Conditional gradient algorithms for large-scale learning

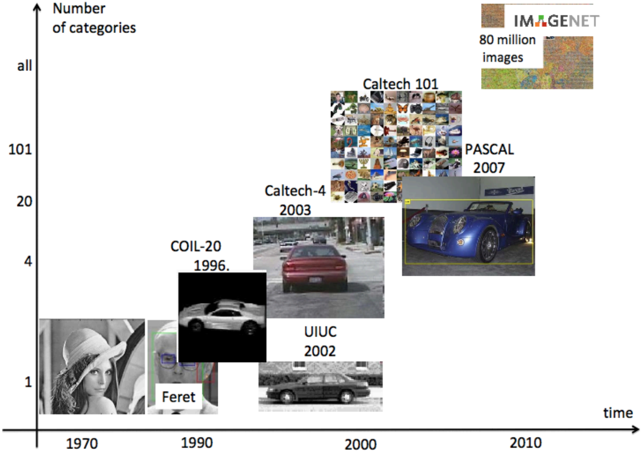
Zaid Harchaoui

LEAR team, INRIA

Joint work with A. Juditsky (Grenoble U., France)
and A. Nemirovski (GeorgiaTech)

IHES

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

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

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

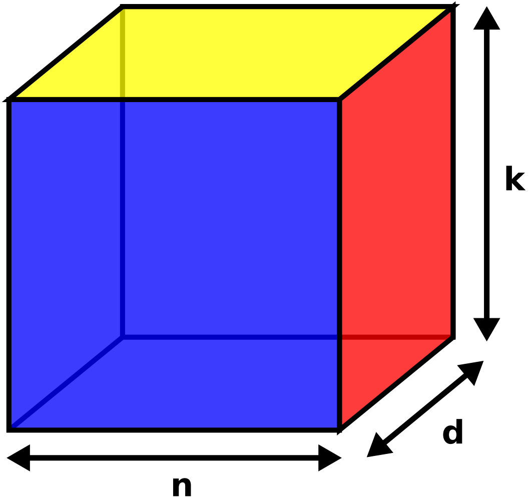
Penalized formulation

$$\text{minimize} \quad \lambda\Omega(\mathbf{W}) + R_{\mathsf{emp}}(\mathbf{W})$$

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 $R_{\mathsf{emp}}(\cdot)$ the empirical risk for any $\mathbf{W} \in \mathbb{R}^{d \times k}$.

Constrained formulation

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

Penalized formulation

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

- 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 will have to be tuned.

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

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

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

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

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

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

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

- 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

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

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

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

- proximal operator corresponds to singular value thresholding, requiring an SVD running in $O(k\,\mathrm{rk}(\mathbf{W})^2)$ 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

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

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 (Jaggi & Sulovsky, 2010)

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

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

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 $\mathbf{M}_t \in \rho \cdot \text{convex hull}\,(\mathcal{M})$ such that

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

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 (Pshenichnyi, 1975 ; Dunn, 1979)

Assumptions

(A) [Smoothness] The empirical risk $R_{\text{emp}}(\cdot)$ is convex continuously differentiable on $D = \rho \cdot \operatorname{conv}(\mathcal{M})$, 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 (Jaggi, 2013) for a nice review and sharp theoretical guarantees.

Question

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

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

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

Linearization oracle

First-order information and linearization 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

and for any $g \in E^*$ we have access to a *linearization oracle*

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

where

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

Linearization oracle

First-order information and linearization 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 *linearization oracle*

$$\mathsf{Oracle}(\nabla R_{\mathsf{emp}}(\mathbf{W}_t)) := \operatorname*{Arg\,min}_{\mathbf{W} \in K_1} \langle \mathbf{W}, \nabla R_{\mathsf{emp}}(\mathbf{W}_t) \rangle \,.$$

where

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

Conditional gradient for penalized formulation

Algorithm

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

  Make a call to the linearization oracle

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

...

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

Conditional gradient for penalized formulation

Algorithm

- **Inputs** : instrumental bound D^+ on $\|x^\star\|$, 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 linearization oracle.

  Perform line-search to get

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

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

Idea

Line-search

For any $\rho \geq 0$, consider the linear form

$$(\xi, \rho) \mapsto \lambda\rho + \langle \nabla R_{\mathsf{emp}}(\mathbf{W}), \xi \rangle \; .$$

As ρ varies in $0 \leq \rho \leq D^{+}$, the set of minima of the linear form span the segment

$$S = \{\rho[\mathsf{Oracle}\nabla R_{\mathsf{emp}}(\mathbf{W}); 1], \; 0 \leq \rho \leq D^{+}\} \; .$$

S can easily be identified by calls resp. to the first-order oracle and the linearization oracle.

Conditional gradient for penalized formulation

Algorithm

- **Inputs** : instrumental bound D^+ on $\|x^\star\|$, 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 linearization 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 $Z_T = [\mathbf{W}_T, r_T]$.

Computational considerations

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

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

Computational considerations

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

Finite-time guarantee

Assumptions

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

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

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

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

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

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

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

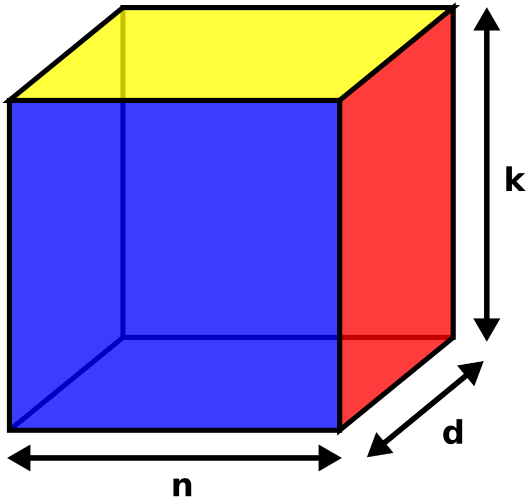
Experimental results

Synthetic data benchmark

- Inspired by the benchmark of optimization algorithms for sparsity-inducing vector penalties of (Bach et al., 2011)
- Varying scales, varying strength of penalty λ, varying conditioning of design matrix (low-correlation and high-correlation of features)

Real data benchmark

- ImageNet dataset
- Subset of classes "Vertebrate-craniate" subset, yielding $k = 1,043$
- State-of-the-art visual descriptors (Fisher vectors, Perronnin & Dance, 2007) $d = 65,000$


k
d
n

Experimental results

Computational considerations

1. parallelized and multi-threaded objective evaluation and gradient evaluation
2. efficient matrix computations for high-dimensional features

Experimental results

Matrix size $k \times d$	Memory-less version		with memory $M = 5$	
	N_{it}	T_{cpu}	N_{it}	T_{cpu}
2000×2000	172.9	349.77	99.7	125.13
4000×4000	153.4	$1.035\ 10^3$	88.23	$0.575\ 10^3$
8000×8000	195.3	$2.755\ 10^3$	120.45	$1.284\ 10^3$
16000×16000	230.2	$6.585\ 10^3$	134.34	$3.413\ 10^3$
32000×32000	271.4	$15.342\ 10^3$	140.45	$7.343\ 10^3$
1043×65000	182.0	$2.101\ 10^3$	110.34	$0.925\ 10^3$

Table : memoryless version vs. version with memory $M = 5$; N_{it} : total number of method iterations; T_{cpu} : CPU usage (sec) reported by MATLAB.

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

- online/mini-batch extensions
- path-following extensions

References

References

- *Conditional gradient algorithms for norm-regularized smooth convex optimization*, Z. Harchaoui, A. Juditsky, A. Nemirovski, 2013
- *Conditional gradient algorithms for machine learning*, Z. Harchaoui, A. Juditsky, A. Nemirovski, NIPS Optimization Workshop, 2012
- *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