Conditional gradient algorithms for machine learning

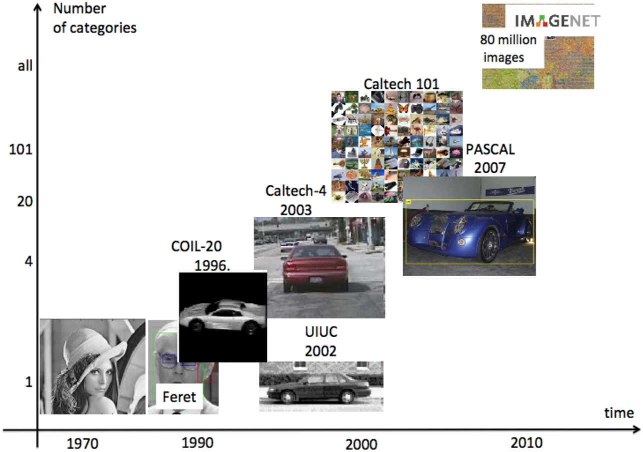
Zaid Harchaoui

LEAR and LJK, INRIA

Joint work with A. Juditsky (Grenoble U., France) and A. Nemirovski (GeorgiaTech)
and Matthijs Douze, Miro Dudik, Jerome Malick, Mattis Paulin

Gargantua day, Grenoble

The advent of large-scale datasets and "big learning"



From "The Promise and Perils of Benchmark Datasets and Challenges", D. Forsyth, A. Efros, F.-F. Li, A. Torralba and A. Zisserman, Talk at "Frontiers of Computer Vision"

Large-scale supervised learning

Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \mathcal{Y}$ be i.i.d. labelled training data, and $R_{\text{emp}}(\cdot)$ the empirical risk for any $\mathbf{W} \in \mathbb{R}^{d \times k}$.

Constrained formulation	Penalized formulation
$\begin{array}{ll} \text{minimize} & R_{\text{emp}}(\mathbf{W}) \\ \text{subject to} & \Omega(\mathbf{W}) \leq \rho \end{array}$	$\text{minimize} \quad \lambda\Omega(\mathbf{W}) + R_{\text{emp}}(\mathbf{W})$

Problem : minimize such objectives in the large-scale setting

$$\# \text{ examples} \gg 1, \quad \# \text{ features} \gg 1, \quad \# \text{ classes} \gg 1$$

Large-scale supervised learning

Large-scale supervised learning

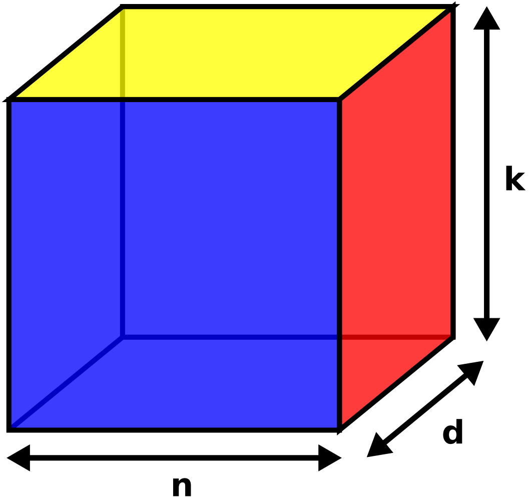
Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \mathcal{Y}$ be i.i.d. labelled training data, and $R_{\mathsf{emp}}(\cdot)$ the empirical risk for any $\mathbf{W} \in \mathbb{R}^{d \times k}$.

Constrained formulation	Penalized formulation
$$\begin{aligned} \text{minimize} \quad & R_{\mathsf{emp}}(\mathbf{W}) \\ \text{subject to} \quad & \Omega(\mathbf{W}) \leq \rho \end{aligned}$$	$$\text{minimize} \quad \lambda\Omega(\mathbf{W}) + R_{\mathsf{emp}}(\mathbf{W})$$

Problem : minimize such objectives in the large-scale setting

$$n \gg 1, \quad d \gg 1, \quad k \gg 1$$

Machine learning cuboid



Motivating example : multi-class classification with trace-norm penalty

Motivating the trace-norm penalty

- Embedding assumption : classes may embedded in a low-dimensional subspace of the feature space
- Computational efficiency : training time and test time efficiency require sparse matrix regularizers

Trace-norm

The trace-norm, aka nuclear norm, is defined as

$$\|\sigma(\mathbf{W})\|_1 = \sum_{p=1}^{\min(d,k)} \sigma_p(\mathbf{W})$$

where $\sigma_1(\mathbf{W}), \ldots, \sigma_{\min(d,k)}(\mathbf{W})$ denote the singular values of $\mathbf{W}$.

Large-scale supervised learning

Multi-class classification with trace-norm regularization

Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \mathcal{Y}$ be i.i.d. labelled training data, and $R_{\text{emp}}(\cdot)$ the empirical risk for any $\mathbf{W} \in \mathbb{R}^{d \times k}$.

Constrained formulation

$$\begin{aligned} \text{minimize} \quad & R_{\text{emp}}(\mathbf{W}) \\ \text{subject to} \quad & \|\sigma(\mathbf{W})\|_1 \leq \rho \end{aligned}$$

Penalized formulation

$$\text{minimize} \quad \lambda \|\sigma(\mathbf{W})\|_1 + R_{\text{emp}}(\mathbf{W})$$

- Trace-norm reg. penalty (Amit et al., 2007 ; Argyriou et al., 2007)
- Enforces a low-rank structure of $\mathbf{W}$ (sparsity of spectrum $\sigma(\mathbf{W})$)
- Convex problems

About the different formulations

"Alleged" equivalence

For a particular set of examples, for any value ρ of the constraint in the constrained formulation, there exists a value of λ in the penalized formulation so that the solutions of resp. the constrained formulation and the penalized formulation coincide.

Statistical learning theory

- theoretical results on penalized estimators and constrained estimators are of different nature $\rightarrow$ no rigorous comparison possible
- equivalence frequently called as the rescue depending on the theoretical tools available to jump from one formulation to the other

Summary

In practice

Recall that eventually "hyperparameters" $(\lambda, \rho, \varepsilon, \cdots)$ will have to be tuned.

Choose the formulation in which you can easily incorporate *prior knowledge*

Constrained formulation I
$$\underset{\mathbf{W}\in\mathbb{R}^{d\times k}}{\text{Minimize}} \left\{ \frac{1}{n}\sum_{i=1}^{n} \mathsf{Loss}_i \ : \ \|\sigma(\mathbf{W})\|_1 \leq \rho \right\}$$

Penalized formulation
$$\underset{\mathbf{W}\in\mathbb{R}^{d\times k}}{\text{Minimize}} \left\{ \frac{1}{n}\sum_{i=1}^{n} \mathsf{Loss}_i + \lambda \|\sigma(\mathbf{W})\|_1 \right\}$$

Constrained formulation II
$$\underset{\mathbf{W}\in\mathbb{R}^{d\times k}}{\text{Minimize}} \left\{ \lambda \|\sigma(\mathbf{W})\|_1 \ : \ \left| \frac{1}{n}\sum_{i=1}^{n} \mathsf{Loss}_i - R_{\mathsf{emp}}^{\mathsf{target}} \right| \leq \varepsilon \right\}$$

Learning with trace-norm penalty : a convex problem

Supervised learning with trace-norm regularization penalty

Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \mathcal{Y}$ be a set of i.i.d. labelled training data, with $\mathcal{Y} = \{0,1\}^k$ for multi-class classification

$$\operatorname*{Minimize}_{\mathbf{W} \in \mathbb{R}^{d \times k}} \quad \underbrace{\frac{1}{n} \sum_{i=1}^{n} \mathsf{Loss}_i + \lambda \|\sigma(\mathbf{W})\|_1}_{\text{convex}}$$

$\boxed{\text{Penalized formulation}}$

- Trace-norm reg. penalty (Amit et al., 2007 ; Argyriou et al., 2007)
- Enforces a low-rank structure of $\mathbf{W}$ (sparsity of spectrum $\sigma(\mathbf{W})$)
- Convex, but non-differentiable

Possible approaches

Generic approaches

- "Blind" approach : subgradient, bundle method $\rightarrow$ slow convergence rate
- Other approaches : alternating optimization, iteratively reweighted least-squares, etc. $\rightarrow$ no finite-time convergence guarantees

Learning with trace-norm penalty : convex but non-smooth

Supervised learning with trace-norm regularization penalty

Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \mathcal{Y}$ be a set of i.i.d. labelled training data, with $\mathcal{Y} = \{0,1\}^k$ for multi-class classification

$$\underset{\mathbf{W} \in \mathbb{R}^{d \times k}}{\text{Minimize}} \quad \underbrace{\lambda \left\| \sigma(\mathbf{W}) \right\|_1}_{\text{nonsmooth}} + \underbrace{\frac{1}{n} \sum_{i=1}^{n} \mathsf{Loss}_i}_{\text{smooth}}$$

where Loss_i is *e.g.* the multinomial logistic loss of i-th example

$$\mathsf{Loss}_i = \log \left(1 + \sum_{\ell \in \mathcal{Y} \setminus \{y_i\}} \exp\left\{ \mathbf{w}_\ell^T \mathbf{x}_i - \mathbf{w}_y^T \mathbf{x}_i \right\} \right)$$

Learning with trace-norm penalty : a convex problem

Supervised learning with trace-norm regularization penalty

Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \mathcal{Y}$ be a set of i.i.d. labelled training data, with $\mathcal{Y} = \{0, 1\}^k$ for multi-class classification

$$\underset{\mathbf{W} \in \mathbb{R}^{d \times k}}{\text{Minimize}} \quad \lambda \|\sigma(\mathbf{W})\|_1 + \frac{1}{n} \sum_{i=1}^{n} \mathsf{Loss}_i$$

Penalized formulation

Composite minimization for penalized formulation

Strengths of composite minimization (aka proximal-gradient)

- Attractive algorithms when proximal operator is cheap, as e.g. for vector ℓ_1-norm
- Accurate with medium-accuracy, finite-time accuracy guarantees

Proximal gradient

Algorithm

- **Initialize** : $\mathbf{W} = 0$
- **Iterate** :

$$\mathbf{W}_{t+1} = \mathsf{Prox}_{\lambda/L\Omega(\cdot)}\left(\mathbf{W}_t - \frac{1}{L}\nabla R_{\mathsf{emp}}(\mathbf{W}_t)\right)$$

$$\text{with } \mathsf{Prox}_{\lambda/L\Omega(\cdot)}(\mathbf{U}) := \min_{\mathbf{W}} \frac{1}{2}\|\mathbf{U} - \mathbf{W}\|^2 + \frac{\lambda}{L}\Omega(\mathbf{W})$$

Composite minimization for penalized formulation

Strengths of composite minimization (aka proximal-gradient)

- Attractive algorithms when proximal operator is cheap, as e.g. for vector ℓ_1-norm
- Accurate with medium-accuracy, finite-time accuracy guarantees

Weaknesses of composite minimization

- Inappropriate when proximal operator is expensive to compute
- Too sensitive to conditioning of design matrix (correlated features)

Situation with trace-norm, *i.e.* $\mathsf{Prox}_{\mu\Omega(\cdot)}(\cdot)$ with $\Omega(\cdot) = \|\cdot\|_{\sigma,1}$

- proximal operator corresponds to singular value thresholding, requiring an SVD running in $O(k\,\mathsf{rk}(\mathbf{W})^2)$ in time $\rightarrow$ impractical for large-scale problems

Alternative approach : conditional gradient

We want an algorithm with no SVD, i.e. without any projection or proximal step. Let us get some inspiration from the constrained setting.

Problem

$$\underset{\mathbf{W}\in\mathbb{R}^{d\times k}}{\text{Minimize}} \quad \left\{ \frac{1}{n}\sum_{i=1}^{n} \mathsf{Loss}_i \ : \ \mathbf{W} \in \rho \cdot \text{convex hull}\left(\{\mathbf{M}_t\}_{t\geq 1}\right) \right\}$$

Gauge/atomic decomposition of trace-norm

$$\|\sigma(\mathbf{W})\|_1 \;=\; \inf_{\theta} \left\{ \sum_{i=1}^{N} \theta_i \mid \exists N, \theta_i > 0, \mathbf{M}_i \in \mathcal{M} \text{ with } \mathbf{W} = \sum_{i=1}^{N} \theta_i \mathbf{M}_i \right\}$$

$$\mathcal{M} \;=\; \{\mathbf{u}\mathbf{v}^T \mid \mathbf{u}\in\mathbb{R}^d,\, \mathbf{v}\in\mathbb{R}^{\mathcal{Y}},\, \|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1\}$$

Conditional gradient descent

Algorithm

- **Initialize** : $\mathbf{W} = 0$
- **Iterate** : Find $\mathbf{M}_t \in \rho \cdot \text{convex hull}\,(\mathcal{M})$, such that

$$\mathbf{M}_t = \underbrace{\operatorname*{Arg\,max}_{\mathbf{M}_\ell \in \mathcal{M}} \langle \mathbf{M}_\ell, -\nabla R_{\mathsf{emp}}(\mathbf{W}_t) \rangle}_{\text{linear min. oracle}}$$

Perform line-search between $\mathbf{W}_t$ and $\mathbf{M}_t$

$$\mathbf{W}_{t+1} = (1-\delta)\mathbf{W}_t + \delta \mathbf{M}_t$$

Conditional gradient descent : example with trace-norm constraint

Algorithm

- **Initialize** : $\mathbf{W} = 0$
- **Iterate** : Find $\mathbf{M}_t \in \rho \cdot \text{convex hull}\,(\mathcal{M})$ such that

$$\mathbf{M}_t = \operatorname*{Arg\,max}_{\ell} \langle \mathbf{u}_\ell \mathbf{v}_\ell^T, -\nabla R_{\mathsf{emp}}(\mathbf{W}_t) \rangle$$
$$= \operatorname*{Arg\,max}_{\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1} \mathbf{u}^T(-\nabla R_{\mathsf{emp}}(\mathbf{W}_t))\mathbf{v}$$

i.e. compute *top pair of singular vectors* of $-\nabla R_{\mathsf{emp}}(\mathbf{W}_t)$.
Perform line-search between $\mathbf{W}_t$ and $\mathbf{M}_t$

$$\mathbf{W}_{t+1} = (1-\delta)\mathbf{W}_t + \delta \mathbf{M}_t$$

Conditional gradient descent

Algorithm

- **Initialize** : $\mathbf{W} = 0$
- **Iterate** : Find $\mathbf{M}_t \in \rho \cdot$ convex hull $(\mathcal{M})$ such that

$$\mathbf{M}_t = \underbrace{\operatorname*{Arg\,max}_{\mathbf{M}_\ell \in \mathcal{M}} \langle \mathbf{M}_\ell, -\nabla R_{\mathsf{emp}}(\mathbf{W}_t) \rangle}_{\text{easy}}$$

Perform line-search between $\mathbf{W}_t$ and $\mathbf{M}_t$

$$\mathbf{W}_{t+1} = (1-\delta)\mathbf{W}_t + \delta \mathbf{M}_t$$

Finite-time guarantee

Assumptions

(A) [Smoothness] The empirical risk $R_{\mathsf{emp}}(\cdot)$ is convex continuously differentiable on $D = \rho \cdot \mathrm{conv}(\mathcal{M})$, with Lipschitz constant L w.r.t D

Let $\{\mathbf{W}_t\}$ be a sequence generated by the conditional gradient algorithm. Then

$$F(\mathbf{W}_t) - F^\star \leq \frac{2L}{t+1}, \quad t = 1, 2, \ldots$$

Conditional gradient algorithm : review

Conditional gradient for constrained programming

- aka the Frank-Wolfe algorithm (1956, originally for quadratic programming)
- convergence results in general Banach spaces in (Demyanov & Rubinov, 1970)
- finite-time guarantees in (Pshenichnyi, 1975 ; Dunn, 1979)
- superseded by sequential quadratic programming in the early 80s, and ended up in the "mathematical programming" attic
- rediscovered several times and revisited with new variants in machine learning ;
  lately, (Hazan, 2008 ; Jaggi & Sulovsky, 2010 ; Tewari et al., 2011 ; Bach et al., 2012)

See (HJN, 2013) and (Jaggi, 2013) for modern proofs.

Conditional gradient algorithms

Question

- is it possible to design a conditional-gradient-type algorithm for penalized formulations ?

Conditional gradient vs Proximal gradient

Conditional gradient : iteration

$$\mathbf{W}_{t+1} = (1-\delta)\mathbf{W}_t + \delta \mathbf{M}_t$$

$$\mathbf{M}_t = \underbrace{\operatorname*{Arg\,max}_{\mathbf{M}_\ell \in \mathcal{M}} \langle \mathbf{M}_\ell, -\nabla R_{\mathsf{emp}}(\mathbf{W}_t)\rangle}_{\text{easy}}$$

Proximal gradient : iteration

$$\mathbf{W}_{t+1} = \mathsf{Prox}_{\lambda/L\Omega(\cdot)}\left(\mathbf{W}_t - 1/L\nabla R_{\mathsf{emp}}(\mathbf{W}_t)\right)$$

$$\mathsf{Prox}_{\lambda/L\Omega(\cdot)}(\mathbf{U}) := \underbrace{\min_{\mathbf{W}} \ \frac{1}{2}\|\mathbf{U}-\mathbf{W}\|^2 + \frac{\lambda}{L}\Omega(\mathbf{W})}_{\text{hard}}$$

Conditional gradient approach for penalized formulations

Let $K \subset E$ a closed convex cone, E a euclidean space, and $\|\cdot\|$ a norm on E.

Problem

$$\underset{\mathbf{W} \in K}{\text{Minimize}} \quad \lambda \|\mathbf{W}\| \quad + \frac{1}{n} \sum_{i=1}^{n} \mathsf{Loss}_i(\mathbf{W})$$

Penalized formulation

Sketch

- Augment the variable $\mathbf{W}$ by one dimension to handle the regularization penalty
- Perform a sequence of iterations akin to the conditional gradient iterations
- and so on...

Turning the problem into a cone constrained problem

Problem

Introducing the variable $Z := [\mathbf{W}, r]$, we get

$$\begin{aligned} \text{minimize} \quad & F(Z) \\ \text{subject to} \quad & Z \in K^+ \end{aligned}$$

where

$$\begin{aligned} F(Z) &:= \lambda r + \frac{1}{n}\sum_{i=1}^{n} \mathsf{Loss}_i(\mathbf{W}) \\ K^+ &:= \{[\mathbf{W}; r],\ \mathbf{W} \in K,\ \|\mathbf{W}\| \leq r\}\ . \end{aligned}$$

linear minimization oracle

First-order information and linear minimization oracle

For any W, we can get

- $R_{\mathsf{emp}}(\mathbf{W})$ the empirical risk
- $\nabla R_{\mathsf{emp}}(\mathbf{W})$ the gradient of the empirical risk

For any $g \in E^*$ we have access to a *linear minimization oracle*

$$\mathsf{Oracle}(g) := \operatorname*{Arg\,max}_{\mathbf{W} \in K_1} \langle \mathbf{W}, -g \rangle \ .$$

where

$$K_1 := \{\mathbf{W} \in K, \|\mathbf{W}\| \leq 1\} \ .$$

Linear minimization oracle

First-order information and linear minimization oracle

For any W, we can get

- $R_{\mathsf{emp}}(\mathbf{W})$ the empirical risk
- $\nabla R_{\mathsf{emp}}(\mathbf{W})$ the derivative of the empirical risk

and any iteration t we have access to a *linear minimization oracle*

$$\mathsf{Oracle}(g) := \operatorname*{Arg\,max}_{\mathbf{W} \in K_1} \langle \mathbf{W}, -g \rangle \ .$$

where

$$K_1 := \{ \mathbf{W} \in K, \, \|\mathbf{W}\| \leq 1 \} \ .$$

Conditional gradient for penalized formulation

Algorithm

- **Inputs** : instrumental bound D^+ on $\|\mathbf{W}^\star\|$, first-order oracle, and minim. oracle
- **Iterate** : Compute $\nabla R_{\mathsf{emp}}(\mathbf{W}_t)$ at $Z_t = (\mathbf{W}_t, r_t)$

  Call the linear minimization oracle

$$\mathrm{Oracle}(\nabla R_{\mathsf{emp}}(\mathbf{W}_t)) := \underbrace{\operatorname*{Arg\,max}_{\mathbf{W} \in K_1} \langle \mathbf{W}, -\nabla R_{\mathsf{emp}}(\mathbf{W}_t) \rangle}_{\text{linear minimization oracle}} .$$

...

$\boxed{\text{The instrumental bound } D^+ \text{ can be loose.}}$

Conditional gradient for penalized formulation

Algorithm

- **Inputs** : instrumental bound D^+ on $\|\mathbf{W}^\star\|$, first-order oracle, and minim. oracle
- **Iterate** :
  Compute $\nabla R_{\mathsf{emp}}(\mathbf{W}_t)$ at $Z_t = (\mathbf{W}_t, r_t)$

  Get $\bar{Z}_t = [\mathsf{Oracle}(\nabla R_{\mathsf{emp}}(\mathbf{W}_t)), 1]$ from the linear minimization oracle.

  Perform line-search to get

$$Z_{t+1} \in \mathsf{argmin}_Z \left\{ F(Z), \;\; Z \in \mathsf{Conv}\{0, Z_t, \, D^+ \bar{Z}_t\} \right\} .$$

$\boxed{\text{The instrumental bound } D^+ \text{ can be loose.}}$

Conditional gradient for penalized formulation

Algorithm

- **Inputs** : instrumental bound D^+ on $\|\mathbf{W}^\star\|$, first-order oracle, and minim. oracle
- **Iterate** :
  Compute $\nabla R_{\text{emp}}(\mathbf{W}_t)$ at $Z_t = (\mathbf{W}_t, r_t)$

  Get $\bar{Z}_t = [\text{Oracle}(\nabla R_{\text{emp}}(\mathbf{W}_t)), 1]$ from the linear minimization oracle.

  Perform line-search to get

$$
\begin{aligned}
Z_{t+1} &= \alpha_{t+1}\bar{Z}_t + \beta_{t+1} Z_t \\
(\alpha_{t+1}, \beta_{t+1}) &= \operatorname*{Arg\,min}_{\alpha,\beta} \left\{ F(\alpha \bar{Z}_t + \beta Z_t),\ \alpha + \beta \leq 1,\ \alpha \geq 0,\ \beta \geq 0 \right\} .
\end{aligned}
$$

- **Output** : $\mathbf{W}_T$ can be retrieved from $Z_T = [\mathbf{W}_T, r_T]$.

Memory-based extensions : convex-hull

Convex-hull memory-based extension ("restricted simplicial acceleration")

Instead to the $2D$ line-search, we can perform at each iteration for some $M > 0$

$$Z_{t+1} \in \operatorname*{Arg\,min}_{Z}\{F(Z),\ Z \in \mathcal{C}_t\}\ .$$

where

$$\mathcal{C}_t = \begin{cases} \text{Conv}\{0;\ D^+\bar{Z}_0,\ ...,\ D^+\bar{Z}_t\}, & t \leq M\ , \\ \text{Conv}\{0; Z_{t-M+1}, ...,\ Z_t;\ D^+\bar{Z}_{t-M+1}, ...,\ D^+\bar{Z}_t\}, & t > M\ . \end{cases}$$

Important computational considerations

- Line-search sub-problem can be solved with ellipsoid algorithm
- Maintaining the factorization of $\mathbf{W}$ along iterations is essential for speed

Memory-based extensions : conic-hull

Conic-hull memory-based extension

Instead to the $2D$ line-search, we can perform at each iteration for some $M > 0$

$$Z_{t+1} \in \operatorname*{Arg\,min}_{Z}\{F(Z),\ Z \in \mathcal{B}_t\}\ .$$

where

$$\mathcal{B}_t = \begin{cases} \text{Conic}\{\bar{Z}_0, ..., \bar{Z}_t\}, & t \leq M\,, \\ \text{Conic}\{Z_{t-M+1}, ..., Z_t;\ \bar{Z}_{t-M+1}, ..., \bar{Z}_t\}, & t > M\,. \end{cases}$$

$M = +\infty$: we recover the **Atom-Descent** algorithm of (DHM, 2012)

Important computational considerations

- Line-search sub-problem can be solved with coordinate-descent
- Maintaining factorization of $\mathbf{W}$ along iterations essential for speed

Finite-time guarantee

Assumptions

(A) [Smoothness] The empirical risk $R_{\text{emp}}(\cdot)$ is convex continuously differentiable with Lipschitz constant L.

(B) [Effective domain] There exists $D < 1$ such that $\|\mathbf{W}\| \leq r$ and $r + R_{\text{emp}}(\mathbf{W}) < R_{\text{emp}}(\mathbf{0})$ imply that $r \leq D$

Let $\{Z_t\}$ be a sequence generated by the algorithm. Then

$$F(Z_t) - F^\star \leq \frac{8LD^2}{t+1}, \quad t = 2, 3, \dots$$

Finite-time guarantee

Finite-time guarantee

Let $\{Z_t\}$ be a sequence generated by the algorithm. Then

$$F(Z_t) - F^\star \leq \frac{8LD^2}{t+1}, \quad t = 2, 3, \ldots$$

Important remark

The $O(1/t)$ convergence rate depends on D (unknown and not required by the algorithm), but *does not depend* on D^+ ! (known and required by the algorithm).

Finite-time guarantee

Finite-time guarantee

Let $\{Z_t\}$ be a sequence generated by the algorithm. Then

$$F(Z_t) - F^\star \leq \frac{8LD^2}{t+1}, \quad t = 2, 3, \ldots$$

$\boxed{\text{Theoretical convergence rate is independent of } D^+.}$

Generalization to gauge regularization penalty

Gauge regularization penalty

- Gauge definition : $\Omega(\mathbf{W}) := \inf\{t \geq 0 \mid \mathbf{W} \in t\mathcal{B}\}$
- Unit "ball" : $\mathcal{B} := \operatorname{conv} \mathcal{M}$
- Atoms set : $\mathcal{M} = \{\mathbf{M}_i \in \mathcal{R}^{d \times k} : \ i \in \mathcal{I}\}$ be a compact set of matrices, called *atoms* $\rightarrow$ "overcomplete basis"

Generalization to gauge regularization penalty

Properties

- $\Omega(t\mathbf{W}) = t\Omega(\mathbf{W})$ for all $\mathbf{W}$ and $t \geq 0$
- $\Omega(\mathbf{W} + \mathbf{W}') \leq \Omega(\mathbf{W}) + \Omega(\mathbf{W}')$ for all $\mathbf{W}$ and $\mathbf{W}'$.

Additional properties

Assuming $\mathbf{0} \in \operatorname{int} \mathcal{B}$, we also have

- $\Omega(\mathbf{W}) \geq 0$, with equality if and only if $\mathbf{W} = \mathbf{0}$
- $\{\mathbf{W} : \Omega(\mathbf{W}) \leq t\} = t\mathcal{B}$ for $t \geq 0$, i.e., level sets are compact.

Polar duality

- Support function : $\Omega^\circ(\mathbf{G}) := \sup_{\mathbf{M}\in\mathcal{B}} \langle \mathbf{M}, \mathbf{G} \rangle = \sup_{\mathbf{M}\in\mathcal{M}} \langle \mathbf{M}, \mathbf{G} \rangle$.

Some examples

Examples of gauges with their atomic decomposition

$$\begin{aligned}
\sum_{i,j} |\mathbf{W}_{i,j}| \quad & \mathcal{M}_{\textsf{lasso}} = \left\{ \pm \mathbf{e}_j \mathbf{e}_\ell^T \mid j \in \{1, \ldots, d\}, \, \ell \in \{1, \ldots, k\} \right\} \\
\sum_{i} \|\mathbf{W}_{i,:}\| \quad & \mathcal{M}_{\textsf{gp-lasso}} = \{ \mathbf{e}_j \mathbf{v}^T \mid j \in \{1, \ldots, d\}, \, \mathbf{v} \in \mathcal{R}^k, \, \|\mathbf{v}\|_2 = 1 \} \\
\sum_{p} \sigma_p(\mathbf{W}_{i,:}) \quad & \mathcal{M}_{\textsf{tr-norm}} = \{ \mathbf{u}\mathbf{v}^T \mid \mathbf{u} \in \mathcal{R}^d, \, \mathbf{v} \in \mathcal{R}^k, \, \|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1 \}
\end{aligned}$$

Conclusion and perspectives

Large-scale learning

- conditional gradient algorithm for learning problems with atomic-decomposition-norm regularization
- efficient and competitive algorithm for large-scale multi-class classification
- scheme applies to all problems with atomic decomposition norm regularizers (Harchaoui et al., 2011, Chandrasekaran et al., 2012) : nuclear-norm, total-variation norm, overlapping-blocks sparse norm, etc.

Extensions

- non-smooth loss functions ; see (Pierucci et al., ICCOPT 2013)
- online/mini-batch extensions
- path-following extensions

References

References

- *Atom-descent with smoothing for machine learning with non-smooth loss function*, F. Pierucci, Z. Harchaoui, A. Juditsky, A. Nemirovski, ICCOPT 2013
- *Conditional gradient algorithms for norm-regularized smooth convex optimization*, Z. Harchaoui, A. Juditsky, A. Nemirovski, sub. Math. Prog. A, 2013
- *Large-scale classification with trace-norm regularization*, Z. Harchaoui, M. Douze, M. Paulin, J. Malick, CVPR 2012
- *Lifted coordinate descent for learning with trace-norm regularization penalty*, M. Dudik, Z. Harchaoui, J. Malick, AISTATS 2011
- *Learning with matrix gauge regularizers*, M. Dudik, Z. Harchaoui, J. Malick, NIPS Opt. 2011