Incremental and Stochastic Majorization-Minimization Algorithms for Large-Scale Machine Learning

Julien Mairal

Inria, LEAR Team, Grenoble

SIAM Optimization 2014, San Diego

Statistical modeling with regularized risk minimization

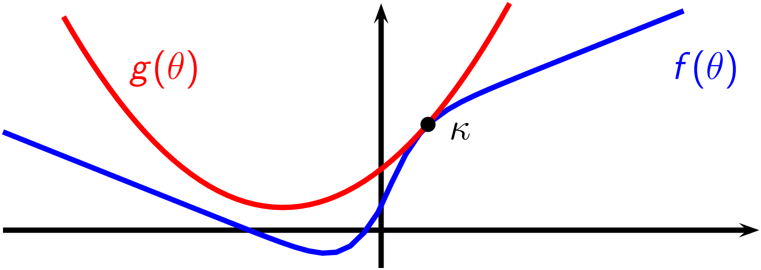
Given some data points $\mathbf{x}_i$, $i = 1, \ldots, n$, learn some model parameters θ in $\mathbb{R}^p$ by minimizing

$$\min_{\theta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{x}_i, \theta) + \lambda \psi(\theta),$$

where ℓ measures the data fit, and ψ is a regularizer.

The goal of this work is to deal with **large** n for relatively non-standard settings (non-convex,non-smooth,stochastic)

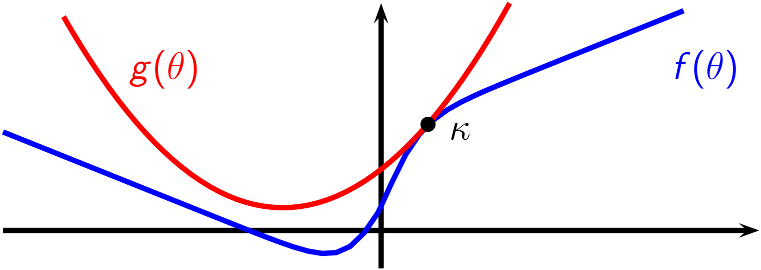
A simple (naive) optimization principle



Objective: $\min_{\theta \in \Theta} f(\theta)$

- Principle called Majorization-Minimization [Lange et al., 2000];
- quite popular in statistics and signal processing.

In this work



- **scalable** Majorization-Minimization algorithms;
- for **convex or non-convex** and **smooth or non-smooth** problems;

References

- J. Mairal. Optimization with First-Order Surrogate Functions. ICML'13;
- J. Mairal. Stochastic Majorization-Minimization Algorithms for Large-Scale Optimization. NIPS'13.

In this work

Methodology

- extend the MM principle to a large variety of settings;
- compute convergence rates for convex problems;
- show stationary point conditions for non-convex ones.

First direction: incremental optimization

- minimizes $(1/n)\sum_{i=1}^{n} f^i(\theta)$;
- requires some memory about past iterates;
- fast convergence rate for several passes over the data.

First direction: stochastic optimization

- no memory about past iterates;
- minimizes $\mathbb{E}_{\mathbf{x}}[f(\theta, \mathbf{x})]$.

Related work

incremental approaches for convex optimization

- stochastic average gradient [Schmidt, Roux, and Bach, 2013];
- stochastic dual coordinate ascent [Shalev-Schwartz and Zhang, 2012].

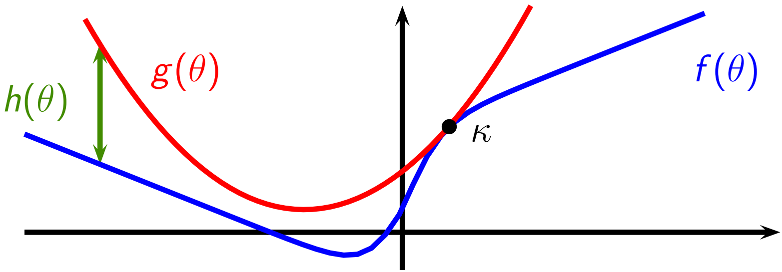
stochastic optimization

- stochastic proximal methods, e.g., [Duchi and Singer, 2009];
- literature about stochastic gradient descent, see, e.g., [Nemirovski et al., 2009];

non-convex optimization

- DC programming, see, e.g., [Gasso et al., 2009];
- online EM [Neal and Hinton, 1998, Cappé and Moulines, 2009].

Setting: First-Order Surrogate Functions



- $g(\theta') \geq f(\theta')$ for all θ' in $\arg\min_{\theta \in \Theta} g(\theta)$;
- the **approximation error** $h \triangleq g - f$ is differentiable, and ∇h is L-Lipschitz. Moreover, $h(\kappa) = 0$ and $\nabla h(\kappa) = 0$;
- we sometimes assume g to be strongly convex.

The Basic MM Algorithm

Algorithm 1 Basic Majorization-Minimization Scheme

Input: $\theta_0 \in \Theta$ (initial estimate); T (number of iterations).
for $t = 1, \ldots, T$ **do**
  Compute a surrogate g_t of f near θ_{t-1};
  Minimize g_t and update the solution:

$$\theta_t \in \operatorname*{arg\,min}_{\theta \in \Theta} g_t(\theta).$$

end for
Output: θ_T (final estimate);

Examples of First-Order Surrogate Functions

- **Lipschitz Gradient Surrogates**:
  f is L-smooth (differentiable + L-Lipschitz gradient).

$$g : \theta \mapsto f(\kappa) + \nabla f(\kappa)^\top (\theta - \kappa) + \frac{L}{2}\|\theta - \kappa\|_2^2.$$

Minimizing g yields a gradient descent step $\theta \leftarrow \kappa - \frac{1}{L}\nabla f(\kappa)$.

Examples of First-Order Surrogate Functions

- **Lipschitz Gradient Surrogates**:
  f is L-smooth (differentiable + L-Lipschitz gradient).
  $$g : \theta \mapsto f(\kappa) + \nabla f(\kappa)^\top (\theta - \kappa) + \frac{L}{2}\|\theta - \kappa\|_2^2.$$
  Minimizing g yields a gradient descent step $\theta \leftarrow \kappa - \frac{1}{L}\nabla f(\kappa)$.
- **Proximal Gradient Surrogates**:
  $f = f_1 + f_2$ with f_1 smooth.
  $$g : \theta \mapsto f_1(\kappa) + \nabla f_1(\kappa)^\top (\theta - \kappa) + \frac{L}{2}\|\theta - \kappa\|_2^2 + f_2(\theta).$$
  Minimizing g amounts to one step of the forward-backward, ISTA, or proximal gradient descent algorithm.
  [Beck and Teboulle, 2009, Combettes and Pesquet, 2010, Wright et al., 2008, Nesterov, 2007].

Examples of First-Order Surrogate Functions

- **Linearizing Concave Functions and DC-Programming**:
  $f = f_1 + f_2$ with f_2 smooth and concave.

$$g : \theta \mapsto f_1(\theta) + f_2(\kappa) + \nabla f_2(\kappa)^\top (\theta - \kappa).$$

When f_1 is convex, the algorithm is called DC-programming.

Examples of First-Order Surrogate Functions

- **Linearizing Concave Functions and DC-Programming**:
  $f = f_1 + f_2$ with f_2 smooth and concave.

  $$g : \theta \mapsto f_1(\theta) + f_2(\kappa) + \nabla f_2(\kappa)^\top (\theta - \kappa).$$

  When f_1 is convex, the algorithm is called DC-programming.

- **Quadratic Surrogates**:
  f is twice differentiable, and $\mathbf{H}$ is a uniform upper bound of $\nabla^2 f$:

  $$g : \theta \mapsto f(\kappa) + \nabla f(\kappa)^\top (\theta - \kappa) + \frac{1}{2}(\theta - \kappa)^\top \mathbf{H}(\theta - \kappa).$$

  Actually a big deal in statistics and machine learning [Böhning and Lindsay, 1988, Khan et al., 2010, Jebara and Choromanska, 2012].

- . . .

Theoretical Guarantees for Non-convex Problems

When using first-order surrogates,

- for **convex** problems: $f(\theta_t) - f^\star = O(1/t)$.
- for μ-**strongly convex** ones: $O((1-\mu/L)^t)$.
- for **non-convex** problems: $f(\theta_t)$ monotonically decreases and

$$\liminf_{t \to +\infty} \inf_{\theta \in \Theta} \frac{\nabla f(\theta_t, \theta - \theta_t)}{\|\theta - \theta_t\|_2} \geq 0, \qquad (1)$$

  which we call asymptotic stationary point condition.

Directional derivative

$$\nabla f(\theta, \kappa) = \lim_{\varepsilon \to 0^+} \frac{f(\theta + \varepsilon\kappa) - f(\theta)}{\varepsilon}.$$

- when in addition $\Theta = \mathbb{R}^p$, (1) is equivalent to $\nabla f(\theta_t) \to 0$.

Incremental Optimization: MISO

Suppose that f splits into many components:

$$f(\theta) = \frac{1}{n}\sum_{i=1}^{n} f^i(\theta).$$

Recipe

- Incrementally update an approximate surrogate $\frac{1}{n}\sum_{i=1}^{n} g^i$;
- add some heuristics for practical implementations.

Related work for convex problems

- related to SAG [Schmidt et al., 2013] and SDCA [Shalev-Schwartz and Zhang, 2012], but offers different update rules.

Incremental Optimization: MISO

Algorithm 2 Incremental Scheme MISO

1: **Input:** $\theta_0 \in \Theta$; T (number of iterations).
2: Choose surrogates g_0^i of f^i near θ_0 for all i;
3: **for** $t = 1, \ldots, T$ **do**
4: Randomly pick up one index $\hat{\imath}_t$ and choose a surrogate $g_t^{\hat{\imath}_t}$ of $f^{\hat{\imath}_t}$ near θ_{t-1}. Set $g_t^i \triangleq g_{t-1}^i$ for $i \neq \hat{\imath}_t$;
5: Update the solution:

$$\theta_t \in \operatorname*{arg\,min}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} g_t^i(\theta).$$

6: **end for**
7: **Output:** θ_T (final estimate);

Incremental Optimization: MISO

Update rule with Lipschitz gradient surrogates

We want to minimize $\frac{1}{n}\sum_{i=1}^{n} f^i(\theta)$.

$$\begin{aligned}
\theta_t &= \arg\min_{\theta \in \Theta} \frac{1}{n}\sum_{i=1}^{n} f^i(\kappa^i) + \nabla f^i(\kappa^i)^\top(\theta - \kappa^i) + \frac{L}{2}\|\theta - \kappa^i\|_2^2 \\
&= \frac{1}{n}\sum_{i=1}^{n} \kappa^i - \frac{1}{Ln}\sum_{i=1}^{n} \nabla f^i(\kappa^i).
\end{aligned}$$

At iteration n, randomly draw one index $\hat{\imath}_t$, and update $\kappa^{\hat{\imath}_t} \leftarrow \theta_t$.

Remarks

- replace $(1/n)\sum_{i=1}^{n} \kappa^i$ by θ_{t-1} yields SAG [Schmidt et al., 2013].
- replace $(1/L)$ by $(1/\mu)$ is almost identical to SDCA [Shalev-Schwartz and Zhang, 2012].

Incremental Optimization: MISO

Update rule for proximal gradient surrogates

We want to minimize $\frac{1}{n}\sum_{i=1}^n f^i(\theta) + \psi(\theta)$.

$$\begin{aligned}\theta_t &= \operatorname*{arg\,min}_{\theta \in \Theta} \frac{1}{n}\sum_{i=1}^n f^i(\kappa_t^i) + \nabla f^i(\kappa_t^i)^\top(\theta - \kappa_t^i) + \frac{L}{2}\|\theta - \kappa_t^i\|_2^2 + \psi(\theta) \\ &= \operatorname*{arg\,min}_{\theta \in \Theta} \frac{1}{2}\left\|\theta - \left(\frac{1}{n}\sum_{i=1}^n \kappa_t^i - \frac{1}{Ln}\sum_{i=1}^n \nabla f^i(\kappa_t^i)\right)\right\|_2^2 + \frac{1}{L}\psi(\theta).\end{aligned}$$

Incremental Optimization: MISO

Theoretical Guarantees

- for **non-convex** problems, the guarantees are the same as the generic MM algorithm with probability one.
- for **convex** problems and proximal gradient surrogates, the expected convergence rate with averaging becomes $O(n/t)$.
- for μ-**strongly convex** problems and proximal gradient surrogates, the expected convergence rate is linear $O((1-\mu/(nL))^t)$.

Remarks for μ-strongly convex problems

- the rates of SDCA and SAG in this setting are better: $\mu/(Ln)$ is replaced by $O(\min(\mu/L, 1/n))$;
- the MM principle is too conservative. For smooth problems, we can match these rates by using "minorizing" surrogates [Mairal, 2014].

Incremental Optimization: MISO

Example for ℓ_2-logistic regression:

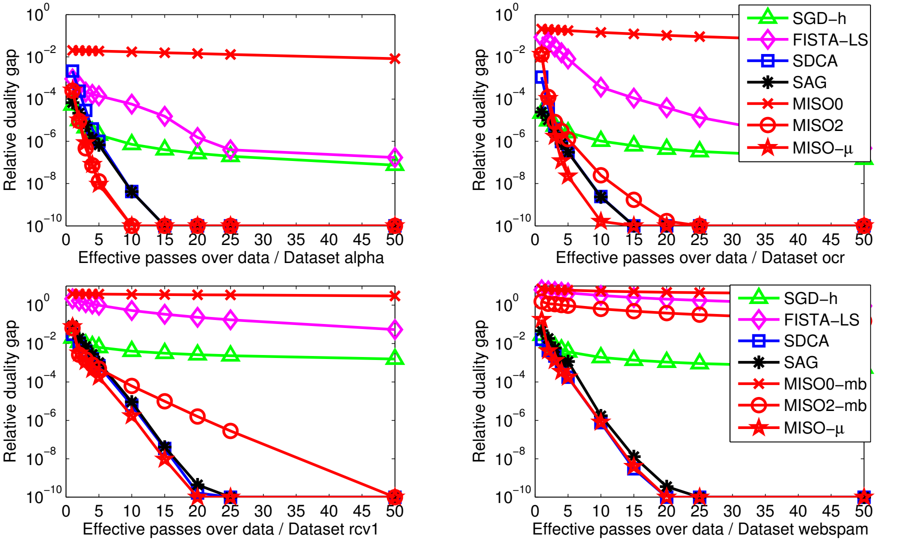
$$\min_{\theta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \log(1 + e^{-y_i \theta^\top \mathbf{x}_i}) + \frac{\lambda}{2} \|\theta\|_2^2.$$

The problem is λ-strongly convex.

Table : Description of datasets used in our experiments.

name	n	p	storage	density	size (GB)
alpha	500 000	500	dense	1	1.86
ocr	2 500 000	1 155	dense	1	21.5
rcv1	781 265	47 152	sparse	0.0016	0.89
webspam	250 000	16 091 143	sparse	0.0002	13.90

Incremental Optimization: MISO



Incremental DC programming

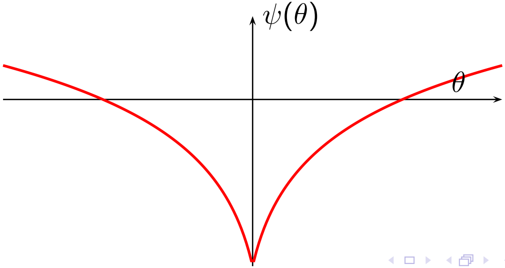
Consider a binary classification problem with n training samples $(y_i, \mathbf{x}_i)$, with y_i in $\{-1, +1\}$ and $\mathbf{x}_i$ in $\mathbb{R}^p$. Assume that there exists a sparse linear model $y \approx \text{sign}(\theta^\top \mathbf{x})$, learned by minimizing

$$\min_{\theta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \log(1 + e^{-y_i \theta^\top \mathbf{x}_i}) + \lambda \psi(\theta).$$

Traditional choices for ψ: $\psi(\theta) = \|\theta\|_2^2$ or $\|\theta\|_1$.

Non-convex sparsity inducing penalty:

- $\psi(\theta) = \sum_{j=1}^{p} \log(|\theta[j]| + \varepsilon)$.



Incremental DC programming

- upper-bound $f_i : \theta \mapsto \log(1 + e^{-y_i \theta^\top \mathbf{x}_i})$ by

$$\theta \mapsto f_i(\kappa^i) + \nabla f_i(\theta_{t-1})^\top (\theta - \theta_{t-1}) + \frac{L}{2}\|\theta - \theta_{t-1}\|_2^2;$$

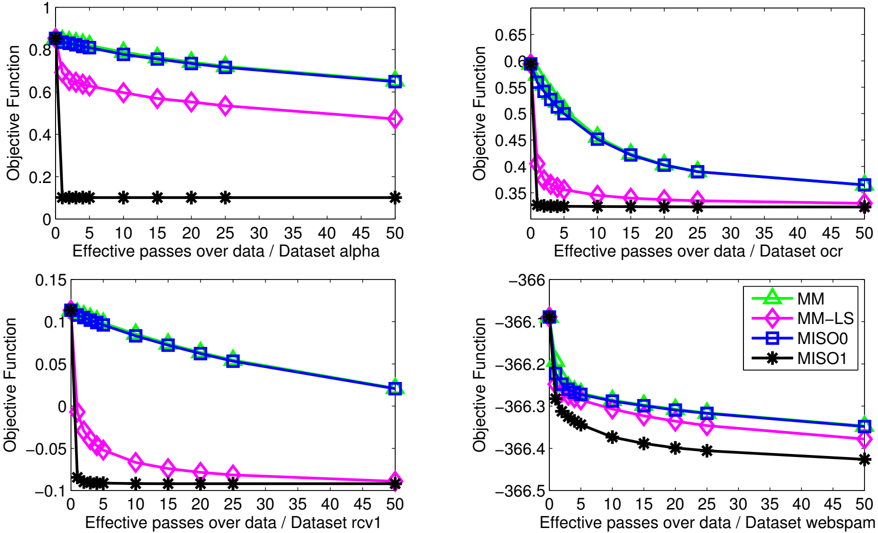
- upper-bound $\lambda \sum_{j=1}^p \log(|\theta[j]| + \varepsilon)$ by

$$\theta \mapsto \lambda \sum_{j=1}^p \frac{|\theta[j]|}{|\theta_{t-1}[j]| + \varepsilon}.$$

this is an incremental reweighted-ℓ_1 algorithm [Candès et al., 2008].

The overall surrogate can be minimized in closed-form by using **soft-thresholding**.

Incremental Optimization: MISO



Stochastic Majorization Minimization: SMM

Suppose that f is an expectation:

$$f(\theta) = \mathbb{E}_{\mathbf{x}}[\ell(\theta, \mathbf{x})].$$

Recipe

- Draw a function $f_t : \theta \mapsto \ell(\theta, \mathbf{x}_t)$ at iteration t;
- Iteratively update an approximate surrogate $\bar{g}_t = (1 - w_t)\bar{g}_{t-1} + w_t g_t$;
- Choose appropriate w_t.

Related Work

- online-EM [Neal and Hinton, 1998, Cappé and Moulines, 2009];
- online dictionary learning [Mairal et al., 2010a].

Stochastic Majorization Minimization: SMM

Algorithm 3 Stochastic Majorization-Minimization Scheme

Input: $\theta_0 \in \Theta$ (initial estimate); T (number of iterations); $(w_t)_{t \geq 1}$, weights in $(0, 1]$;
initialize the approximate surrogate: $\bar{g}_0 : \theta \mapsto \frac{\rho}{2}\|\theta - \theta_0\|_2^2$;
for $t = 1, \ldots, T$ **do**
  draw a training point $\mathbf{x}_t$;
  choose a surrogate function g_t of $f_t : \theta \mapsto \ell(\mathbf{x}_t, \theta)$ near θ_{t-1};
  update the approximate surrogate: $\bar{g}_t = (1 - w_t)\bar{g}_{t-1} + w_t g_t$;
  update the current estimate:

$$\theta_t \in \operatorname*{arg\,min}_{\theta \in \Theta} \bar{g}_t(\theta);$$

end for
Output: θ_T (current estimate);

Stochastic Majorization Minimization: SMM

Update Rule for Proximal Gradient Surrogate

$$\theta_t \leftarrow \arg\min_{\theta \in \Theta} \sum_{i=1}^{t} w_t^i \left[\nabla f_i(\theta_{i-1})^\top \theta + \frac{L}{2} \|\theta - \theta_{i-1}\|_2^2 + \psi(\theta) \right]. \quad \text{(SMM)}$$

Other schemes in the literature [Duchi and Singer, 2009]:

$$\theta_t \leftarrow \arg\min_{\theta \in \Theta} \nabla f_t(\theta_{t-1})^\top \theta + \frac{1}{2\eta_t} \|\theta - \theta_{t-1}\|_2^2 + \psi(\theta), \quad \text{(FOBOS)}$$

or regularized dual averaging (RDA) of Xiao [2010]:

$$\theta_t \leftarrow \arg\min_{\theta \in \Theta} \frac{1}{t} \sum_{i=1}^{t} \nabla f_i(\theta_{i-1})^\top \theta + \frac{1}{2\eta_t} \|\theta\|_2^2 + \psi(\theta). \quad \text{(RDA)}$$

or others...

Stochastic Majorization Minimization: SMM

Theoretical Guarantees - Non-Convex Problems

under a set of reasonable assumptions,

- $f(\theta_t)$ almost surely converges;
- the function $\bar{g}_t$ asymptotically behaves as a first-order surrogate;
- we almost surely have asymptotic stationary point conditions.

Theoretical Guarantees - Convex Problems

for proximal gradient surrogates, we obtain similar expected rates as SGD with averaging [see Nemirovski et al., 2009]: $O(1/t)$ for strongly convex problems, $O(\log(t)/\sqrt{t})$ for convex ones.
(under bounded subgradients assumptions and specific w_t).

Experimental Conclusions for ℓ_2-logistic Regression

- Incremental and stochastic schemes were significantly faster than batch ones;
- MISO with heuristics was competitive with the state of the art (SAG, SGD, Liblinear);
- after one pass over the data, SMM quickly achieves a **low-precision** solution. For higher precision, MISO is prefered.
- **problems tested were large but relatively well conditioned**.

Online Sparse Matrix Factorization

Consider some signals $\mathbf{x}$ in $\mathbb{R}^m$. We want to find a dictionary $\mathbf{D}$ in $\mathbb{R}^{m \times K}$. The quality of $\mathbf{D}$ is measured through the loss

$$\ell(\mathbf{x}, \mathbf{D}) \triangleq \min_{\boldsymbol{\alpha} \in \mathbb{R}^K} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda_1 \|\boldsymbol{\alpha}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\alpha}\|_2^2.$$

Then, learning the dictionary amounts to solving

$$\min_{\mathbf{D} \in \mathcal{C}} \mathbb{E}_{\mathbf{x}}\left[\ell(\mathbf{x}, \mathbf{D})\right] + \varphi(\mathbf{D}),$$

Why is it a matrix factorization problem?

$$\min_{\mathbf{D} \in \mathcal{C}, \mathbf{A} \in \mathbb{R}^{K \times n}} \frac{1}{n}\left[\frac{1}{2}\|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \sum_{i=1}^{n} \lambda_1 \|\boldsymbol{\alpha}_i\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\alpha}_i\|_2^2\right] + \varphi(\mathbf{D}).$$

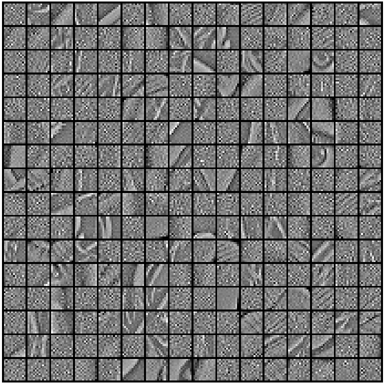
Online Structured Matrix Factorization

- when $\mathcal{C} = \{\mathbf{D} \in \mathbb{R}^{m \times K} \text{ s.t. } \|\mathbf{d}_j\|_2 \leq 1\}$ and $\varphi = 0$, the problem is called **sparse coding** or **dictionary learning** [Olshausen and Field, 1997, Elad and Aharon, 2006, Mairal et al., 2010a].
- non-negativity constraints can be easily added. It yields an online **nonnegative matrix factorization** algorithm.
- φ can be a function encouraging a particular structure in $\mathbf{D}$ [Jenatton et al., 2009].

Online Structured Matrix Factorization

Dictionary Learning on Natural Image Patches

Consider $n = 250\,000$ whitened natural image patches of size $m = 12 \times 12$. We learn a dictionary with $K = 256$ elements.

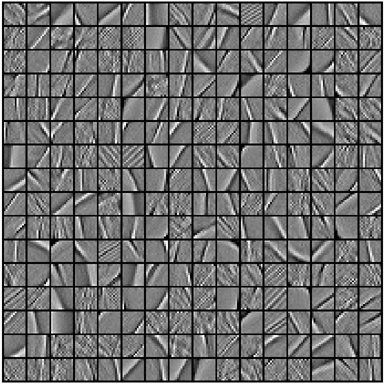


0s on an old laptop 1.2GHz dual-core CPU. (initialization)

Online Structured Matrix Factorization

Dictionary Learning on Natural Image Patches

Consider $n = 250\,000$ whitened natural image patches of size $m = 12 \times 12$. We learn a dictionary with $K = 256$ elements.

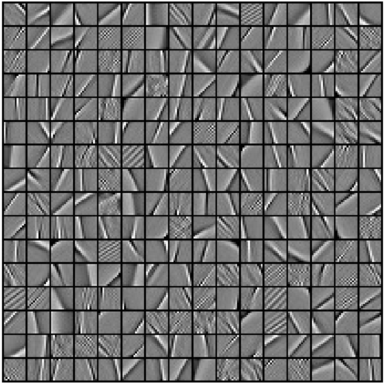


1.15s on an old laptop 1.2GHz dual-core CPU (0.1 pass)

Online Structured Matrix Factorization

Dictionary Learning on Natural Image Patches

Consider $n = 250\,000$ whitened natural image patches of size $m = 12 \times 12$. We learn a dictionary with $K = 256$ elements.

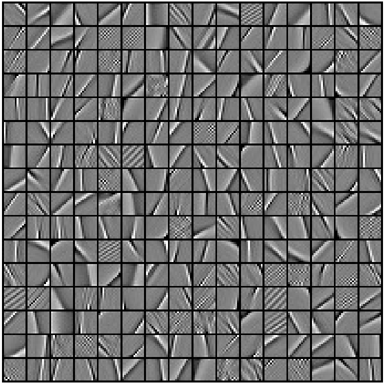


5.97s on an old laptop 1.2GHz dual-core CPU (0.5 pass)

Online Structured Matrix Factorization

Dictionary Learning on Natural Image Patches

Consider $n = 250\,000$ whitened natural image patches of size $m = 12 \times 12$. We learn a dictionary with $K = 256$ elements.

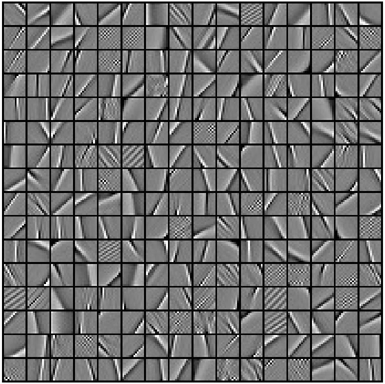


12.44s on an old laptop 1.2GHz dual-core CPU (1 pass)

Online Structured Matrix Factorization

Dictionary Learning on Natural Image Patches

Consider $n = 250\,000$ whitened natural image patches of size $m = 12 \times 12$. We learn a dictionary with $K = 256$ elements.

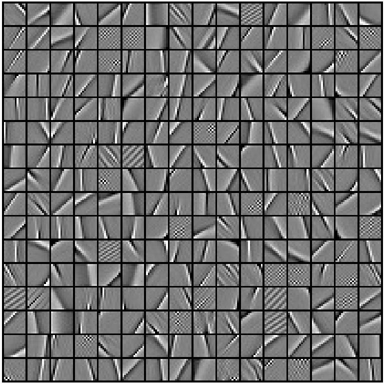


23.22s on an old laptop 1.2GHz dual-core CPU (2 passes)

Online Structured Matrix Factorization

Dictionary Learning on Natural Image Patches

Consider $n = 250\,000$ whitened natural image patches of size $m = 12 \times 12$. We learn a dictionary with $K = 256$ elements.



60.60s on an old laptop 1.2GHz dual-core CPU (5 passes)

Conclusion

What we have done

- we have given a unified view of a large number of algorithms;
- ... and introduced new ones for large-scale optimization.

A take-home message

- our algorithms are likely to be useful for large-scale **non-convex** and possibly **non-smooth** problems, which is a relatively non-standard, but useful, setting.

Source Code

- code is now available in the toolbox SPAMS (C++ interfaced with Matlab, Python, R). `http://spams-devel.gforge.inria.fr/`;

Examples of First-Order Surrogate Functions

- **More Exotic Surrogates**:
  Consider a smooth approximation of the trace (nuclear) norm see François Caron's talk)

$$f_\mu : \theta \mapsto \text{Tr}\left((\theta^\top \theta + \mu \mathbf{I})^{1/2} \right) = \sum_{i=1}^{p} \sqrt{\lambda_i(\theta^\top \theta) + \mu},$$

  $f' : \mathbf{H} \mapsto \text{Tr}\left(\mathbf{H}^{1/2}\right)$ is concave on the set of p.d. matrices and $\nabla f'(\mathbf{H}) = (1/2)\mathbf{H}^{-1/2}$.

$$g_\mu : \theta \mapsto f_\mu(\kappa) + \frac{1}{2} \text{Tr}\left((\kappa^\top \kappa + \mu \mathbf{I})^{-1/2} (\theta^\top \theta - \kappa^\top \kappa) \right),$$

  which yields the algorithm of Mohan and Fazel [2012].
  a
- and also **variational, saddle-point, Jensen surrogates**...

Examples of First-Order Surrogate Functions

- **Variational Surrogates**: $f(\theta_1) \triangleq \min_{\theta_2 \in \Theta_2} \tilde{f}(\theta_1, \theta_2)$, where $\tilde{f}$ is "smooth" w.r.t θ_1 and strongly convex w.r.t θ_2:

$$g : \theta_1 \mapsto \tilde{f}(\theta_1, \kappa_2^\star) \text{ with } \kappa_2^\star \triangleq \operatorname*{arg\,min}_{\theta_2 \in \Theta_2} \tilde{f}(\kappa_1, \theta_2).$$

- **Saddle-Point Surrogates**: $f(\theta_1) \triangleq \max_{\theta_2 \in \Theta_2} \tilde{f}(\theta_1, \theta_2)$, where $\tilde{f}$ is "smooth" w.r.t θ_1 and strongly concave w.r.t θ_2:

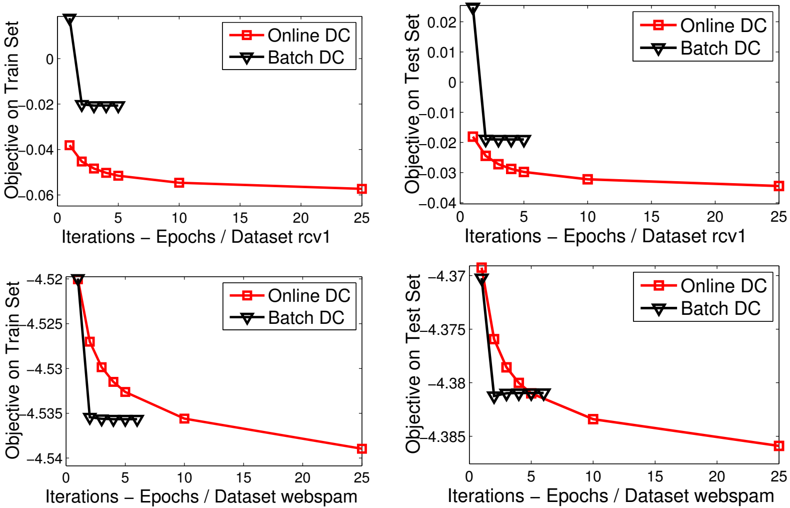
$$g : \theta_1 \mapsto \tilde{f}(\theta_1, \kappa_2^\star) + \frac{L''}{2}\|\theta_1 - \kappa_1\|_2^2.$$

- **Jensen Surrogates**: $f(\theta) \triangleq \tilde{f}(\mathbf{x}^\top \theta)$, where $\tilde{f}$ is L-smooth. Choose a weight vector $\mathbf{w}$ in $\mathbb{R}_+^p$ such that $\|\mathbf{w}\|_1 = 1$ and $\mathbf{w}_i \neq 0$ whenever $\mathbf{x}_i \neq 0$.

$$g : \theta \mapsto \sum_{i=1}^{p} \mathbf{w}_i f\left(\frac{\mathbf{x}_i}{\mathbf{w}_i}(\theta_i - \kappa_i) + \mathbf{x}^\top \kappa\right),$$

Stochastic DC programming

For logistic-regression with non-convex sparsity-inducing penalty.



Other variants of MM

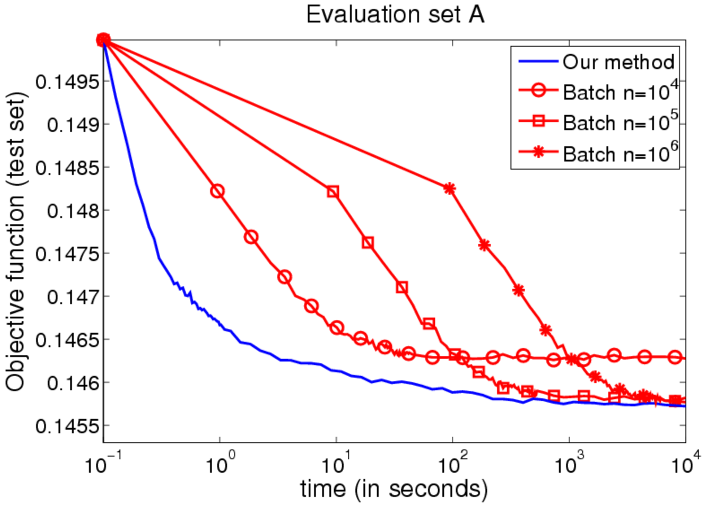
We also study in [Mairal, 2013a] a block coordinate scheme for **non-convex and convex** optimization.

Also several variants for **convex optimization**:

- an accelerated one (Nesterov's like);
- a "Frank-Wolfe" majorization-minimization algorithm.

Online Dictionary Learning

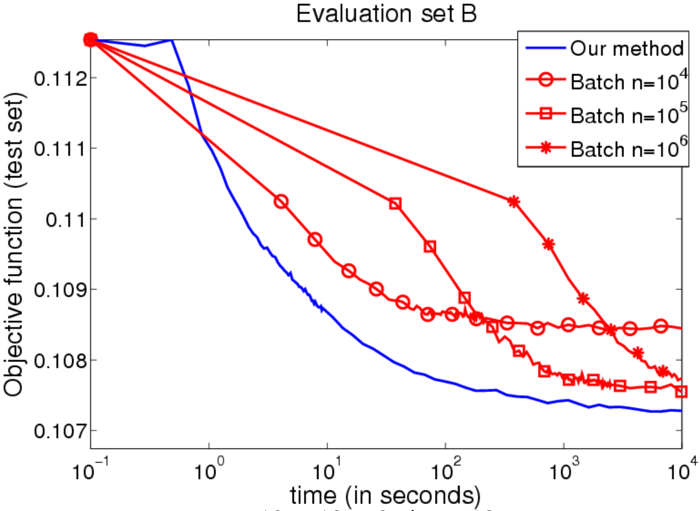
Experimental results, batch vs online



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

Online Dictionary Learning

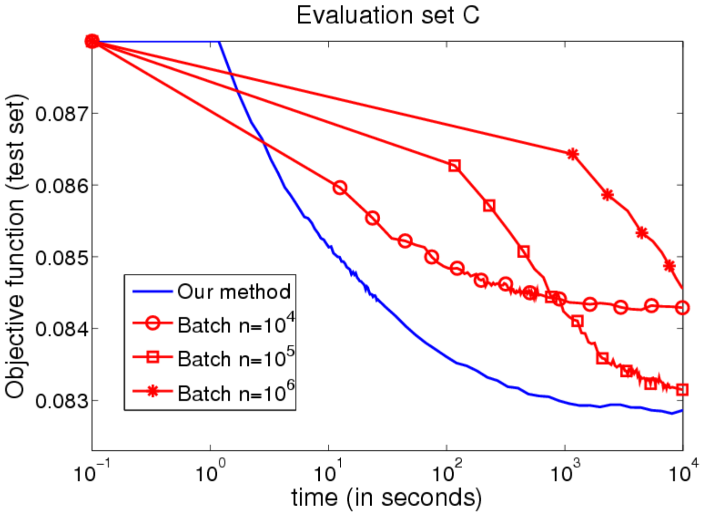
Experimental results, batch vs online



$m = 12 \times 12 \times 3, \; k = 512$

Online Dictionary Learning

Experimental results, batch vs online



$m = 16 \times 16$, $k = 1024$

Online Structured Matrix Factorization

With a structured regularization function φ [Jenatton et al., 2009]

$\varphi(\mathbf{D}) \triangleq \gamma_1 \sum_{j=1}^{K} \sum_{g \in \mathcal{G}} \max_{k \in g} |\mathbf{d}_j[k]| + \gamma_2 \|\mathbf{D}\|_{\mathsf{F}}^2.$

The proximal operator of φ can be computed by using network flow optimization [Mairal et al., 2010b].

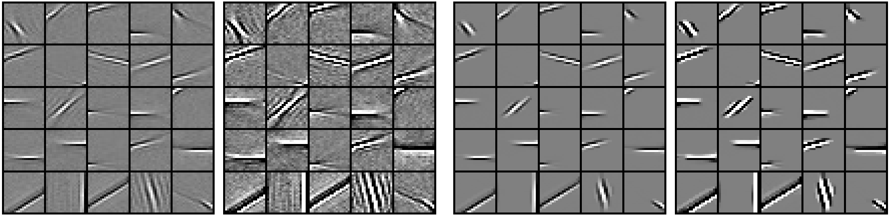


Figure : Left: subset of a larger dictionary obtained with ℓ_1; Right: subset obtained with φ after initialization with the dictionary on the left.

About 20 minutes per pass on the data on the 1.2GHz laptop CPU.

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. Böhning and B. G. Lindsay. Monotonicity of quadratic-approximation algorithms. *Annals of the Institute of Statistical Mathematics*, 40(4): 641–663, 1988.

E. J. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted l1 minimization. *Journal of Fourier Analysis and Applications*, 14: 877–905, 2008.

O. Cappé and E. Moulines. On-line expectation–maximization algorithm for latent data models. 71(3):593–613, 2009.

P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer, 2010.

References II

J. Duchi and Y. Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10: 2899–2934, 2009.

M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 54(12):3736–3745, December 2006.

G. Gasso, A. Rakotomamonjy, and S. Canu. Recovering sparse signals with non-convex penalties and DC programming. *IEEE Transactions on Signal Processing*, 57(12):4686–4698, 2009.

T. Jebara and A. Choromanska. Majorization for CRFs and latent likelihoods. In *Advances in Neural Information Processing Systems*, 2012.

R. Jenatton, J-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. Technical report, 2009. preprint arXiv:0904.3523v1.

References III

Emtiyaz Khan, Ben Marlin, Guillaume Bouchard, and Kevin Murphy. Variational bounds for mixed-data factor analysis. In *Advances in Neural Information Processing Systems*, 2010.

K. Lange, D.R. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. 9(1):1–20, 2000.

J. Mairal. Optimization with first-order surrogate functions. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2013a.

J. Mairal. Stochastic majorization-minimization algorithms for large-scale optimization. In *Advances in Neural Information Processing Systems*, 2013b.

J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *preprint arXiv:1402.4419*, 2014.

References IV

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 2010a.

J. Mairal, R. Jenatton, G. Obozinski, and F. Bach. Network flow algorithms for structured sparsity. In *Advances in Neural Information Processing Systems*, 2010b.

K. Mohan and M. Fazel. Iterative reweighted algorithms for matrix rank minimization. *Journal of Machine Learning Research*, (13):3441–3473, 2012.

R.M. Neal and G.E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*, 89, 1998.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. 19(4): 1574–1609, 2009.

References V

Y. Nesterov. Gradient methods for minimizing composite objective function. Technical report, CORE, 2007.

B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37: 3311–3325, 1997.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *arXiv:1309.2388*, 2013.

S. Shalev-Schwartz and T. Zhang. Proximal stochastic dual coordinate ascent. *preprint arXiv 1211.2717v1*, 2012.

S. Wright, R. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 2008.

L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11: 2543–2596, 2010.