, piecewise quadratic).
- LARS [Efron et al., 2004] starts from a trivial solution, and follows the regularization path of the Lasso, which is **piecewise linear**.

Homotopy, LARS

[Osborne et al., 2000], [Efron et al., 2004]

$$\forall i = 1, \dots, p, \quad \begin{cases} |\mathbf{d}_i^T(\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}^*)| \leq \lambda & \text{if } \alpha^*[i] = 0 \\ \mathbf{d}_i^T(\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}^*) = \lambda \text{sign}(\alpha^*[i]) & \text{if } \alpha^*[i] \neq 0 \end{cases} \quad (1)$$

The regularization path is piecewise linear:

$$\mathbf{D}_\Gamma^T(\mathbf{x} - \mathbf{D}_\Gamma\boldsymbol{\alpha}_\Gamma^*) = \lambda \text{sign}(\boldsymbol{\alpha}_\Gamma^*)$$

$$\boldsymbol{\alpha}_\Gamma^*(\lambda) = (\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma)^{-1}(\mathbf{D}_\Gamma^T \mathbf{x} - \lambda \text{sign}(\boldsymbol{\alpha}_\Gamma^*)) = \mathbf{A} + \lambda \mathbf{B}$$

A simple interpretation of LARS

- Start from the trivial solution ($\lambda = \|\mathbf{D}^T \mathbf{x}\|_\infty, \boldsymbol{\alpha}^*(\lambda) = 0$).
- Maintain the computations of $|\mathbf{d}_i^T(\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}^*(\lambda))|$ for all i .
- Maintain the computation of the current direction \mathbf{B} .
- Follow the path by reducing λ until the next kink.

Example with the software SPAMS

<http://www.di.ens.fr/willow/SPAMS/>

```
>> I=double(imread('data/lena.eps'))/255;
>> %extract all patches of I
>> X=normalize(im2col(I,[8 8],'sliding'));
>> %load a dictionary of size 64 x 256
>> D=load('dict.mat');
>>
>> %set the sparsity parameter lambda to 0.15
>> param.lambda=0.15;
>> alpha=mexLasso(X,D,param);
```

On a 8-cores 2.83Ghz machine: **77000 signals processed per second!**
Note that it can also solve **constrained** version of the problem. The complexity is more or less the same as OMP and uses the same tricks (Cholesky decomposition).

Coordinate Descent

- Coordinate descent + nonsmooth objective: **WARNING: not convergent in general**
- Here, the problem is equivalent to a convex smooth optimization problem with **separable** constraints

$$\min_{\alpha_+, \alpha_-} \frac{1}{2} \|\mathbf{x} - \mathbf{D}_+ \alpha_+ + \mathbf{D}_- \alpha_-\|_2^2 + \lambda \alpha_+^T \mathbf{1} + \lambda \alpha_-^T \mathbf{1} \quad \text{s.t.} \quad \alpha_-, \alpha_+ \geq 0.$$

- For this **specific** problem, coordinate descent is **convergent**.
- Supposing $\|\mathbf{d}_i\|_2 = 1$, updating the coordinate i :

$$\begin{aligned} \alpha[i] &\leftarrow \arg \min_{\beta} \frac{1}{2} \left\| \mathbf{x} - \underbrace{\sum_{j \neq i} \alpha[j] \mathbf{d}_j}_{\mathbf{r}} - \beta \mathbf{d}_i \right\|_2^2 + \lambda |\beta| \\ &\leftarrow \text{sign}(\mathbf{d}_i^T \mathbf{r}) (|\mathbf{d}_i^T \mathbf{r}| - \lambda)^+ \end{aligned}$$

- \Rightarrow **soft-thresholding!**

Example with the software SPAMS

<http://www.di.ens.fr/willow/SPAMS/>

```
>> I=double(imread('data/lena.eps'))/255;
>> %extract all patches of I
>> X=normalize(im2col(I,[8 8],'sliding'));
>> %load a dictionary of size 64 x 256
>> D=load('dict.mat');
>>
>> %set the sparsity parameter lambda to 0.15
>> param.lambda=0.15;
>> param.tol=1e-2;
>> param.itermax=200;
>> alpha=mexCD(X,D,param);
```

On a 8-cores 2.83Ghz machine: **93000 signals processed per second!**

first-order/proximal methods

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

- f is strictly convex and continuously differentiable with a Lipschitz gradient.
- Generalize the idea of gradient descent

$$\begin{aligned}\alpha_{k+1} &\leftarrow \arg \min_{\alpha \in \mathbb{R}} f(\alpha_k) + \nabla f(\alpha_k)^T (\alpha - \alpha_k) + \frac{L}{2} \|\alpha - \alpha_k\|_2^2 + \lambda\psi(\alpha) \\ &\leftarrow \arg \min_{\alpha \in \mathbb{R}} \frac{1}{2} \|\alpha - (\alpha_k - \frac{1}{L} \nabla f(\alpha_k))\|_2^2 + \frac{\lambda}{L} \psi(\alpha)\end{aligned}$$

When $\lambda = 0$, this is equivalent to a classical gradient descent step.

first-order/proximal methods

- They require solving efficiently the proximal operator

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{u} - \alpha\|_2^2 + \lambda \psi(\alpha)$$

- For the ℓ_1 -norm, this amounts to a soft-thresholding:

$$\alpha^*[i] = \text{sign}(\mathbf{u}[i])(\mathbf{u}[i] - \lambda)^+.$$

- There exists accelerated versions based on Nesterov optimal first-order method (gradient method with “extrapolation”) [Beck and Teboulle, 2009, Nesterov, 2007, 1983]
- suited for large-scale experiments.

Optimization for Grouped Sparsity

The formulation:

$$\min_{\alpha \in \mathbb{R}^p} \underbrace{\frac{1}{2} \|\mathbf{x} - \mathbf{D}\alpha\|_2^2}_{\text{data fitting term}} + \underbrace{\lambda \sum_{g \in \mathcal{G}} \|\alpha_g\|_q}_{\text{group-sparsity-inducing regularization}}$$

The main class of algorithms for solving grouped-sparsity problems are

- Greedy approaches
- Block-coordinate descent
- Proximal methods

Optimization for Grouped Sparsity

The proximal operator:

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{u} - \alpha\|_2^2 + \lambda \sum_{g \in \mathcal{G}} \|\alpha_g\|_q$$

For $q = 2$,

$$\alpha_g^* = \frac{\mathbf{u}_g}{\|\mathbf{u}_g\|_2} (\|\mathbf{u}_g\|_2 - \lambda)^+, \quad \forall g \in \mathcal{G}$$

For $q = \infty$,

$$\alpha_g^* = \mathbf{u}_g - \Pi_{\|\cdot\|_1 \leq \lambda}[\mathbf{u}_g], \quad \forall g \in \mathcal{G}$$

These formula generalize soft-thresholding to groups of variables. They are used in **block-coordinate descent and proximal algorithms**.

Reweighted ℓ_2

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

$$\min_{\alpha, \eta_j \geq \epsilon} \frac{1}{2} \|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \frac{\lambda}{2} \sum_{j=1}^p \frac{\alpha[j]^2}{\eta_j} + \eta_j.$$

Summary so far

- Greedy methods directly address the NP-hard ℓ_0 -decomposition problem.
- Homotopy methods can be extremely efficient for small or medium-sized problems, or when the solution is very sparse.
- Coordinate descent provides in general quickly a solution with a small/medium precision, but gets slower when there is a lot of correlation in the dictionary.
- First order methods are very attractive in the large scale setting.
- Other good alternatives exists, active-set, reweighted ℓ_2 methods, stochastic variants, variants of OMP,...

Optimization for Dictionary Learning

$$\min_{\substack{\alpha \in \mathbb{R}^{p \times n} \\ \mathbf{D} \in \mathcal{C}}} \sum_{i=1}^n \frac{1}{2} \|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1$$

$$\mathcal{C} \triangleq \{\mathbf{D} \in \mathbb{R}^{m \times p} \text{ s.t. } \forall j = 1, \dots, p, \|\mathbf{d}_j\|_2 \leq 1\}.$$

- Classical optimization alternates between \mathbf{D} and α .
- Good results, but **very slow!**

Optimization for Dictionary Learning

[Mairal, Bach, Ponce, and Sapiro, 2009a]

Classical formulation of dictionary learning

$$\min_{\mathbf{D} \in \mathcal{C}} f_n(\mathbf{D}) = \min_{\mathbf{D} \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^n l(\mathbf{x}_i, \mathbf{D}),$$

where

$$l(\mathbf{x}, \mathbf{D}) \triangleq \min_{\boldsymbol{\alpha} \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1.$$

Which formulation are we interested in?

$$\min_{\mathbf{D} \in \mathcal{C}} \left\{ f(\mathbf{D}) = \mathbb{E}_{\mathbf{x}}[l(\mathbf{x}, \mathbf{D})] \approx \lim_{n \rightarrow +\infty} \frac{1}{n} \sum_{i=1}^n l(\mathbf{x}_i, \mathbf{D}) \right\}$$

[Bottou and Bousquet, 2008]: Online learning can

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

Optimization for Dictionary Learning

Require: $\mathbf{D}_0 \in \mathbb{R}^{m \times p}$ (initial dictionary); $\lambda \in \mathbb{R}$

1: $\mathbf{A}_0 = \mathbf{0}, \mathbf{B}_0 = \mathbf{0}$.

2: **for** $t=1, \dots, T$ **do**

3: Draw \mathbf{x}_t

4: Sparse Coding

$$\boldsymbol{\alpha}_t \leftarrow \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{x}_t - \mathbf{D}_{t-1} \boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1,$$

5: Aggregate sufficient statistics

$$\mathbf{A}_t \leftarrow \mathbf{A}_{t-1} + \boldsymbol{\alpha}_t \boldsymbol{\alpha}_t^T, \mathbf{B}_t \leftarrow \mathbf{B}_{t-1} + \mathbf{x}_t \boldsymbol{\alpha}_t^T$$

6: Dictionary Update (block-coordinate descent)

$$\mathbf{D}_t \leftarrow \arg \min_{\mathbf{D} \in \mathcal{C}} \frac{1}{t} \sum_{i=1}^t \left(\frac{1}{2} \|\mathbf{x}_i - \mathbf{D} \boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1 \right).$$

7: **end for**

Optimization for Dictionary Learning

Which guarantees do we have?

Under a few reasonable assumptions,

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

$$\lim_{t \rightarrow +\infty} \hat{f}_t(\mathbf{D}_t) - f(\mathbf{D}_t) = 0,$$

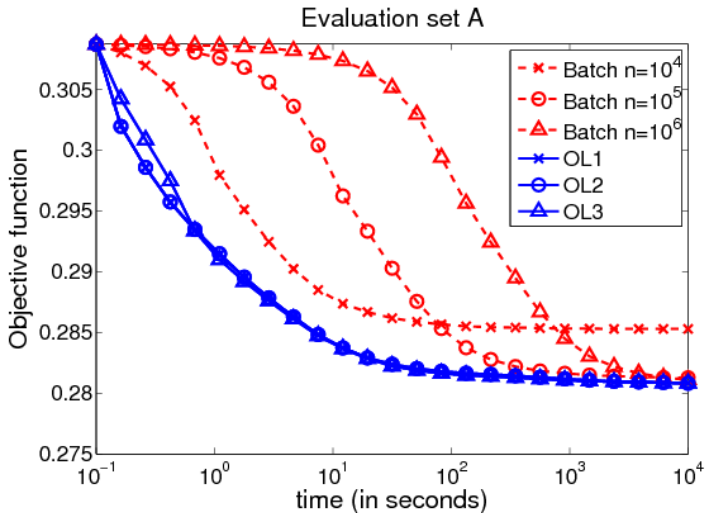
- \mathbf{D}_t is asymptotically close to a stationary point.

Extensions (all implemented in SPAMS)

- non-negative matrix decompositions.
- sparse PCA (sparse dictionaries).
- fused-lasso regularizations (piecewise constant dictionaries)

Optimization for Dictionary Learning

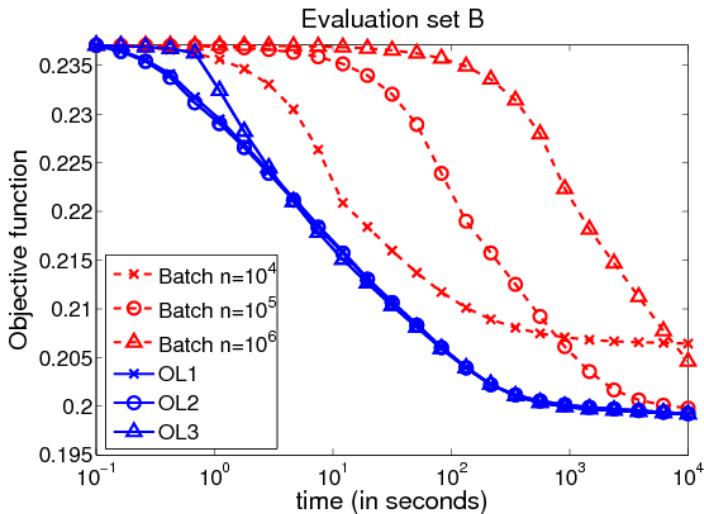
Experimental results, batch vs online



$$m = 8 \times 8, p = 256$$

Optimization for Dictionary Learning

Experimental results, batch vs online



$$m = 12 \times 12 \times 3, p = 512$$

Optimization for Dictionary Learning

Inpainting a 12-Mpixel photograph

THE SALINAS VALLEY is in Northern California. It is a long narrow swale between two ranges of mountains, and the Salinas River winds and twists up the center until it falls at last into Monterey Bay.

I remember my childhood games for grasses and secret flowers. I remember where a toad may live and what time the birds awaken in the summer and what trees and seasons smelled like-how people looked and walked and smelled even. The memory of odors is very rich.

I remember that the Gabilan Mountains to the east of the valley were light gay mountains full of sun and loveliness and a kind of invitation, so that you wanted to climb into their warm foothills almost as you want to climb into the lap of a beloved mother. They were beckoning mountains with a blown grass love. The Santa Lucia stood up 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 dread 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 Gabilans 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 in from its edges 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 swirl places under a high bank. The tules and grasses grew back, and willows straightened up with the flood debris in their upper branches. The Salinas was only a part-time river. The summer sun drove it underground. It was not a flat river at all, but it was the only one we had and so we boasted about it 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 bored a well. The drill came up first with topsoil and then with gravel and then with white sea sand full of shells and even pe...

Optimization for Dictionary Learning

Inpainting a 12-Mpixel photograph



Optimization for Dictionary Learning

Inpainting a 12-Mpixel photograph



Optimization for Dictionary Learning

Inpainting a 12-Mpixel photograph



Extension to NMF and sparse PCA

[Mairal, Bach, Ponce, and Sapiro, 2009b]

NMF extension

$$\min_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^{p \times n} \\ \mathbf{D} \in \mathcal{C}}} \sum_{i=1}^n \frac{1}{2} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 \quad \text{s.t.} \quad \boldsymbol{\alpha}_i \geq 0, \quad \mathbf{D} \geq 0.$$

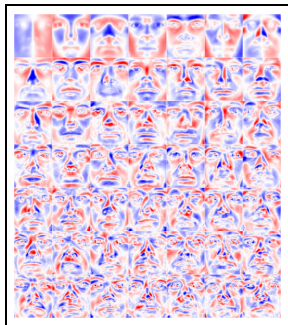
SPCA extension

$$\min_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^{p \times n} \\ \mathbf{D} \in \mathcal{C}'}} \sum_{i=1}^n \frac{1}{2} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_1\|_1$$

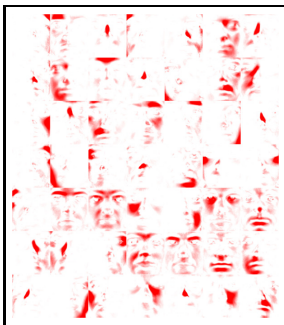
$$\mathcal{C}' \triangleq \{\mathbf{D} \in \mathbb{R}^{m \times p} \quad \text{s.t.} \quad \forall j \quad \|\mathbf{d}_j\|_2^2 + \gamma \|\mathbf{d}_j\|_1 \leq 1\}.$$

Extension to NMF and sparse PCA

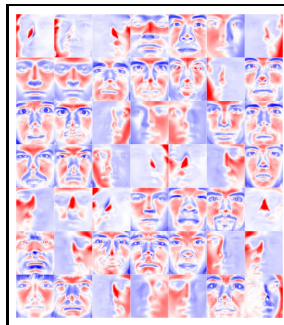
Faces: Extended Yale Database B



(a) PCA



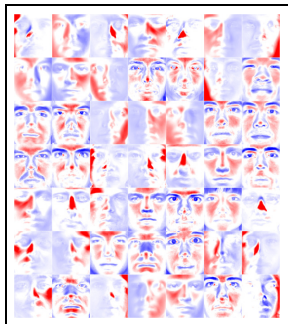
(b) NNMF



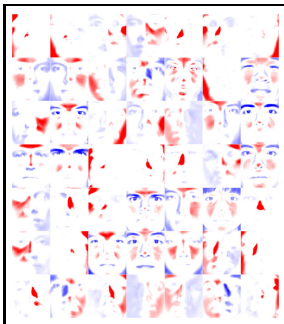
(c) DL

Extension to NMF and sparse PCA

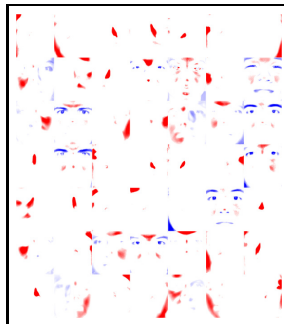
Faces: Extended Yale Database B



(d) SPCA, $\tau = 70\%$



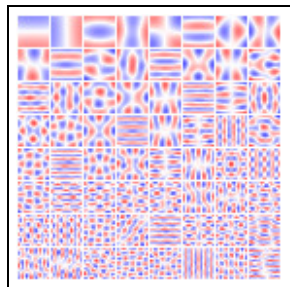
(e) SPCA, $\tau = 30\%$



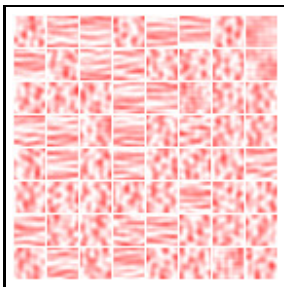
(f) SPCA, $\tau = 10\%$

Extension to NMF and sparse PCA

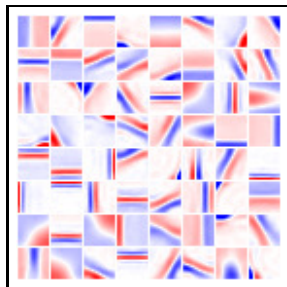
Natural Patches



(a) PCA



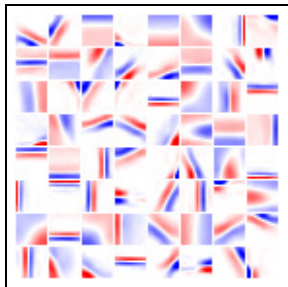
(b) NNMF



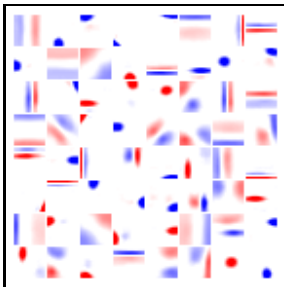
(c) DL

Extension to NMF and sparse PCA

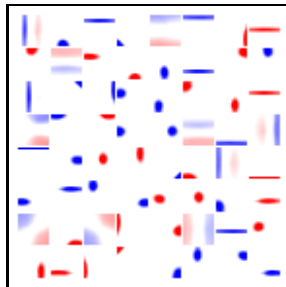
Natural Patches



(d) SPCA, $\tau = 70\%$



(e) SPCA, $\tau = 30\%$



(f) SPCA, $\tau = 10\%$

References I

- A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- D. P. Bertsekas. *Nonlinear programming*. Athena Scientific Belmont, Mass, 1999.
- J.F. Bonnans, J.C. Gilbert, C. Lemarechal, and C.A. Sagastizabal. *Numerical optimization: theoretical and practical aspects*. Springer-Verlag New York Inc, 2006.
- J. M. Borwein and A. S. Lewis. *Convex analysis and nonlinear optimization: Theory and examples*. Springer, 2006.
- L. Bottou and O. Bousquet. The trade-offs of large scale learning. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, pages 161–168. MIT Press, Cambridge, MA, 2008.
- S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math*, 57: 1413–1457, 2004.

References II

- I. Daubechies, R. DeVore, M. Fornasier, and S. Gunturk. Iteratively re-weighted least squares minimization for sparse recovery. *Commun. Pure Appl. Math*, 2009.
- B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of statistics*, 32(2):407–499, 2004.
- J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of statistics*, 1(2):302–332, 2007.
- W. J. Fu. Penalized regressions: The bridge versus the Lasso. *Journal of computational and graphical statistics*, 7:397–416, 1998.
- J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2009a.
- J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *ArXiv:0908.0050v1*, 2009b. submitted.
- S. Mallat and Z. Zhang. Matching pursuit in a time-frequency dictionary. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.
- H. M. Markowitz. The optimization of a quadratic function subject to linear constraints. *Naval Research Logistics Quarterly*, 3:111–133, 1956.

References III

- Y. Nesterov. Gradient methods for minimizing composite objective function. Technical report, CORE, 2007.
- Y. Nesterov. A method for solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Math. Dokl.*, 27:372–376, 1983.
- J. Nocedal and SJ Wright. *Numerical Optimization*. Springer: New York, 2006. 2nd Edition.
- M. R. Osborne, B. Presnell, and B. A. Turlach. On the Lasso and its dual. *Journal of Computational and Graphical Statistics*, 9(2):319–37, 2000.
- V. Roth and B. Fischer. The group-lasso for generalized linear models: uniqueness of solutions and efficient algorithms. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2008.
- S. Weisberg. *Applied Linear Regression*. Wiley, New York, 1980.