# Nonsmooth Optimization for Statistical Learning with Structured Matrix Regularization


PhD defense of
**Federico Pierucci**

Thesis supervised by

Prof. Zaid Harchaoui
Prof. Anatoli Juditsky
Dr. Jérôme Malick

Université Grenoble Alpes
Inria - Thoth and BiPoP



Inria Grenoble
June 23rd, 2017

# Application 1: Collaborative filtering

**Collaborative filtering** for recommendation systems
Matrix completion optimization problem.

Ratings **X**:

| | film 1 | film 2 | film 3 |
|---|---|---|---|
| Albert | ⋆⋆⋆⋆⋆ | ⋆⋆ | ⋆ |
| Ben | | | ⋆⋆ |
| Celine | ⋆ | ⋆⋆⋆⋆⋆ | ⋆⋆⋆⋆ |
| Diana | ⋆ | | |
| Elia | | ⋆⋆ | |
| Franz | ⋆⋆⋆⋆ | | ⋆ |

- **Data:** for user $i$ and movie $j$
  $X_{ij} \in \mathbb{R}$, with $(i,j) \in \mathcal{I}$: known ratings
- **Purpose**: predict a future rating
  New $(i,j) \longmapsto X_{ij} = ?$

# Application 1: Collaborative filtering

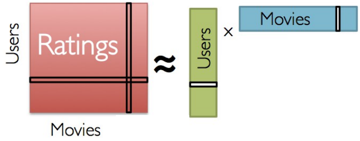
**Collaborative filtering** for recommendation systems
Matrix completion optimization problem.

Ratings $\mathbf{X}$:

| | film 1 | film 2 | film 3 |
|---|---|---|---|
| Albert | ⋆⋆⋆⋆⋆ | ⋆⋆ | ⋆ |
| Ben | | | ⋆⋆ |
| Celine | ⋆ | ⋆⋆⋆⋆⋆ | ⋆⋆⋆⋆ |
| Diana | ⋆ | | |
| Elia | | ⋆⋆ | |
| Franz | ⋆⋆⋆⋆ | | ⋆ |

- **Data:** for user $i$ and movie $j$
  $X_{ij} \in \mathbb{R}$, with $(i,j) \in \mathcal{I}$: known ratings
- **Purpose**: predict a future rating
  New $(i,j) \longmapsto X_{ij} = ?$

**Low rank assumption**:
Movies can be divided into a small number of types




For example:

$$\min_{\mathbf{W} \in \mathbb{R}^{d \times k}} \quad \frac{1}{N} \sum_{(i,j) \in \mathcal{I}} |\mathbf{W}_{ij} - X_{ij}| \quad + \quad \lambda \, \|\mathbf{W}\|_{\sigma,1}$$

$\|\mathbf{W}\|_{\sigma,1}$ Nuclear norm = sum of singular values
- convex function
- surrogate of rank

# Application 2: Multiclass classification

**Multiclass classification** of images
Example: ImageNet challenge

- **Data** $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}^k$ : pairs of (image, category)
- **Purpose**: predict the category for a new image
  New image $x \longmapsto y = ?$

$\longmapsto$ `marmot`

$\longmapsto$ `edgehog`

$\longmapsto$ ?

## Application 2: Multiclass classification

**Multiclass classification** of images
Example: ImageNet challenge

- **Data** $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}^k$ : pairs of (image, category)
- **Purpose**: predict the category for a new image
  New image $x \longmapsto y = ?$

$\longmapsto$ marmot

$\longmapsto$ edgehog

$\longmapsto$ ?

**Low rank assumption**: The features are assumed to be embedded in a lower dimensional space

Multiclass version of support vector machine (SVM):

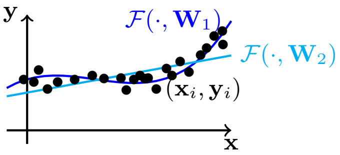
$$\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \quad \frac{1}{N}\sum_{i=1}^{N} \underbrace{\max\left\{0,\ 1 + \max_{r \text{ s.t. } r\neq y_i} \{\mathbf{W}_r^\top x_i - \mathbf{W}_{y_i}^\top x_i\}\right\}}_{\left\|(\mathcal{A}_{x,y}\mathbf{W})_+\right\|_\infty} \quad + \quad \lambda\,\|\mathbf{W}\|_{\sigma,1}$$

$\mathbf{W}_j \in \mathbb{R}^d$ : the $j$-th column of $\mathbf{W}$

# Matrix learning problem

- These two problems have the form:

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \underbrace{\frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{y}_i, \overbrace{\mathcal{F}(\mathbf{x}_i,\mathbf{W})}^{\hat{\mathbf{y}}_i})}_{=:R(\mathbf{W}),\quad \text{empirical risk}} + \lambda \underbrace{\|\mathbf{W}\|}_{\text{regularization}}$$




- Notation

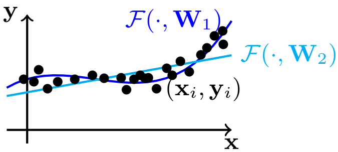
Prediction
- $\mathcal{F}$ prediction function
- $\ell$ loss function

Data:
- $N$ number of examples
- $\mathbf{x}_i$ feature vector
- $\mathbf{y}_i$ outcome
- $\hat{\mathbf{y}}_i$ predicted outcome

# Matrix learning problem

- These two problems have the form:

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \underbrace{\frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{y}_i, \overbrace{\mathcal{F}(\mathbf{x}_i, \mathbf{W})}^{\hat{\mathbf{y}}_i})}_{=:R(\mathbf{W}),\quad \text{empirical risk}} + \lambda \underbrace{\|\mathbf{W}\|}_{\text{regularization}}$$




- Notation

<u>Prediction</u>

$\mathcal{F}$ prediction function

$\ell$ loss function

<u>Data:</u>

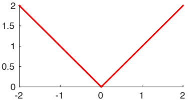
$N$ number of examples

$\mathbf{x}_i$ feature vector

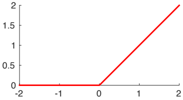
$\mathbf{y}_i$ outcome

$\hat{\mathbf{y}}_i$ predicted outcome

- Nonsmooth empirical risk:




$g(\xi) = |\xi|$




$\max\{0, \xi\}$

- Challenges
  - ⋆ Large scale: N, k, d
  - ⋆ Robust learning:

Generalization $\rightarrow$ **nonsmooth** regularization
Noisy data, outliers $\rightarrow$ **nonsmooth** empirical risk

# My thesis in one slide

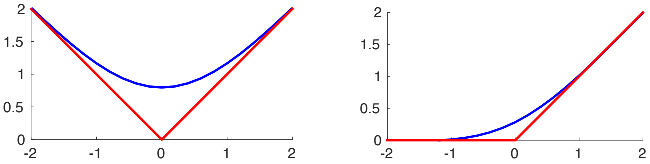
$$\underbrace{\min_{\mathbf{W}}}_{\text{2nd contribution}} \quad \underbrace{\frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{y}_i, \mathcal{F}(\mathbf{x}_i, \mathbf{W}))}_{\text{1st contribution}} \quad + \quad \lambda \underbrace{\|\mathbf{W}\|}_{\text{3rd contribution}}$$

1 - Smoothing techniques
2 - Conditional gradient algorithms
3 - Group nuclear norm

## Part 1
## Unified view of smoothing techniques for first order optimization

**Motivations**:

- Smoothing is a key tool in optimization
- Smooth loss allows the use of gradient-based optimization

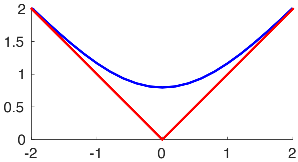



$g(\xi) = |\xi|$ $\quad$ $g(\xi) = \max\{0, \xi\}$

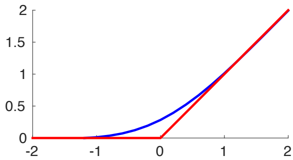
## Part 1
## Unified view of smoothing techniques for first order optimization

**Motivations**:

- Smoothing is a key tool in optimization
- Smooth loss allows the use of gradient-based optimization




$g(\xi) = |\xi|$




$g(\xi) = \max\{0, \xi\}$

**Contributions**:

- Unified view of smoothing techniques for nonsmooth functions
- New example: smoothing of top-$k$ error (for list ranking and classification)
- Study of algorithms = smoothing + state of art algorithms for smooth problems

## Part 2
## Conditional gradient algorithms for doubly nonsmooth learning

**Motivations**:

- Common matrix learning problems formulated as

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \quad \underbrace{R(\mathbf{W})}_{\text{nonsmooth emp.risk}} \quad + \quad \lambda \quad \underbrace{\|\mathbf{W}\|}_{\text{nonsmooth regularization}}$$

- Nonsmooth empirical risk, e.g. L1 norm $\rightarrow$ robust to noise and outlyers
- Standard nonsmooth optimization methods not always scalable (e.g. nuclear norm)

## Part 2
## Conditional gradient algorithms for doubly nonsmooth learning

**Motivations**:

- Common matrix learning problems formulated as

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \quad \underbrace{R(\mathbf{W})}_{\text{nonsmooth emp.risk}} \quad + \quad \lambda \quad \underbrace{\|\mathbf{W}\|}_{\text{nonsmooth regularization}}$$

- Nonsmooth empirical risk, e.g. L1 norm $\rightarrow$ robust to noise and outlyers
- Standard nonsmooth optimization methods not always scalable (e.g. nuclear norm)

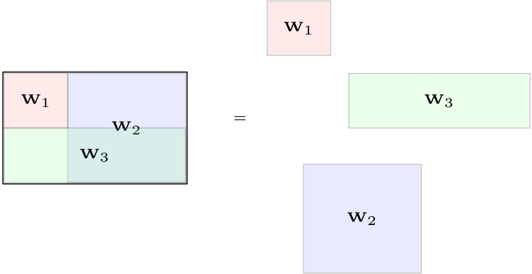
**Contributions**:

- New algorithms based on (composite) conditional gradient
- Convergence analysis: rate of convergence + guarantees
- Some numerical experiences on real data

## Part 3
## Regularization by group nuclear norm

**Motivations**:

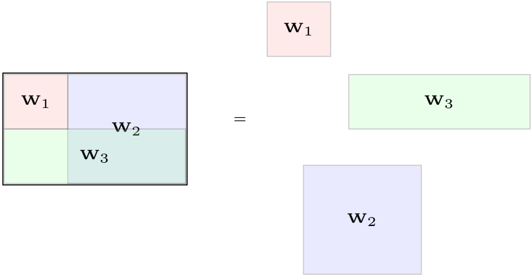
- Structured matrices can join information coming from different sources
- Low-rank models improve robustness and dimensionality reduction



## Part 3
## Regularization by group nuclear norm

**Motivations**:

- Structured matrices can join information coming from different sources
- Low-rank models improve robustness and dimensionality reduction




**Contributions**:

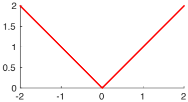
- Definition of a new norm for matrices with underlying groups
- Analysis of its convexity properties
- Used as regularizer $\rightarrow$ provides low rank by groups and aggregate models

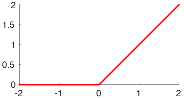
# Outline

# Smoothing techniques

**Purpose**:
to smooth a convex function

$$g : \mathbb{R}^n \to \mathbb{R}$$

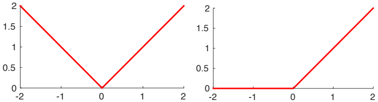


## Smoothing techniques

**Purpose**:
to smooth a convex function

$$g : \mathbb{R}^n \to \mathbb{R}$$




**Two techniques**:
1) Product convolution [Bertsekas 1978] [Duchi et al. 2012]

$$g_\gamma^{pc}(\xi) := \int_{\mathbb{R}^n} g(\xi - \mathbf{z}) \, \frac{1}{\gamma} \mu \left( \frac{\mathbf{z}}{\gamma} \right) \, \mathrm{d}\mathbf{z} \qquad \mu : \text{probability density}$$

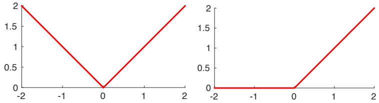
2) Infimal convolution [Moreau 1965] [Nesterov 2007] [Beck, Teboulle 2012]

$$g_\gamma^{ic}(\xi) := \inf_{\mathbf{z} \in \mathbb{R}^n} \left\{ g(\xi - \mathbf{z}) + \gamma \, \omega \left( \frac{\mathbf{z}}{\gamma} \right) \right\} \qquad \omega : \text{smooth convex function}$$

# Smoothing techniques

**Purpose**:
to smooth a convex function

$$g : \mathbb{R}^n \to \mathbb{R}$$




**Two techniques**:
1) Product convolution [Bertsekas 1978] [Duchi et al. 2012]

$$g_\gamma^{pc}(\xi) := \int_{\mathbb{R}^n} g(\xi - \mathbf{z}) \, \frac{1}{\gamma} \mu\left(\frac{\mathbf{z}}{\gamma}\right) \mathrm{d}\mathbf{z} \qquad \mu : \text{probability density}$$

2) Infimal convolution [Moreau 1965] [Nesterov 2007] [Beck, Teboulle 2012]

$$g_\gamma^{ic}(\xi) := \inf_{\mathbf{z} \in \mathbb{R}^n} \left\{ g(\xi - \mathbf{z}) + \gamma \, \omega\left(\frac{\mathbf{z}}{\gamma}\right) \right\} \qquad \omega : \text{smooth convex function}$$

**Result**
- $g_\gamma$ is uniform approximation of $g$, i.e. $\exists m, \, M \geq 0 \, : \quad -\gamma m \leq g_\gamma(\mathbf{x}) - g(\mathbf{x}) \leq \gamma M$
- $g_\gamma$ is $L_\gamma$-smooth, i.e. $g_\gamma$ differentiable, convex,
  $\|\nabla g_\gamma(\mathbf{x}) - \nabla g_\gamma(\mathbf{y})\|_* \leq L_\gamma \|\mathbf{x} - \mathbf{y}\|$ ($L_\gamma$ proportional to $\frac{1}{\gamma}$)

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$

# Smoothing surrogates of nonsmooth functions

- **Purpose:** obtain $g_\gamma$ to be used into algorithms
  - $\star$ (possibly) explicit expression
  - $\star$ easy to evaluate numerically

- **Elementary example** (in $\mathbb{R}$) :
absolute value $g(x) = |x|$

$\star$ Product convolution, with $\mu(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$

$$g_\gamma^c(x) = -xF(-\tfrac{x}{\gamma}) - \frac{\sqrt{2}}{\sqrt{\pi}}\gamma e^{-\frac{x^2}{2\gamma^2}} + xF(\tfrac{x}{\gamma})$$

$F(x) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}}\, dt$ cumulative distribution of Gaussian

$\star$ Infimal convolution, with $\omega(x) = \frac{1}{2}\|x\|^2$

$$g_\gamma^{ic}(x) = \begin{cases} \frac{1}{2\gamma}x^2 + \frac{\gamma}{2} & \text{if } |x| \le \gamma \\ |x| & \text{if } |x| > \gamma \end{cases}$$

- **Motivating nonsmooth function: top-$k$ loss** (next)

# Motivating nonsmooth functions: top-$k$ loss
# Example: top-3 loss

- Top-3 loss for Classification

# Motivating nonsmooth functions: top-$k$ loss
# Example: top-3 loss

- Top-3 loss for Classification

| Cat |
|---|

Ground truth

$\longleftrightarrow$

| | |
|---|---|
| 1 | Paper towel |
| 2 | Wall |
| 3 | **Cat** |

Prediction

$\Longrightarrow$ loss = 0

Good prediction if the true class is among the first 3 predicted.

# Motivating nonsmooth functions: top-$k$ loss
## Example: top-3 loss

- Top-3 loss for Classification

Cat (Ground truth) $\longleftrightarrow$

| | Prediction |
|---|---|
| 1 | Paper towel |
| 2 | Wall |
| 3 | **Cat** |

$\Longrightarrow$ loss = 0

Good prediction if the true class is among the first 3 predicted.

- Top-3 loss for Ranking

Grund truth:

| | |
|---|---|
| 1 | Janis Joplins |
| 2 | David Bowie |
| 3 | Eric Clapton |
| 4 | Patty Smith |
| 5 | Jean-Jacques Goldman |
| 6 | Francesco Guccini |
| | ⋮ |

$\longleftrightarrow$

Prediction:

| | |
|---|---|
| 1 | **David Bowie** |
| 2 | Patty Smith |
| 3 | **Janis Joplins** |

$\Longrightarrow$ loss $= 0 + \frac{1}{3} + 0$

Predict an ordered list, the loss counts the mismatches to the true list

# Smoothing of top-$k$

**Convex top-$k$ error function**, written as a sublinear function

$$g(\mathbf{x}) = \max_{\mathbf{z}\in\mathcal{Z}} \langle \mathbf{x}, \mathbf{z} \rangle$$

$$\mathcal{Z} := \left\{ \mathbf{z} \in \mathbb{R}^n : \ 0 \le z_i \le \tfrac{1}{k}, \ \sum_{i=1}^{n} z_i \le 1 \right\} \quad = \text{cube} \cap \text{simplex}$$

- Case $k = 1$ **Top-1**

$g(\mathbf{x}) = \|\mathbf{x}_+\|_\infty = \max\{0, \max_i \{\mathbf{x}_i\}\}$

Infimal convolution with $\omega(\mathbf{x}) = \left( \sum_{i=1}^{n} x_i \ln(x_i) - x_i \right)^*$

$$g_\gamma(\mathbf{x}) = \begin{cases} \gamma \left(1 + \ln \sum_{i=1}^{n} e^{\frac{x_i}{\gamma}}\right) & \text{if} \quad \sum_{i=1}^{n} e^{\frac{x_i}{\gamma}} > 1 \\ \gamma \sum_{i=1}^{n} e^{\frac{x_i}{\gamma}} & \text{if} \quad \sum_{i=1}^{n} e^{\frac{x_i}{\gamma}} \le 1 \end{cases}$$

## Smoothing of top-$k$

**Convex top-$k$ error function**, written as a sublinear function

$$g(\mathbf{x}) = \max_{\mathbf{z}\in\mathcal{Z}} \langle \mathbf{x}, \mathbf{z} \rangle$$

$$\mathcal{Z} := \left\{ \mathbf{z} \in \mathbb{R}^n : \; 0 \le z_i \le \tfrac{1}{k}, \sum_{i=1}^{n} z_i \le 1 \right\} \quad = \text{cube} \cap \text{simplex}$$

• Case $k = 1$ **Top-1**

$g(\mathbf{x}) = \|\mathbf{x}_+\|_\infty = \max\{0, \max_i \{\mathbf{x}_i\}\}$

Infimal convolution with $\omega(\mathbf{x}) = \left( \sum_{i=1}^{n} x_i \ln(x_i) - x_i \right)^*$

$$g_\gamma(\mathbf{x}) = \begin{cases} \gamma \left( 1 + \ln \sum_{i=1}^{n} e^{\frac{x_i}{\gamma}} \right) & \text{if} \quad \sum_{i=1}^{n} e^{\frac{x_i}{\gamma}} > 1 \quad \longleftarrow \text{Classification} \\ \end{cases}$$

Same result as in statistics [Hastie et al., 2008]

$\gamma = 1 \rightarrow$ multinomial logistic loss

## Smoothing of top-$k$ case $k > 1$

Infimal convolution with $\omega = \frac{1}{2}\|\cdot\|^2$

$$g_\gamma(\mathbf{x}) = -\lambda_\star(\mathbf{x},\gamma) + \sum_{i=1}^{n} H_\gamma(x_i + \lambda_\star(\mathbf{x},\gamma))$$

$$H_\gamma(t) = \begin{cases} 0 & t < 0 \\ \frac{1}{2}t^2 & t \in [0, \frac{1}{k}] \\ \frac{t}{k} - \frac{1}{k^2} & t > \frac{1}{k} \end{cases}$$

- We need to solve an auxiliary problem (smooth dual problem)

Evaluate $g_\gamma(\mathbf{x})$ through the dual problem

Define

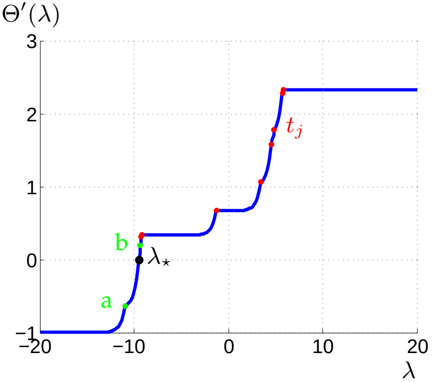
$$P_{\mathbf{x}} := \{x_i, x_i - k : \ i = 1 \dots n\}$$

$$\Theta'(\lambda) = 1 - \sum_{t_j \in P_{\mathbf{x}}} \pi_{[0,1/k]}(t_j + \lambda)$$

Find

$$a, b \in P_{\mathbf{x}} \quad \text{s.t.} \quad \Theta'(a) \leq 0 \leq \Theta'(b)$$

$$\lambda_\star(\mathbf{x},\gamma) = \max\left\{0, a - \frac{\Theta'(a)(b-a)}{\Theta'(b)-\Theta'(a)}\right\}$$



# Outline

# Matrix learning problem

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \quad \underbrace{R(\mathbf{W})}_{\text{nonsmooth}} \quad + \quad \lambda \underbrace{\Omega(\mathbf{W})}_{\text{nonsmooth}}$$

**Empirical risk** $\quad R(\mathbf{W}) := \frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{W}, \mathbf{x}_i, \mathbf{y}_i)$

- Top-k for ranking and multiclass classification $\quad \ell_1(\mathbf{W}, \mathbf{x}, \mathbf{y}) := \|(\mathcal{A}_{\mathbf{x},\mathbf{y}}\mathbf{W})_+\|_\infty$
- L1 for regression $\quad \ell_1(\mathbf{W}, \mathbf{x}, \mathbf{y}) := |\mathcal{A}_{\mathbf{x},\mathbf{y}}\mathbf{W}|$

**Regularizer** (typically norm) $\quad \Omega(\mathbf{W})$

- Nuclear norm $\|\mathbf{W}\|_{\sigma,1}$ $\longrightarrow$ sparsity on singular values
- L1 norm $\|\mathbf{W}\|_1 := \sum_{i=1}^{d}\sum_{j=1}^{k} |\mathbf{W}_{ij}|$ $\longrightarrow$ sparsity on entries
- Group nuclear norm $\Omega_{\mathcal{G}}(\mathbf{W})$ $\quad$ (of contribution 3)

sparsity $\leftrightarrow$ feature selection

# Existing algorithms for nonsmooth optimization

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \underbrace{R(\mathbf{W})}_{\text{nonsmooth}} + \lambda \underbrace{\Omega(\mathbf{W})}_{\text{nonsmooth}}$$

- Subgradient, bundle algorithms [Nemirovski, Yudin 1976] [Lemarechal 1979]
- Proximal algorithms [Douglas, Rachford 1956]

Algorithms are not scalable for nuclear norm: **iteration cost** $\simeq$ full SVD $= O(dk^2)$

# Existing algorithms for nonsmooth optimization

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \underbrace{R(\mathbf{W})}_{\text{nonsmooth}} + \underbrace{\lambda\,\Omega(\mathbf{W})}_{\text{nonsmooth}}$$

- Subgradient, bundle algorithms [Nemirovski, Yudin 1976] [Lemarechal 1979]
- Proximal algorithms [Douglas, Rachford 1956]

Algorithms are not scalable for nuclear norm: **iteration cost** $\simeq$ full SVD $= O(dk^2)$

**What if the loss were smooth?**

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \underbrace{S(\mathbf{W})}_{\text{smooth}} + \underbrace{\lambda\,\Omega(\mathbf{W})}_{\text{nonsmooth}}$$

Algorithms with faster convergence when S is smooth

- Proximal gradient algorithms
  [Nesterov 2005] [Beck, Teboulle, 2009]
  Still not scalable for nuclear norm: **iteration cost** $\simeq$ full SVD
- (Composite) conditional gradient algorithms
  [Frank, Wolfe, 1956][Harchaoui, Juditsky, Nemirovski, 2013]
  Efficient iterations for nuclear norm:
  **iteration cost** $\simeq$ compute largest singular value $= O(dk)$

# Composite conditional gradient algorithm

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \underbrace{S(\mathbf{W})}_{\text{smooth}} + \lambda \underbrace{\Omega(\mathbf{W})}_{\text{nonsmooth}}$$

State of art algorithm:

**Composite conditional gradient algorithm**

Let $\mathbf{W}_0 = \mathbf{0}$
$r_0$ such that $\Omega(\mathbf{W}_\star) \leq r_0$
**for** $t = 0 \dots T$ **do**
Compute

$$\mathbf{Z}_t = \operatorname*{argmin}_{\mathbf{D} \text{ s.t. } \Omega(\mathbf{D}) \leq r_t} \langle \nabla S(\mathbf{W}_t), \mathbf{D} \rangle \qquad \text{[gradient step]}$$

$$\alpha_t, \beta_t = \operatorname*{argmin}_{\alpha,\beta\geq 0;\ \alpha+\beta\leq 1} S(\alpha \mathbf{Z}_t + \beta \mathbf{W}_t) + \lambda(\alpha+\beta) r_t \qquad \text{[optimal stepsize]}$$

Update

$$\mathbf{W}_{t+1} = \alpha_t \mathbf{Z}_t + \beta_t \mathbf{W}_t$$

$$r_{t+1} = (\alpha_t + \beta_t) r_t$$

**end for**

where
$\mathbf{W}_t, \mathbf{Z}_t, \mathbf{D} \in \mathbb{R}^{d\times k}$

Efficient and scalable for some $\Omega$ e.g. nuclear norm, where $\mathbf{Z}_t = uv^\top$

# Conditional gradient despite nonsmooth loss

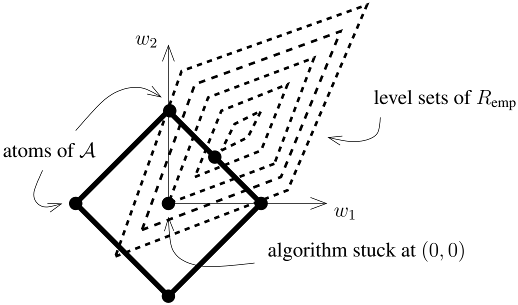
Use conditional gradient replacing $\nabla S(\mathbf{W}_t)$ with a subgradient $s_t \in \partial R(\mathbf{W}_t)$

# Conditional gradient despite nonsmooth loss

Use conditional gradient replacing $\nabla S(\mathbf{W}_t)$ with a subgradient $s_t \in \partial R(\mathbf{W}_t)$

Simple counter example in $\mathbb{R}^2$

$$\min_{\mathbf{w}\in\mathbb{R}^2} \|A\mathbf{w} + b\|_1 + \|\mathbf{w}\|_1$$



# Smoothed composite conditional gradient algorithm

**Idea**: Replace the nonsmooth loss with a smoothed loss

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \underbrace{R(\mathbf{W})}_{\text{nonsmooth}} + \lambda\,\Omega(\mathbf{W}) \quad\longrightarrow\quad \min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \underbrace{R_\gamma(\mathbf{W})}_{\text{smooth}} + \lambda\,\Omega(\mathbf{W})$$

$\{R_\gamma\}_{\gamma>0}$ family of smooth approximations of $R$

Let $\mathbf{W}_0 = \mathbf{0}$

$r_0$ such that $\Omega(\mathbf{W}_\star) \le r_0$

**for** $t = 0 \dots T$ **do**

Compute

$$\mathbf{Z}_t = \operatorname*{argmin}_{\mathbf{D}\text{ s.t. }\Omega(\mathbf{D})\le r_t} \langle \nabla R_{\gamma_t}(\mathbf{W}_t), \mathbf{D}\rangle$$

$$\alpha_t, \beta_t = \operatorname*{argmin}_{\alpha,\beta\ge 0;\ \alpha+\beta\le 1} R_{\gamma_t}(\alpha\mathbf{Z}_t + \beta\mathbf{W}_t) + \lambda(\alpha+\beta)r_t$$

Update

$$\mathbf{W}_{t+1} = \alpha_t\mathbf{Z}_t + \beta_t\mathbf{W}_t$$

$$r_{t+1} = (\alpha_t + \beta_t)r_t$$

**end for**

$\alpha_t, \beta_t$ = stepsize $\quad \gamma_t$ = smoothing parameter

**Note**: We want solve the initial 'doubly nonsmooth' problem

# Convergence analysis

Doubly nonsmooth problem

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} F(\mathbf{W}) = R(\mathbf{W}) + \lambda\,\Omega(\mathbf{W})$$

$\mathbf{W}_\star$ optimal solution
$\gamma_t$ = smoothing parameter $(\neq$ stepsize)

## Theorems of convergence

- Fixed smoothing of $R$ $\quad\gamma_t = \gamma$

$$F(\mathbf{W}_t) - F(\mathbf{W}_\star) \quad\leq\quad \gamma M + \frac{2}{\gamma(t+14)}$$

Dimensionality freedom of $M$ depends on $\omega$ or $\mu$
The best $\gamma$ depends on the required accuracy $\varepsilon$

- Time-varying smoothing of $R$ $\quad\gamma_t = \frac{\gamma_0}{\sqrt{t+1}}$

$$F(\mathbf{W}_t) - F(\mathbf{W}_\star) \quad\leq\quad \frac{C}{\sqrt{t}}$$

Dimensionality freedom of $C$ depends on $\omega$ or $\mu$, $\gamma_0$ and $\|\mathbf{W}_\star\|$

# Algoritm implementation

**Package**
All the Matlab code written from scratch, in particular:

- Multiclass SVM
- Top-$k$ multiclass SVM
- All other smoothed functions

**Memory**
Efficient memory management

- Tools to operate with low rank variables
- Tools to work with sparse sub-matrices of low rank matrices (collaborative filtering)

**Numerical experiments - 2 motivating applications**

- Fix smoothing - matrix completion (regression)
- Time-varying smoothing - top-5 multiclass classification

## Fix smoothing
## Example with matrix completion, regression

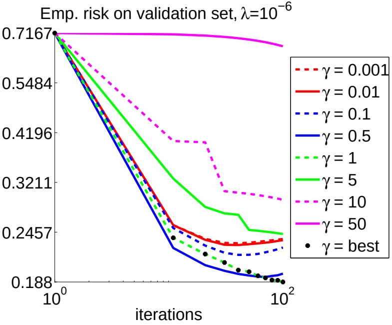
**Data**: Movielens
$d = 71\,567$ users
$k = 10\,681$ movies
10 000 054 ratings $(= 1.3\%)$
( normalized into [0,1] )

**Benchmark**
- Iterates $\mathbf{W}_t$ generated on a train set
- We observe $R(\mathbf{W}_t)$ on the validation set
- Choose the best $\gamma$ that minimizes $R(\mathbf{W}_t)$ in the validation set

# Fix smoothing
# Example with matrix completion, regression

**Data**: Movielens
$d = 71\,567$ users
$k = 10\,681$ movies
10 000 054 ratings ($= 1.3\%$)
( normalized into [0,1] )

**Benchmark**
• Iterates $\mathbf{W}_t$ generated on a train set
• We observe $R(\mathbf{W}_t)$ on the validation set
• Choose the best $\gamma$ that minimizes $R(\mathbf{W}_t)$ in the validation set




Each $\gamma$ gives a different optimization problem

Tiny smoothing $\rightarrow$ slower convergence
Large smoothing $\rightarrow$ objective much different than the initial one

# Time-varying smoothing
# Example with top-5 multiclass classification

**Data**: ImageNet
$k = 134$ classes
$N = 13\,400$ images
**Features**: BOW
$d = 4096$ features

**Benchmark**
• Iterates $\mathbf{W}_t$ generated on a train set
• We observe top-5 misclassification error on the validation set
• To compare: find best fixed smoothing parameter (using the other benchmark)

# Time-varying smoothing
# Example with top-5 multiclass classification

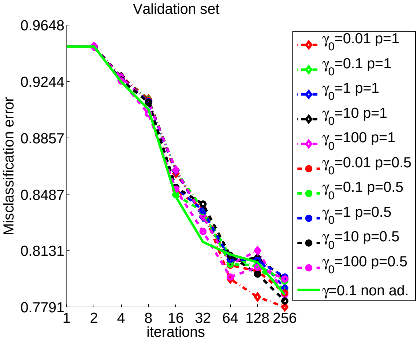
**Data**: ImageNet
$k = 134$ classes
$N = 13\,400$ images
**Features**: BOW
$d = 4096$ features

**Benchmark**
- Iterates $\mathbf{W}_t$ generated on a train set
- We observe top-5 misclassification error on the validation set
- To compare: find best fixed smoothing parameter (using the other benchmark)

**Time-varying** smoothing parameter

$$\gamma_t = \frac{\gamma_0}{(1+t)^p}$$

$p \in \left\{\frac{1}{2}, 1\right\}$




No need to tune $\gamma_0$:
- Time-varying smoothing matches the performances of the best experimentally tuned fixed smoothing

# Outline

# Group nuclear norm

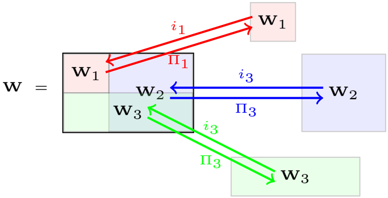
- Matrix generalization of the popular **group lasso norm** [Turlach et al., 2005] [Yuan and Lin, 2006] [Zhao et al., 2009] [Jacob et al., 2009]

- Nuclear norm $\|\mathbf{W}\|_{\sigma,1}$ : sum of singular values of $\mathbf{W}$

# Group nuclear norm

- Matrix generalization of the popular **group lasso norm** [Turlach et al., 2005] [Yuan and Lin, 2006] [Zhao et al., 2009] [Jacob et al., 2009]

- Nuclear norm $\|\mathbf{W}\|_{\sigma,1}$ : sum of singular values of $\mathbf{W}$




$i_g$: immersion
$\Pi_g$: projection

$\mathcal{G} = \{1, 2, 3\}$

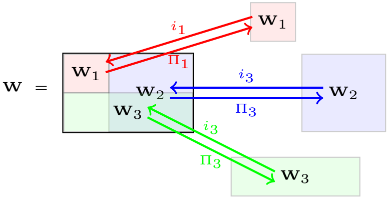
$$\Omega_{\mathcal{G}}(\mathbf{W}) := \min_{\mathbf{w} = \sum_{g \in \mathcal{G}} i_g(\mathbf{w}_g)} \sum_{g \in \mathcal{G}} \alpha_g \|\mathbf{W}_g\|_{\sigma,1}$$

[Tomioka, Suzuki 2013] non-overlapping groups

# Group nuclear norm

- Matrix generalization of the popular **group lasso norm**
[Turlach et al., 2005] [Yuan and Lin, 2006] [Zhao et al., 2009] [Jacob et al., 2009]

- Nuclear norm $\|\mathbf{W}\|_{\sigma,1}$ : sum of singular values of $\mathbf{W}$




$i_g$: immersion
$\Pi_g$: projection

$\mathcal{G} = \{1, 2, 3\}$

$$\Omega_{\mathcal{G}}(\mathbf{W}) := \min_{\mathbf{W}=\sum_{g\in\mathcal{G}} i_g(\mathbf{W}_g)} \sum_{g\in\mathcal{G}} \alpha_g \|\mathbf{W}_g\|_{\sigma,1}$$

[Tomioka, Suzuki 2013] non-overlapping groups

Convex analysis - theoretical study

- Fenchel conjugate $\Omega^*_{\mathcal{G}}$
- Dual norm $\Omega^\circ_{\mathcal{G}}$
- Expression of $\Omega_{\mathcal{G}}$ as a support function
- Convex hull of functions involving rank

# Convex hull - Results

In words, the **convex hull** is the largest convex function lying below the given one

Properly restricted to a ball,
the nuclear norm is the convex hull of rank [Fazel 2001] $\rightarrow$ generalization

**Theorem**

Properly restricted to a ball, group nuclear norm is the **convex hull** of:

- The 'reweighted group rank' function:

$$\Omega_{\mathcal{G}}^{\text{rank}}(\mathbf{W}) := \inf_{\mathbf{W}=\sum_{g\in\mathcal{G}} i_g(\mathbf{W}_g)} \sum_{g\in\mathcal{G}} \alpha_g \operatorname{rank}(\mathbf{W}_g)$$

- The 'reweighted restricted rank' function:

$$\Omega^{\text{rank}}(\mathbf{W}) := \min_{g\in\mathcal{G}} \quad \alpha_g \operatorname{rank}(\mathbf{W}) + \delta_g(\mathbf{W})$$

$\delta_g$ indicator function

Learning with group nuclear norm enforces low-rank property on groups

# Learning with group nuclear norm

Usual optimization algorithms can handle the group nuclear norm:
⋆ composite conditional gradient algorithms
⋆ (accelerated) proximal gradient algorithms

# Learning with group nuclear norm

Usual optimization algorithms can handle the group nuclear norm:
⋆ composite conditional gradient algorithms
⋆ (accelerated) proximal gradient algorithms

**Illustration with proximal gradient optimization algorithm**
The key computations are parallelized on each group

Good scalability when there are **many small** groups

• prox of group nuclear norm

$$\operatorname{prox}_{\gamma\Omega_{\mathcal{G}}}((\mathbf{W}_g)_g) = \left(\mathbf{U}_g D_\gamma(\mathbf{S}_g)\mathbf{V}_g{}^\top\right)_{g\in\mathcal{G}}$$

where $D_\gamma$ : soft thresholding operator

• SVD decomposition

$$\mathbf{W}_g = \mathbf{U}_g\mathbf{S}_g\mathbf{V}_g{}^\top$$

$$D_\gamma(\mathbf{S}) = \operatorname{Diag}(\{\max\{s_i - \gamma, 0\}\}_{1\le i\le r}).$$

# Learning with group nuclear norm

Usual optimization algorithms can handle the group nuclear norm:
⋆ composite conditional gradient algorithms
⋆ (accelerated) proximal gradient algorithms

**Illustration with proximal gradient optimization algorithm**
The key computations are parallelized on each group

Good scalability when there are **many small** groups

• prox of group nuclear norm

$$\mathrm{prox}_{\gamma\Omega_{\mathcal{G}}}((\mathbf{W}_g)_g) = \left(\mathbf{U}_g D_\gamma(\mathbf{S}_g)\mathbf{V}_g{}^\top\right)_{g\in\mathcal{G}}$$

where $D_\gamma$ : soft thresholding operator

• SVD decomposition

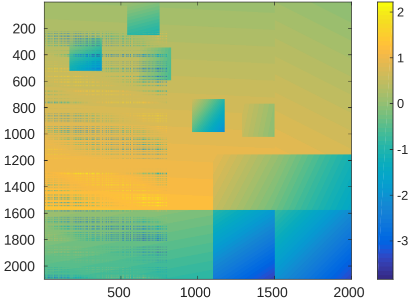
$$\mathbf{W}_g = \mathbf{U}_g\mathbf{S}_g\mathbf{V}_g{}^\top$$

$$D_\gamma(\mathbf{S}) = \mathrm{Diag}(\{\max\{s_i - \gamma, 0\}\}_{1\le i\le r}).$$

Package in Matlab, in particular:
→ vector space of group nuclear norm, overloading of + *

# Numerical illustration: matrix completion

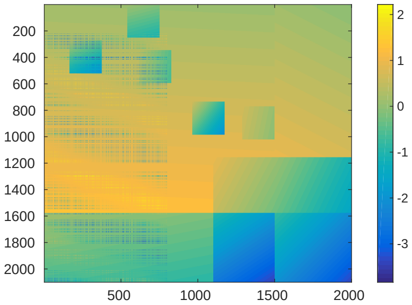
**"Ground truth"**




Synthetic low rank matrix $\mathbf{X}$
sum of 10 rank-1 groups
normalized to have $\mu = 0, \sigma = 1$

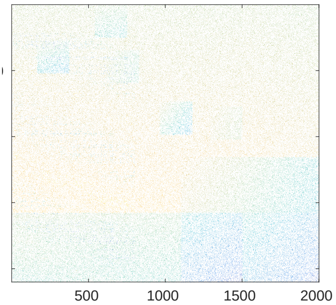
# Numerical illustration: matrix completion

**"Ground truth"**



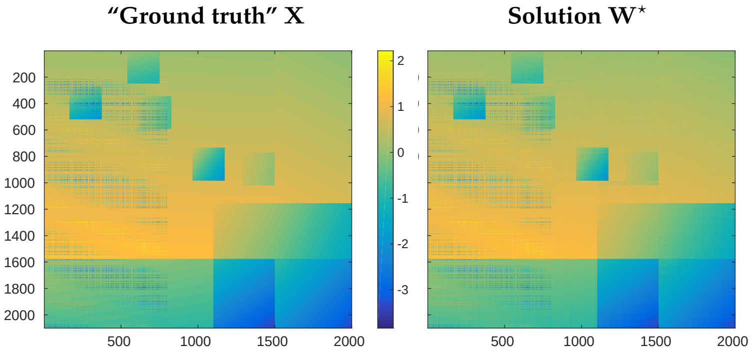

Synthetic low rank matrix $\mathbf{X}$
sum of 10 rank-1 groups
normalized to have $\mu = 0$, $\sigma = 1$

**Observation**




Uniform 10% sampling $\mathbf{X}_{ij}$
with $(i, j) \in \mathcal{I}$
Gaussian additive noise $\sigma = 0.2$

# Numerical illustration: matrix completion




Recovery error:

$$\frac{1}{2N}\left\|\mathbf{W}^{\star}-\mathbf{X}\right\|^{2}=0.0051$$

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \quad \frac{1}{N}\sum_{(i,j)\in\mathcal{I}} \tfrac{1}{2}(\mathbf{W}_{ij}-\mathbf{X}_{ij})^{2} \quad + \quad \lambda\,\Omega_{\mathcal{G}}(\mathbf{W})$$

# Outline

# Summary

- Smoothing
  - ⋆ Versatile tool in optimization
  - ⋆ Ways to combine smoothing with many existing algorithms
- Time-varying smoothing
  - ⋆ Theory: minimization convergence analysis
  - ⋆ Practice: recover the best, no need to tune $\gamma$
- Group nuclear norm
  - ⋆ Theory and practice to combine groups and rank sparsity
  - ⋆ Overlapping groups

# Perspectives

- Smoothing for faster convergence:
  Moreau-Yosida smoothing can be used to improve the condition number of poorly conditioned objectives before applying linearly-convergent convex optimization algorithms [Hongzhou et al. 2017]
- Smoothing for better prediction:
  Smoothing can be adapted to properties of the dataset and be used to improve the prediction performance of machine learning algorithms
- Learning group structure and weights for better prediction:
  The group structure in the group nuclear norm can be learned to leveraged underlying structure and improve the prediction
- Extensions to group Schatten norm
- Potential applications of group nuclear norm
  - ⋆ multi-attribute classification
  - ⋆ multiple tree hierarchies
  - ⋆ dimensionality reduction, feature selection e.g. concatenate features, avoid PCA

# Perspectives

- Smoothing for faster convergence:
  Moreau-Yosida smoothing can be used to improve the condition number of poorly conditioned objectives before applying linearly-convergent convex optimization algorithms [Hongzhou et al. 2017]
- Smoothing for better prediction:
  Smoothing can be adapted to properties of the dataset and be used to improve the prediction performance of machine learning algorithms
- Learning group structure and weights for better prediction:
  The group structure in the group nuclear norm can be learned to leveraged underlying structure and improve the prediction
- Extensions to group Schatten norm
- Potential applications of group nuclear norm
  - ⋆ multi-attribute classification
  - ⋆ multiple tree hierarchies
  - ⋆ dimensionality reduction, feature selection e.g. concatenate features, avoid PCA

Thank You