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

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

ψ induces sparsity in $\boldsymbol{\alpha}$. It can be

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

This is a selection problem.

Finding your way in the sparse coding literature...

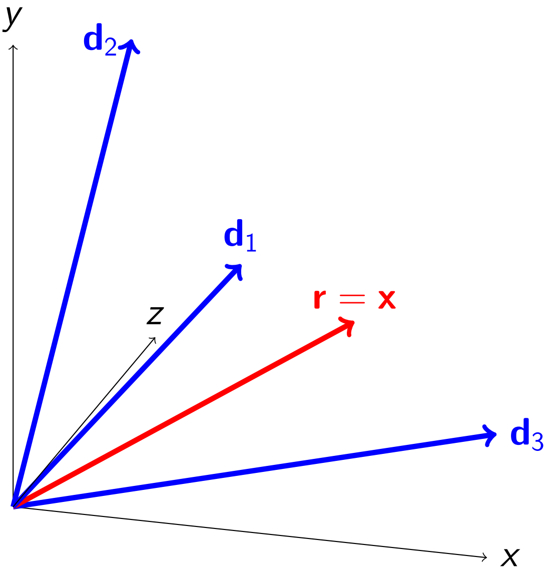
...is not easy. The literature is vast, redundant, sometimes confusing and many papers are claiming victory...

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].
- ...

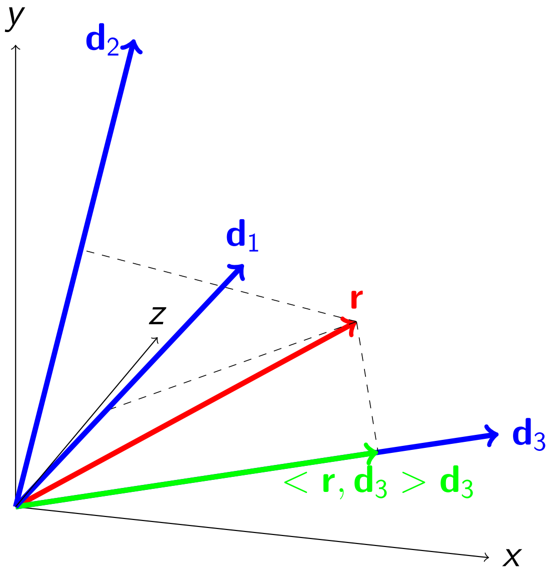
Matching Pursuit

$$\boldsymbol{\alpha} = (0, 0, 0)$$



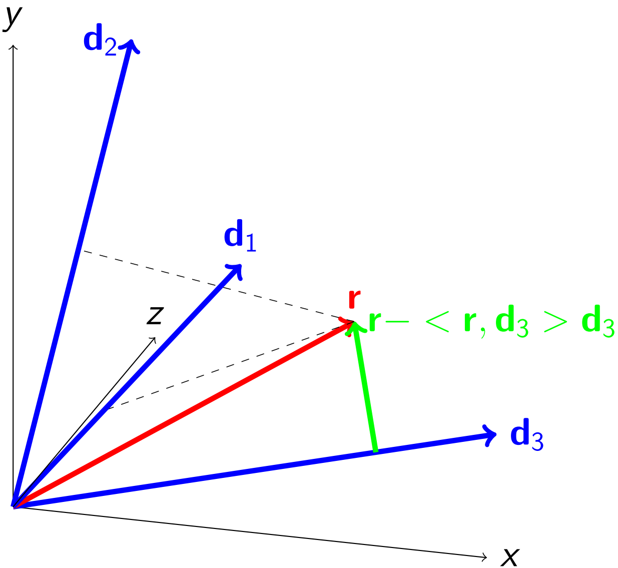
Matching Pursuit

$\boldsymbol{\alpha} = (0, 0, 0)$



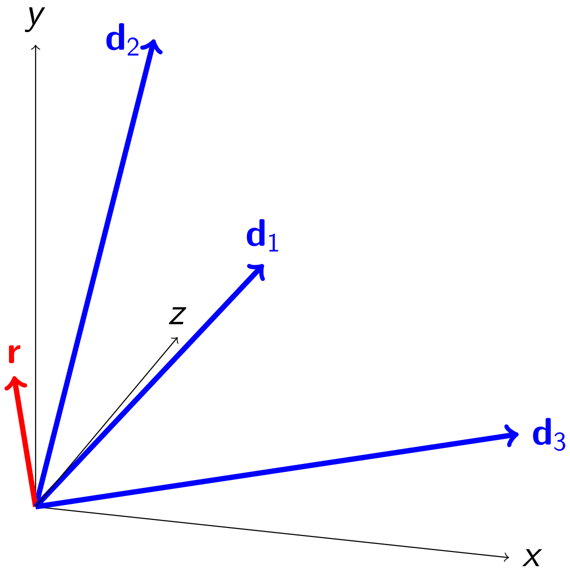
Matching Pursuit

$\boldsymbol{\alpha} = (0, 0, 0)$



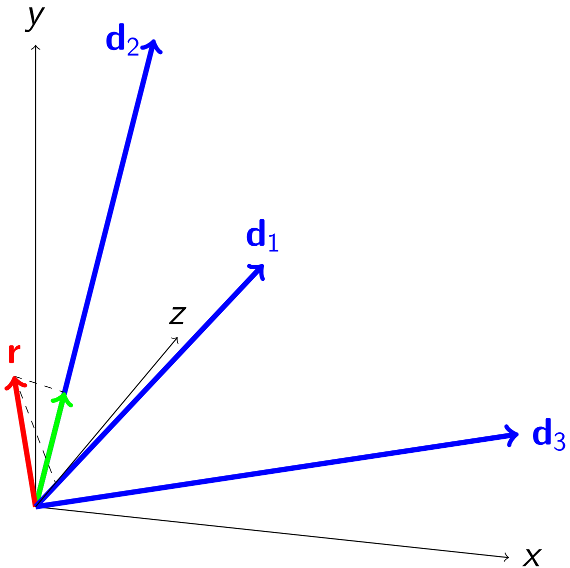
Matching Pursuit

$$\boldsymbol{\alpha} = (0, 0, 0.75)$$



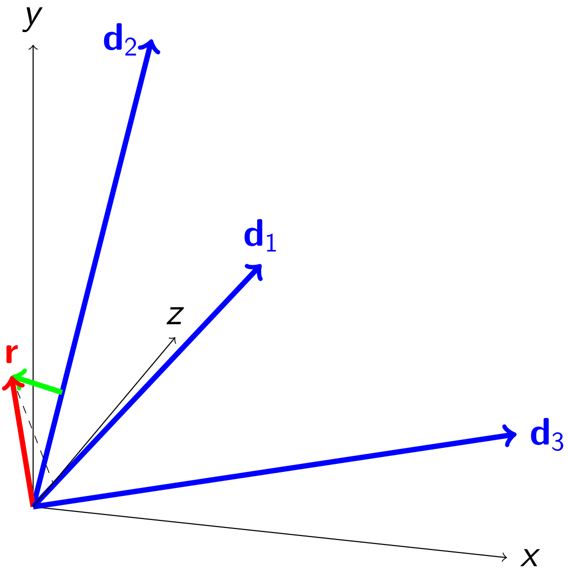
Matching Pursuit

$$\boldsymbol{\alpha} = (0, 0, 0.75)$$



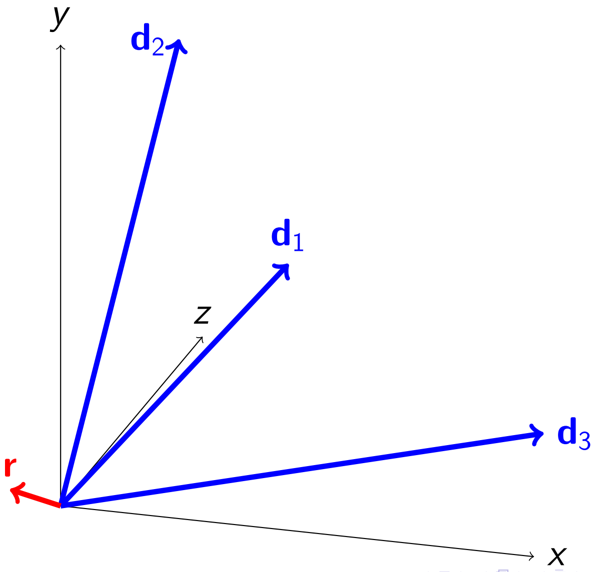
Matching Pursuit

$$\boldsymbol{\alpha} = (0, 0, 0.75)$$



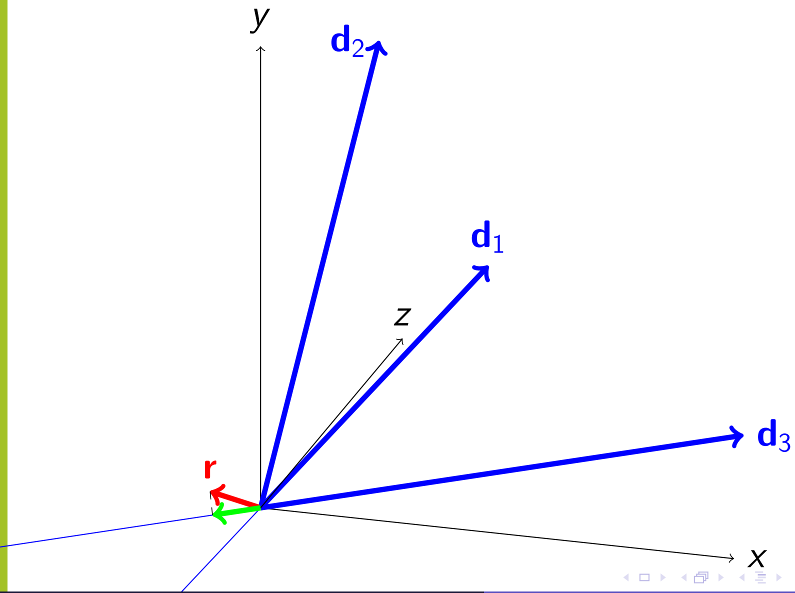
Matching Pursuit

$$\boldsymbol{\alpha} = (0, 0.24, 0.75)$$



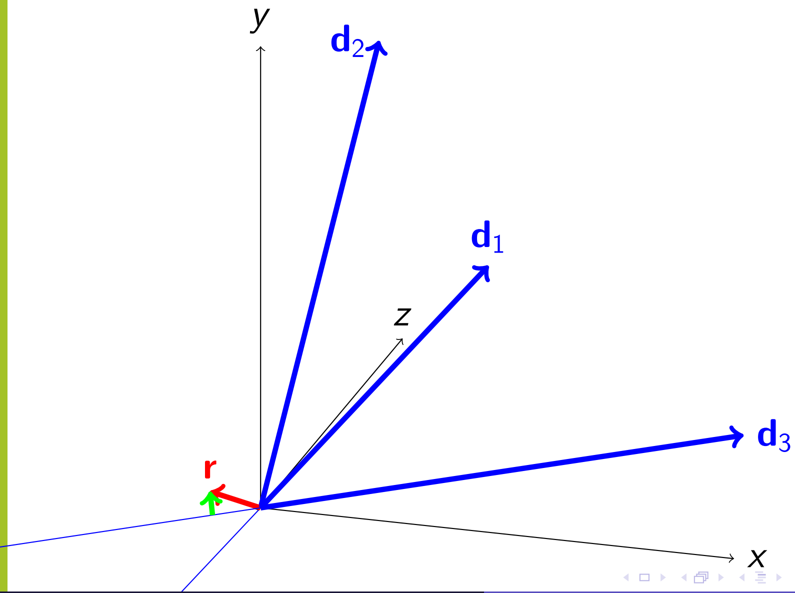
Matching Pursuit

$$\boldsymbol{\alpha} = (0, 0.24, 0.75)$$



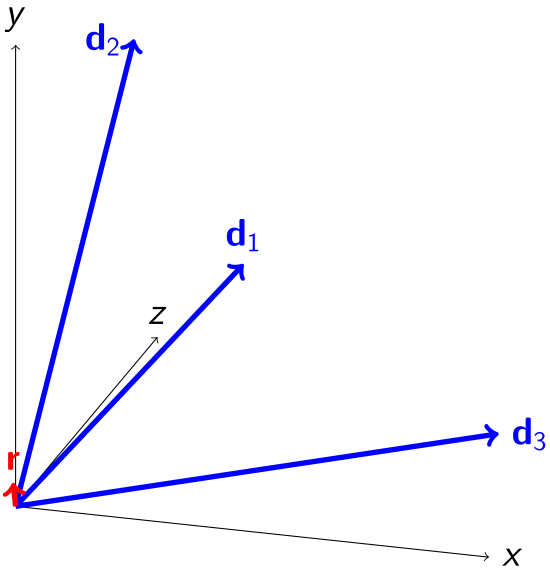
Matching Pursuit

$\boldsymbol{\alpha} = (0, 0.24, 0.75)$



Matching Pursuit

$$\boldsymbol{\alpha} = (0, 0.24, 0.65)$$



Matching Pursuit

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

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

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

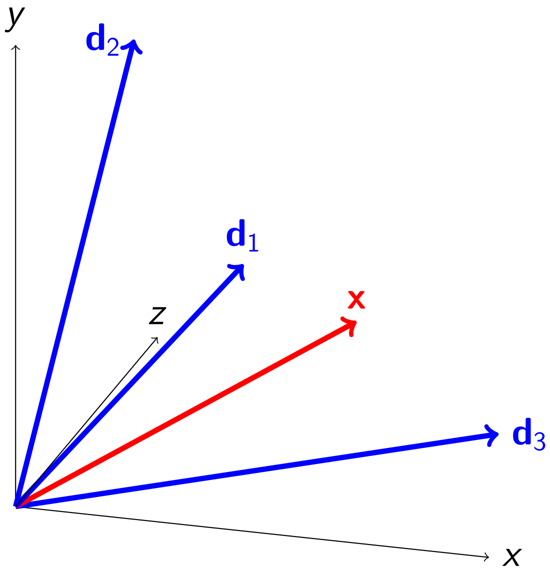
5: Update the residual and the coefficients

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

6: **end while**

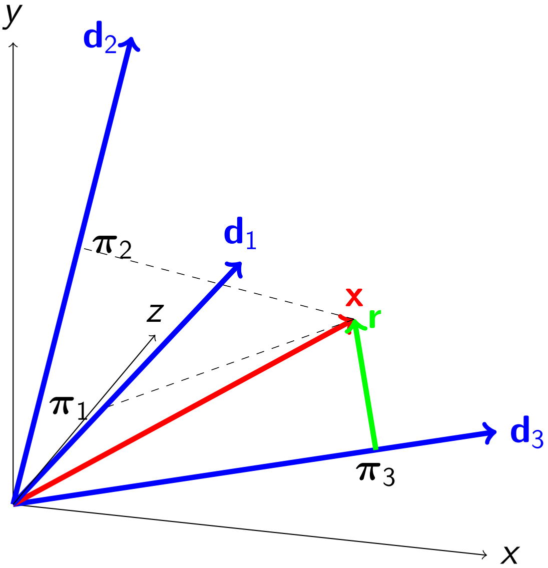
Orthogonal Matching Pursuit

$$\boldsymbol{\alpha} = (0, 0, 0)$$
$$\Gamma = \emptyset$$



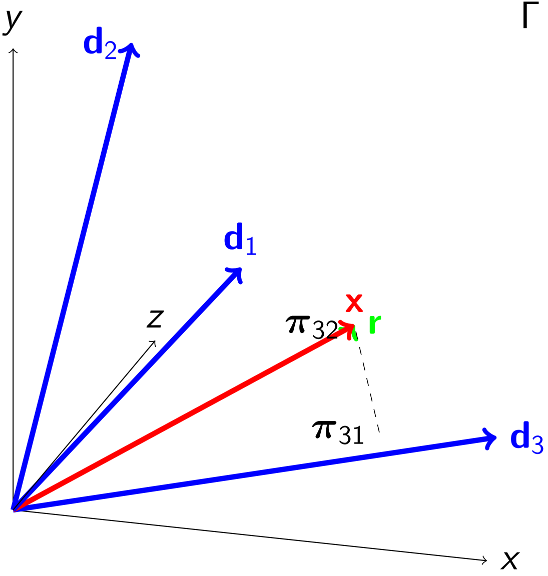
Orthogonal Matching Pursuit

$$\boldsymbol{\alpha} = (0, 0, 0.75)$$
$$\Gamma = \{3\}$$



Orthogonal Matching Pursuit

$$\boldsymbol{\alpha} = (0, 0.29, 0.63)$$
$$\Gamma = \{3, 2\}$$



Orthogonal Matching Pursuit

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

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

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

4: Update the active set: $\Gamma \leftarrow \Gamma \cup \{\hat{\imath}\}$.
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

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

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 $(\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma)^{-1}$ 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: **230000 signals processed per second!**

Optimality conditions of the Lasso

Nonsmooth optimization

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

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

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 $\boldsymbol{\alpha}$:

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

- 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 $\boldsymbol{\alpha}$, $\nabla f(\boldsymbol{\alpha}, \mathbf{u}) = \nabla f(\boldsymbol{\alpha})^T \mathbf{u}$.
- **Proposition 2:** $\boldsymbol{\alpha}$ is optimal iff for all $\mathbf{u}$ in $\mathbb{R}^p$, $\nabla f(\boldsymbol{\alpha}, \mathbf{u}) \geq 0$.

Optimality conditions of the Lasso

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

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

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

which is equivalent to the following conditions:

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

Homotopy

- A homotopy method provides a set of solutions indexed by a parameter.
- The regularization path $(\lambda, \boldsymbol{\alpha}^\star(\lambda))$ 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 is **piecewise linear**.

Homotopy, LARS

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

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

The regularization path is piecewise linear:

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

A simple interpretation of LARS

- Start from the trivial solution $(\lambda = ||\mathbf{D}^T\mathbf{x}||_\infty, \boldsymbol{\alpha}^\star(\lambda) = 0)$.
- Maintain the computations of $|\mathbf{d}_i^T(\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}^\star(\lambda))|$ 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

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

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

$$\boldsymbol{\alpha}[i] \leftarrow \arg\min_{\beta} \frac{1}{2}|| \underbrace{\mathbf{x} - \sum_{j\neq i} \boldsymbol{\alpha}[j]\mathbf{d}_j}_{\mathbf{r}} - \beta\mathbf{d}_i||_2^2 + \lambda|\beta|$$

$$\leftarrow \text{sign}(\mathbf{d}_i^T\mathbf{r})(|\mathbf{d}_i^T\mathbf{r}| - \lambda)^+$$

- $\Rightarrow$ **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

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

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

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

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

first-order/proximal methods

- They require solving efficiently the proximal operator

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

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

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

- 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:

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

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:

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

For $q = 2$,

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

For $q = \infty$,

$$\boldsymbol{\alpha}_g^\star = \mathbf{u}_g - \Pi_{\|.\|_1 \leq \lambda}[\mathbf{u}_g], \quad \forall g \in \mathcal{G}$$

These formula generalize soft-thrsholding 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

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

The formulation becomes

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

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

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

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

- Classical optimization alternates between $\mathbf{D}$ and $\boldsymbol{\alpha}$.
- 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?

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

[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); $\lambda \in \mathbb{R}$

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

for t=1,...,T **do**

Draw $\mathbf{x}_t$

Sparse Coding

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

Aggregate sufficient statistics

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

Dictionary Update (block-coordinate descent)

$$\mathbf{D}_t \leftarrow \operatorname*{arg\,min}_{\mathbf{D} \in \mathcal{C}} \frac{1}{t} \sum_{i=1}^{t} \Big(\frac{1}{2} ||\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda ||\boldsymbol{\alpha}_i||_1 \Big).$$

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

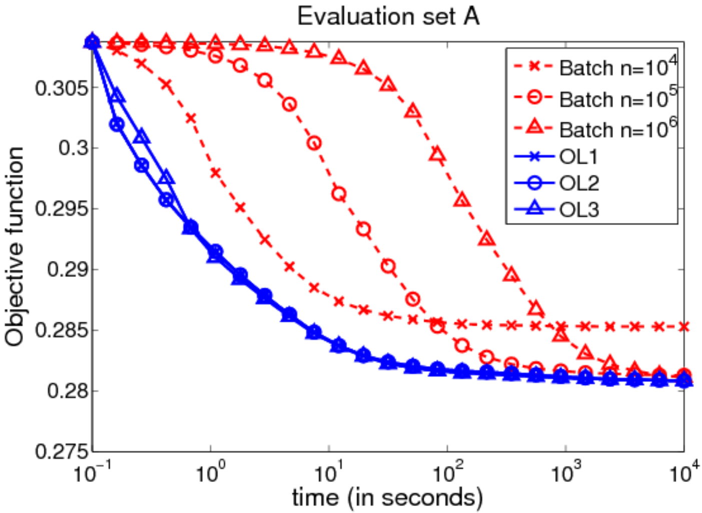
$$\lim_{t \to +\infty} \hat{f}_t(\mathbf{D}_t) - f(\mathbf{D}_t) = 0,$$

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

Extensions (all implemented in SPAMS)

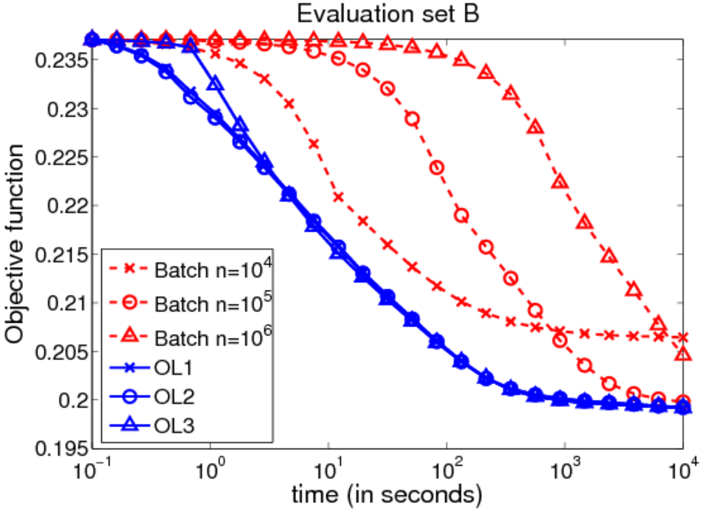
- non-negative matrix decompositions.
- sparse PCA (sparse dictionaries).
- fused-lasso regularizations (piecewise constant dictionaries)

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

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

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

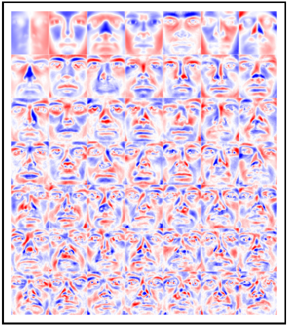
SPCA extension

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

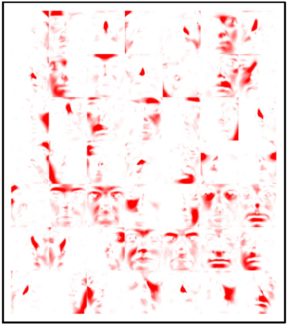
$$\mathcal{C}' \triangleq \{\mathbf{D} \in \mathbb{R}^{m \times p} \;\; \text{s.t.} \;\; \forall j \;\; ||\mathbf{d}_j||_2^2 + \gamma ||\mathbf{d}_j||_1 \leq 1\}.$$

Extension to NMF and sparse PCA

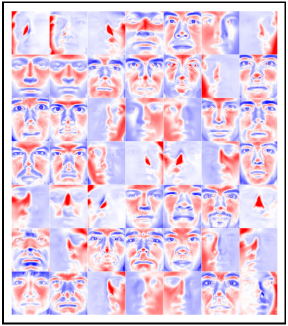
Faces: Extended Yale Database B



(a) PCA



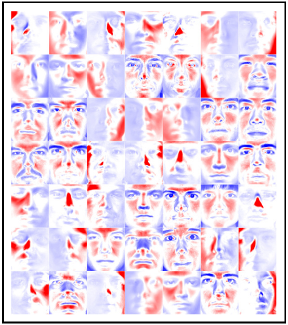
(b) NNMF



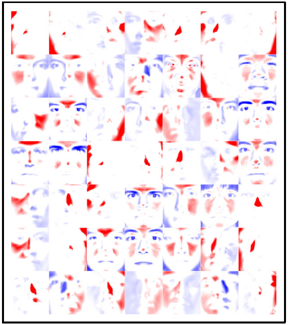
(c) DL

Extension to NMF and sparse PCA

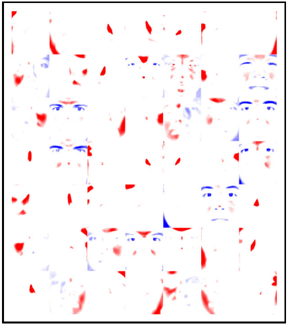
Faces: Extended Yale Database B



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



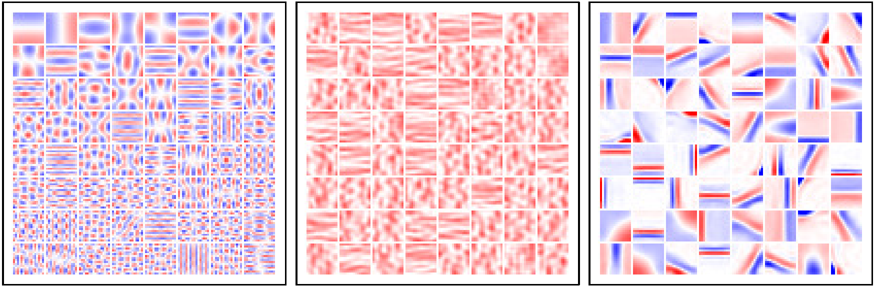
(e) SPCA, $\tau = 30\%$



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

Extension to NMF and sparse PCA

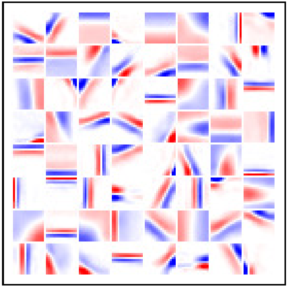
Natural Patches



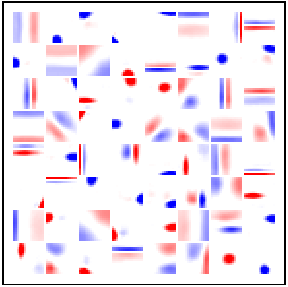
(a) PCA (b) NNMF (c) DL

Extension to NMF and sparse PCA

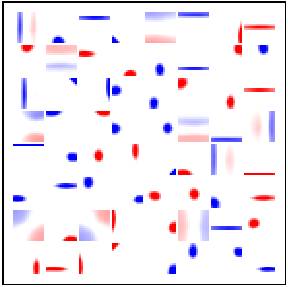
Natural Patches



(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. Hölfling, 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.