Frank-Wolfe Splitting via Augmented Lagrangian Method



Gauthier Gidel[1] Fabian Pedregosa[2] Simon Lacoste-Julien[1]

[1]MILA, DIRO Université de Montréal [2]UC Berkeley & ETH Zurich

April 2018

Why Frank-Wolfe is wonderful.

- Constrained optimization algorithm:

$$\min_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x})$$

  f convex, $\mathcal{C}$ convex *compact*.

- Interesting for highly structured constraint sets:

Alignment constraint: [Alayrac et al., 2016]

Permutahedron: [Lancia and Serafini, 2018] [Evangelopoulos et al., 2017]

Why Frank-Wolfe is wonderful.

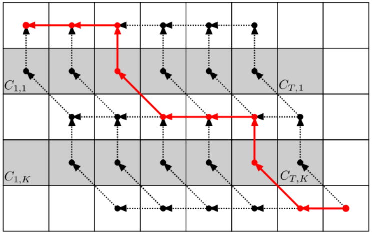
- Constrained optimization algorithm:

$$\min_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x})$$

  f convex, $\mathcal{C}$ convex *compact*.

- Interesting for highly structured constraint sets:

  Alignment constraint:



[Alayrac et al., 2016]

Permutahedron: [Lancia and Serafini, 2018] [Evangelopoulos et al., 2017]

Why Frank-Wolfe is wonderful.

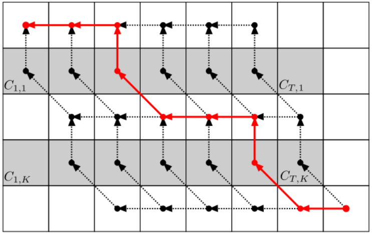
- Constrained optimization algorithm:

$$\min_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x})$$

f convex, $\mathcal{C}$ convex *compact.*

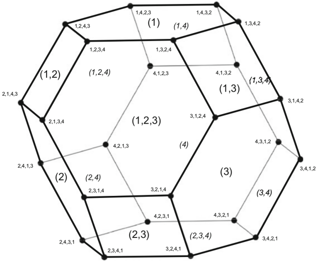
- Interesting for highly structured constraint sets:

Alignment constraint:



[Alayrac et al., 2016]

Permutahedron:



[Lancia and Serafini, 2018]
[Evangelopoulos et al., 2017]

Why Frank-Wolfe is wonderful.

- Constrained optimization algorithm:

$$\min_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x})$$

  f convex, $\mathcal{C}$ convex *compact*.

- Interesting when projection is **not practical**:

  ~~**Projection**~~ **Linear Minimization Oracle**

- When projection is practical **better use** projected gradient method.

Why Frank-Wolfe sometimes is not enough.

- FW requires *linear minimization* (LMO) over these set.

$$\mathrm{LMO}(\boldsymbol{d}) := \arg\min_{\boldsymbol{x}\in\mathcal{C}} \langle \boldsymbol{d}, \boldsymbol{x} \rangle$$

- Intersection of constraint sets: $\mathcal{C}_1 \cap \mathcal{C}_2$.
- $\mathrm{LMO}_{\mathcal{C}_1\cap\mathcal{C}_2}(\boldsymbol{d})$ may be too expensive.

- FW-AL just requires $\mathrm{LMO}_{\mathcal{C}_1}(\boldsymbol{d})$ and $\mathrm{LMO}_{\mathcal{C}_2}(\boldsymbol{d})$.

Why Frank-Wolfe sometimes is not enough.

- FW requires *linear minimization* (LMO) over these set.

$$\mathrm{LMO}(\boldsymbol{d}) := \arg\min_{\boldsymbol{x} \in \mathcal{C}} \langle \boldsymbol{d}, \boldsymbol{x} \rangle$$

- Intersection of constraint sets: $\mathcal{C}_1 \cap \mathcal{C}_2$.
- $\mathrm{LMO}_{\mathcal{C}_1 \cap \mathcal{C}_2}(\boldsymbol{d})$ may be too expensive.

- FW-AL just requires $\mathrm{LMO}_{\mathcal{C}_1}(\boldsymbol{d})$ and $\mathrm{LMO}_{\mathcal{C}_2}(\boldsymbol{d})$.

Why Frank-Wolfe sometimes is not enough.

- FW requires *linear minimization* (LMO) over these set.

$$\mathrm{LMO}(\boldsymbol{d}) := \arg\min_{\boldsymbol{x}\in\mathcal{C}} \langle \boldsymbol{d}, \boldsymbol{x}\rangle$$

- Intersection of constraint sets: $\mathcal{C}_1 \cap \mathcal{C}_2$.
- $\mathrm{LMO}_{\mathcal{C}_1\cap\mathcal{C}_2}(\boldsymbol{d})$ may be too expensive.

- FW-AL just requires $\mathrm{LMO}_{\mathcal{C}_1}(\boldsymbol{d})$ and $\mathrm{LMO}_{\mathcal{C}_2}(\boldsymbol{d})$.

Simultaneously sparse and low rank matrix recovery

Proposed by Richard et al. [2012]:

$$\min_{S \succeq 0, \|S\|_1 \leq \beta_1, \|S\|_* \leq \beta_2} \|S - \hat{\Sigma}\|_2^2 \,.$$

- Sparcity constraint: $\mathcal{C}_1 := \{S \succeq 0, \, \|S\|_1 \leq \beta_1\}$,

$$\mathrm{LMO}_{\mathcal{C}_1}(D) = \text{Largest coefficient of the matrix: } O(d^2)$$

- Low rank constraint: $\mathcal{C}_2 := \{S \succeq 0, \, \|S\|_* \leq \beta_2\}$.

$$\mathrm{LMO}_{\mathcal{C}_2}(D) = \text{Largest eigenvector: } O(d^2/\sqrt{\epsilon})$$

Simultaneously sparse and low rank matrix recovery

Proposed by Richard et al. [2012]:

$$\min_{S \succeq 0, \|S\|_1 \leq \beta_1, \|S\|_* \leq \beta_2} \|S - \hat{\Sigma}\|_2^2 \, .$$

- Sparcity constraint: $\mathcal{C}_1 := \{S \succeq 0, \, \|S\|_1 \leq \beta_1\}$,

  $$\mathrm{LMO}_{\mathcal{C}_1}(D) = \text{Largest coefficient of the matrix: } O(d^2)$$

- Low rank constraint: $\mathcal{C}_2 := \{S \succeq 0, \, \|S\|_* \leq \beta_2\}$.

  $$\mathrm{LMO}_{\mathcal{C}_2}(D) = \text{Largest eigenvector: } O(d^2/\sqrt{\epsilon})$$

Simultaneously sparse and low rank matrix recovery

Proposed by Richard et al. [2012]:

$$\min_{S \succeq 0, \|S\|_1 \leq \beta_1, \|S\|_* \leq \beta_2} \|S - \hat{\Sigma}\|_2^2 \,.$$

- Sparcity constraint: $\mathcal{C}_1 := \{S \succeq 0, \; \|S\|_1 \leq \beta_1\}$,

$$\mathrm{LMO}_{\mathcal{C}_1}(D) = \text{Largest coefficient of the matrix: } O(d^2)$$

- Low rank constraint: $\mathcal{C}_2 := \{S \succeq 0, \; \|S\|_* \leq \beta_2\}$.

$$\mathrm{LMO}_{\mathcal{C}_2}(D) = \text{Largest eigenvector: } O(d^2/\sqrt{\epsilon})$$

Multiple sequence alignment

Proposed by Yen et al. [2016a]:

$$\min_{W \in \mathcal{A} \cap \mathcal{P}} \langle W, D \rangle$$

- W: alignment the sequences. D: cost matrix.
- $\mathcal{A}$: *alignment constraint*. Each alignment with the consensus sequence is valid.
- $\mathcal{P}$: consensus constraint. Alignments consistent between each other.

Multiple sequence alignment

Proposed by Yen et al. [2016a]:

$$\min_{W \in \mathcal{A} \cap \mathcal{P}} \langle W, D \rangle$$

- W: alignment the sequences. D: cost matrix.
- $\mathcal{A}$: *alignment constraint*. Each alignment with the consensus sequence is valid.
- $\mathcal{P}$: consensus constraint. Alignments consistent between each other.

Multiple sequence alignment

Proposed by Yen et al. [2016a]:

$$\min_{W \in \mathcal{A} \cap \mathcal{P}} \langle W, D \rangle$$

- W: alignment the sequences. D: cost matrix.
- $\mathcal{A}$: *alignment constraint.* Each alignment with the consensus sequence is valid.
- $\mathcal{P}$: consensus constraint. Alignments consistent between each other.

Multiple sequence alignment

Proposed by Yen et al. [2016a]:

$$\min_{W \in \mathcal{A} \cap \mathcal{P}} \langle W, D \rangle$$

- W: alignment the sequences. D: cost matrix.
- $\mathcal{A}$: *alignment constraint*. Each alignment with the consensus sequence is valid.
- $\mathcal{P}$: consensus constraint. Alignments consistent between each other.

Structured SVM

Proposed by Yen et al. [2016b]:

dual problem:

$$\min_{\alpha_f \in \Delta^{|\mathcal{Y}_f|}} \frac{1}{2} \sum_{F \in \mathcal{T}} \|A_F \alpha\|_2^2 - \sum_{j \in \mathcal{V}} \delta_j^\top \alpha_j$$

$$\text{s.t.} \quad M_{fi}\, \alpha_f = \alpha_i\,, \quad f \in F,\, F \in \mathcal{T},\, i \in \mathcal{N}(f)\,.$$

- $\mathcal{V}$: Variables. $\mathcal{T}$: Factor templates. $\mathcal{N}(f)$: neighbors of f.
- Consistency constraint: $M_{11}x^{(1)} = \alpha_1, M_{12}x^{(1)} = \alpha_2, \ldots$

Structured SVM

Proposed by Yen et al. [2016b]:

dual problem:

$$\min_{\alpha_f \in \Delta^{|\mathcal{Y}_f|}} \frac{1}{2} \sum_{F \in \mathcal{T}} \|A_F \alpha\|_2^2 - \sum_{j \in \mathcal{V}} \delta_j^\top \alpha_j$$

$$\text{s.t.} \quad M_{fi}\, \alpha_f = \alpha_i\,, \quad f \in F,\, F \in \mathcal{T},\, i \in \mathcal{N}(f)\,.$$

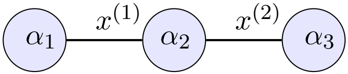
- $\mathcal{V}$: Variables. $\mathcal{T}$: Factor templates. $\mathcal{N}(f)$: neighbors of f.
- Consistency constraint: $M_{11}x^{(1)} = \alpha_1, M_{12}x^{(1)} = \alpha_2, \ldots$

Structured SVM

Proposed by Yen et al. [2016b]:

dual problem:
$$\min_{\alpha_f \in \Delta^{|\mathcal{Y}_f|}} \frac{1}{2} \sum_{F \in \mathcal{T}} \|A_F \alpha\|_2^2 - \sum_{j \in \mathcal{V}} \delta_j^\top \alpha_j$$
$$\text{s.t.} \quad M_{fi}\, \alpha_f = \alpha_i \,, \quad f \in F,\, F \in \mathcal{T},\, i \in \mathcal{N}(f)\,.$$

- $\mathcal{V}$: Variables. $\mathcal{T}$: Factor templates. $\mathcal{N}(f)$: neighbors of f.
- Consistency constraint: $M_{11}x^{(1)} = \alpha_1, M_{12}x^{(1)} = \alpha_2, \ldots$



General Formulation

$$\underset{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(k)}}{\text{minimize}}\; f(\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(k)})\;,$$
$$\boldsymbol{x}^{(k)} \in \mathcal{C}_k,\; k \in [K],\; \sum_{k=1}^{K} A_k \boldsymbol{x}^{(k)} = 0\,.$$

- f is convex and smooth (gradient Lipschitz).
- $\mathcal{C}_k$, $k \in \{1,\dots,K\}$ are convex compact.

Augmented Lagrangian Method

- Augmented Lagrangian trick to get rid of $\sum_{k=1}^K A_k \boldsymbol{x}^{(k)} = 0$.
- M s.t. $M\boldsymbol{x} = 0 \Leftrightarrow \sum_{k=1}^K A_k \boldsymbol{x}^{(k)} = 0$ and the functions,

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) := f(\boldsymbol{x}) + \langle \boldsymbol{y}, M\boldsymbol{x} \rangle + \frac{\lambda}{2}\|M\boldsymbol{x}\|^2 .$$

$$p(\boldsymbol{x}) := \max_{\boldsymbol{y} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} f(\boldsymbol{x}) & \text{if} \quad M\boldsymbol{x} = 0\,, \\ +\infty & \text{otherwise.} \end{cases}$$

- **Augmented Lagrangian formulation** of our problem,

$$\underset{\boldsymbol{x}}{\text{minimize}} \max_{\boldsymbol{y} \in \mathbb{R}^d} \; \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$$
$$\text{s.t.} \quad \boldsymbol{x} \in \mathcal{X} := \times_{k=1}^K \mathcal{C}_k \;.$$

Augmented Lagrangian Method

- Augmented Lagrangian trick to get rid of $\sum_{k=1}^K A_k \boldsymbol{x}^{(k)} = 0$.
- M s.t. $M\boldsymbol{x} = 0 \Leftrightarrow \sum_{k=1}^K A_k \boldsymbol{x}^{(k)} = 0$ and the functions,

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) := f(\boldsymbol{x}) + \langle \boldsymbol{y}, M\boldsymbol{x} \rangle + \frac{\lambda}{2}\|M\boldsymbol{x}\|^2.$$

$$p(\boldsymbol{x}) := \max_{\boldsymbol{y} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} f(\boldsymbol{x}) & \text{if} \quad M\boldsymbol{x} = 0\,, \\ +\infty & \text{otherwise.} \end{cases}$$

- **Augmented Lagrangian formulation** of our problem,

$$\underset{\boldsymbol{x}}{\text{minimize}} \max_{\boldsymbol{y} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$$

$$\text{s.t.} \quad \boldsymbol{x} \in \mathcal{X} := \times_{k=1}^K \mathcal{C}_k\,.$$

Augmented Lagrangian Method

- Augmented Lagrangian trick to get rid of $\sum_{k=1}^{K} A_k \boldsymbol{x}^{(k)} = 0$.
- M s.t. $M\boldsymbol{x} = 0 \Leftrightarrow \sum_{k=1}^{K} A_k \boldsymbol{x}^{(k)} = 0$ and the functions,

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) := f(\boldsymbol{x}) + \langle \boldsymbol{y}, M\boldsymbol{x} \rangle + \tfrac{\lambda}{2}\|M\boldsymbol{x}\|^2 .$$

$$p(\boldsymbol{x}) := \max_{\boldsymbol{y} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} f(\boldsymbol{x}) & \text{if} \quad M\boldsymbol{x} = 0\,, \\ +\infty & \text{otherwise.} \end{cases}$$

- **Augmented Lagrangian formulation** of our problem,

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} \max_{\boldsymbol{y} \in \mathbb{R}^d} \; \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) \\ &\text{s.t.} \quad \boldsymbol{x} \in \mathcal{X} := \times_{k=1}^{K} \mathcal{C}_k \; . \end{aligned}$$

FW-AL algorithm

$$\underset{\boldsymbol{x}}{\text{minimize}} \max_{\boldsymbol{y} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$$
$$\text{s.t.} \quad \boldsymbol{x} \in \mathcal{X} := \times_{k=1}^K \mathcal{C}_k \ .$$

- Standard AL method:

$$\begin{cases} \boldsymbol{x}_{t+1} = \underset{\boldsymbol{x} \in \mathcal{X}}{\arg\min}\, \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}_t) & \textit{(argmin step)} \ , \\ \boldsymbol{y}_{t+1} = \boldsymbol{y}_t + \eta_t M \boldsymbol{x}_{t+1} & \textit{(Gradient ascent step)} \ . \end{cases}$$

- Replace arg min steps by FW steps. FW-AL:

$$\begin{cases} \boldsymbol{x}_{t+1} = \mathcal{FW}(\boldsymbol{x}_t; \mathcal{L}(\cdot, \boldsymbol{y}_t)) & \textit{(Frank-Wolfe step)} \ , \\ \boldsymbol{y}_{t+1} = \boldsymbol{y}_t + \eta_t M \boldsymbol{x}_{t+1} & \textit{(Gradient ascent step)} \ . \end{cases}$$

FW-AL algorithm

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} \max_{\boldsymbol{y} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) \\ &\text{s.t.} \quad \boldsymbol{x} \in \mathcal{X} := \times_{k=1}^{K} \mathcal{C}_k \,. \end{aligned}$$

- Standard AL method:

$$\begin{cases} \boldsymbol{x}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathcal{X}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}_t) & \textit{(argmin step)}\,, \\ \boldsymbol{y}_{t+1} = \boldsymbol{y}_t + \eta_t M \boldsymbol{x}_{t+1} & \textit{(Gradient ascent step)}\,. \end{cases}$$

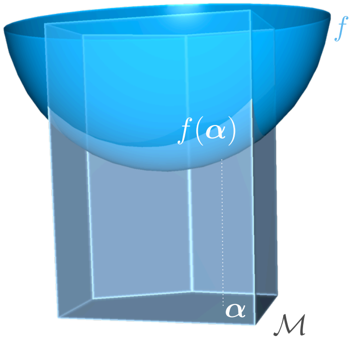
- Replace arg min steps by FW steps. FW-AL:

$$\begin{cases} \boldsymbol{x}_{t+1} = \mathcal{FW}(\boldsymbol{x}_t; \mathcal{L}(\cdot, \boldsymbol{y}_t)) & \textit{(Frank-Wolfe step)}\,, \\ \boldsymbol{y}_{t+1} = \boldsymbol{y}_t + \eta_t M \boldsymbol{x}_{t+1} & \textit{(Gradient ascent step)}\,. \end{cases}$$

The FW algorithm

Algorithm 1 One Frank-Wolfe step

1: Let $\boldsymbol{x}^{(t)} \in \mathcal{M}$
2: Compute $\boldsymbol{r}^{(t)} = \nabla f(\boldsymbol{x}^{(t)})$
3: Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{s} \in \mathcal{C}}{\operatorname{argmin}} \left\langle \boldsymbol{s}, \boldsymbol{r}^{(t)} \right\rangle$
4: Compute $g_t := \left\langle \boldsymbol{x}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
5: **if** $g_t \leq \epsilon$ **then return** $\boldsymbol{x}^{(t)}$
6: Let $\gamma = \frac{2}{2+t}$ (or do line-search)
7: Update $\boldsymbol{x}^{(t+1)} := (1-\gamma)\boldsymbol{x}^{(t)} + \gamma \boldsymbol{s}^{(t)}$



The FW algorithm

Algorithm 2 One Frank-Wolfe step

Let $\boldsymbol{x}^{(t)} \in \mathcal{M}$

Compute $\boldsymbol{r}^{(t)} = \nabla f(\boldsymbol{x}^{(t)})$

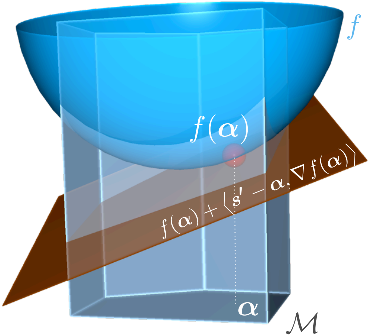
Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{s} \in \mathcal{C}}{\operatorname{argmin}} \left\langle \boldsymbol{s}, \boldsymbol{r}^{(t)} \right\rangle$

Compute $g_t := \left\langle \boldsymbol{x}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$

if $g_t \leq \epsilon$ **then return** $\boldsymbol{x}^{(t)}$

Let $\gamma = \frac{2}{2+t}$ (or do line-search)

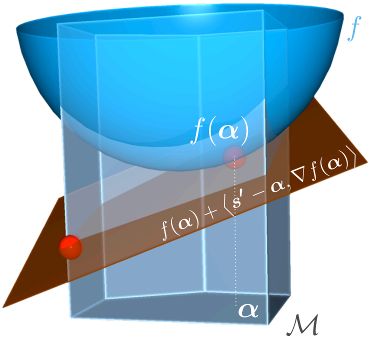
Update $\boldsymbol{x}^{(t+1)} := (1-\gamma)\boldsymbol{x}^{(t)} + \gamma \boldsymbol{s}^{(t)}$



The FW algorithm

Algorithm 3 One Frank-Wolfe step

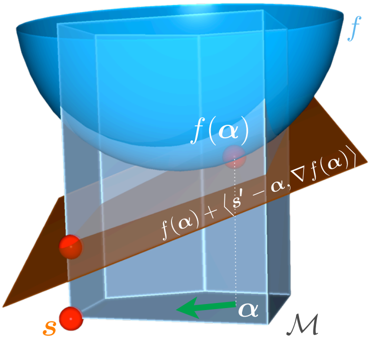
1: Let $\boldsymbol{x}^{(t)} \in \mathcal{M}$
2: Compute $\boldsymbol{r}^{(t)} = \nabla f(\boldsymbol{x}^{(t)})$
3: Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{s}\in\mathcal{C}}{\text{argmin}} \left\langle \boldsymbol{s}, \boldsymbol{r}^{(t)} \right\rangle$
4: Compute $g_t := \left\langle \boldsymbol{x}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
5: **if** $g_t \le \epsilon$ **then return** $\boldsymbol{x}^{(t)}$
6: Let $\gamma = \frac{2}{2+t}$ (or do line-search)
7: Update $\boldsymbol{x}^{(t+1)} := (1-\gamma)\boldsymbol{x}^{(t)} + \gamma \boldsymbol{s}^{(t)}$



The FW algorithm

Algorithm 4 One Frank-Wolfe step

1: Let $\boldsymbol{x}^{(t)} \in \mathcal{M}$
2: Compute $\boldsymbol{r}^{(t)} = \nabla f(\boldsymbol{x}^{(t)})$
3: Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{s} \in \mathcal{C}}{\operatorname{argmin}} \langle \boldsymbol{s}, \boldsymbol{r}^{(t)} \rangle$
4: Compute $g_t := \langle \boldsymbol{x}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \rangle$
5: **if** $g_t \leq \epsilon$ **then return** $\boldsymbol{x}^{(t)}$
6: Let $\gamma = \frac{2}{2+t}$ (or do line-search)
7: Update $\boldsymbol{x}^{(t+1)} := (1-\gamma)\boldsymbol{x}^{(t)} + \gamma \boldsymbol{s}^{(t)}$



Related work: GDMM

- Replace arg min step by a FW step initially proposed by Yen et al. [2016a] to solve MSA problem.
- Afterwards used for Structured SVM [Yen et al., 2016b] and MAP inference [Huang et al., 2017].
- Restricted to polytopes and simple (linear and quadratic) functions.

Contributions:

- Extension of GDMM for general convex sets. (e.g. Trace norm ball)
- Fix a crucial missing part in the previous proofs.

Related work: GDMM

- Replace arg min step by a FW step initially proposed by Yen et al. [2016a] to solve MSA problem.
- Afterwards used for Structured SVM [Yen et al., 2016b] and MAP inference [Huang et al., 2017].
- Restricted to polytopes and simple (linear and quadratic) functions.

Contributions:

- Extension of GDMM for general convex sets. (e.g. Trace norm ball)
- Fix a crucial missing part in the previous proofs.

Related work: GDMM

- Replace arg min step by a FW step initially proposed by Yen et al. [2016a] to solve MSA problem.
- Afterwards used for Structured SVM [Yen et al., 2016b] and MAP inference [Huang et al., 2017].
- Restricted to polytopes and simple (linear and quadratic) functions.

Contributions:

- Extension of GDMM for general convex sets. (e.g. Trace norm ball)
- Fix a crucial missing part in the previous proofs.

Related work: GDMM

- Replace arg min step by a FW step initially proposed by Yen et al. [2016a] to solve MSA problem.
- Afterwards used for Structured SVM [Yen et al., 2016b] and MAP inference [Huang et al., 2017].
- Restricted to polytopes and simple (linear and quadratic) functions.

Contributions:

- Extension of GDMM for general convex sets. (e.g. Trace norm ball)
- Fix a crucial missing part in the previous proofs.

Theoretical contribution

Additional assumption:

Slater's condition: $\exists\, \boldsymbol{x}^{(k)} \in \mathrm{relint}(\mathcal{C}_k),\, k \in [K]$ s.t. $\sum_{k=1}^{K} A_k \boldsymbol{x}^{(k)} = 0\,.$

New lemma:

Let d be the augmented dual function,

$$d(\boldsymbol{y}) := \min_{x \in \mathcal{X}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})\,.$$

There exist a constant $\alpha > 0$ such that close enough to $\mathcal{Y}^*$,

$$d^* - d(\boldsymbol{y}) \geq \alpha \mathrm{dist}(\boldsymbol{y}, \mathcal{Y}^*)^2.$$

Theoretical contribution

Additional assumption:

Slater's condition: $\exists\, \boldsymbol{x}^{(k)} \in \mathrm{relint}(\mathcal{C}_k),\ k \in [K]$ s.t. $\displaystyle\sum_{k=1}^{K} A_k \boldsymbol{x}^{(k)} = 0\,.$

New lemma:

Let d be the augmented dual function,

$$d(\boldsymbol{y}) := \min_{\boldsymbol{x}\in\mathcal{X}} \mathcal{L}(\boldsymbol{x},\boldsymbol{y})\,.$$

There exist a constant $\alpha > 0$ such that close enough to $\mathcal{Y}^*$,

$$d^* - d(\boldsymbol{y}) \geq \alpha \mathrm{dist}(\boldsymbol{y},\mathcal{Y}^*)^2.$$

Convergence results

- **For general convex sets:**
  With decreasing step size $\eta_t := O\left(\frac{1}{t+1}\right)$,

$$\text{subopt: } \Delta_t \leq \frac{O(1)}{t} \,, \quad \text{feasibility: } \min_{t_0 \leq s \leq t} \|M\boldsymbol{x}_s\|^2 \leq \frac{O(1)}{t} \,.$$

- **For $\mathcal{X}$ a polytope:**
  With small enough constant step size η_t:

$$\Delta_t \leq \frac{\Delta_{t_0}}{(1+\rho)^{t-t_0}} \,, \quad \|M\boldsymbol{x}_{t+1}\|^2 \leq \frac{O(1)}{(1+\rho)^{t-t_0}} \,.$$

  Only holds for generalized strongly convex function and uses a variant of FW with away-step.
- Standard splitting algorithms have faster rate **per iteration** in practice.
- Advantage only comes from the **cheaper oracle** !

Convergence results

- **For general convex sets:**
  With decreasing step size $\eta_t := O\left(\frac{1}{t+1}\right)$,

$$\text{subopt: } \Delta_t \leq \frac{O(1)}{t}\,, \quad \text{feasibility: } \min_{t_0 \leq s \leq t} \|M\boldsymbol{x}_s\|^2 \leq \frac{O(1)}{t}\,.$$

- **For $\mathcal{X}$ a polytope:**
  With small enough constant step size η_t:

$$\Delta_t \leq \frac{\Delta_{t_0}}{(1+\rho)^{t-t_0}}\,, \quad \|M\boldsymbol{x}_{t+1}\|^2 \leq \frac{O(1)}{(1+\rho)^{t-t_0}}\,.$$

  Only holds for generalized strongly convex function and uses a variant of FW with away-step.

- Standard splitting algorithms have faster rate **per iteration** in practice.
- Advantage only comes from the **cheaper oracle** !

Experiments

Simultaneously sparse and low rank matrix recovery:

$$\min_{S \succeq 0, \|S\|_1 \leq \beta_1, \|S\|_* \leq \beta_2} \|S - \hat{\Sigma}\|_2^2 \, .$$

- Sparcity constraint: $\mathcal{C}_1 := \{S \succeq 0, \, \|S\|_1 \leq \beta_1\}$,

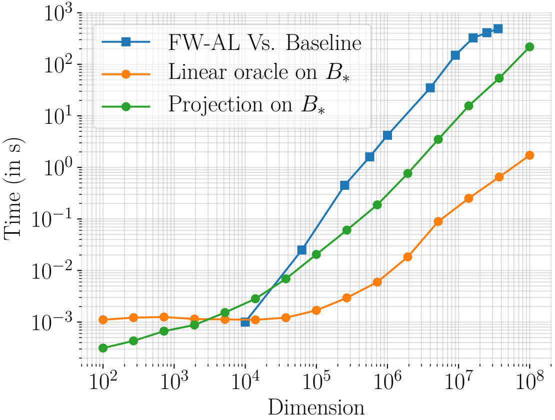
$$\mathrm{LMO}_{\mathcal{C}_1}(D) = \text{Largest coefficient of the matrix: } O(d^2)$$

- Low rank constraint: $\mathcal{C}_2 := \{S \succeq 0, \, \|S\|_* \leq \beta_2\}$.

$$\mathrm{LMO}_{\mathcal{C}_2}(D) = \text{Largest eigenvector: } O(d^2/\sqrt{\epsilon})$$

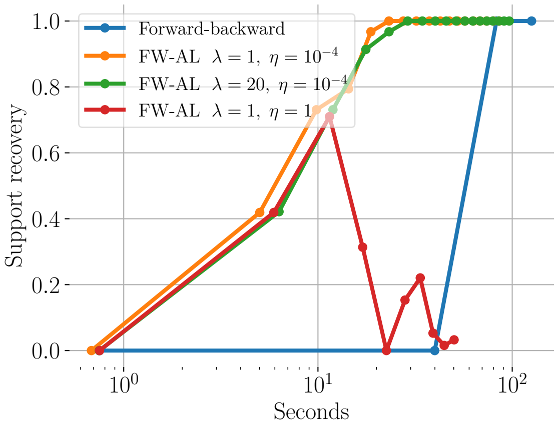
Experiments

LMO vs. projection for trace norm ball:



Experiments

Support recovered by FW-AL and the generalized forward backward algorithm as a function of time:



Conclusion

Task: Minimize a function over an *intersection* of convex sets.

Problem:

- Projections or linear minimization oracle (LMO) over the intersection is **expensive**.
- Projection onto each individual set is **expensive**.

Our solution:

- Requires **linear minimization oracles** over **individual constraints**.
- Based on the **Augmented Lagrangian Method**.

Contributions:

- Extension of GDMM for general convex sets.
- Fix a missing part of the previous proofs.

Conclusion

Task: Minimize a function over an *intersection* of convex sets.
Problem:

- Projections or linear minimization oracle (LMO) over the intersection is **expensive**.
- Projection onto each individual set is **expensive**.

Our solution:

- Requires **linear minimization oracles** over **individual constraints**.
- Based on the **Augmented Lagrangian Method**.

Contributions:

- Extension of GDMM for general convex sets.
- Fix a missing part of the previous proofs.

Conclusion

Task: Minimize a function over an *intersection* of convex sets.
Problem:

- Projections or linear minimization oracle (LMO) over the intersection is **expensive**.
- Projection onto each individual set is **expensive**.

Our solution:

- Requires **linear minimization oracles** over **individual constraints**.
- Based on the **Augmented Lagrangian Method**.

Contributions:

- Extension of GDMM for general convex sets.
- Fix a missing part of the previous proofs.

Conclusion

Task: Minimize a function over an *intersection* of convex sets.
Problem:

- Projections or linear minimization oracle (LMO) over the intersection is **expensive**.
- Projection onto each individual set is **expensive**.

Our solution:

- Requires **linear minimization oracles** over **individual constraints**.
- Based on the **Augmented Lagrangian Method**.

Contributions:

- Extension of GDMM for general convex sets.
- Fix a missing part of the previous proofs.

Conclusion

Task: Minimize a function over an *intersection* of convex sets.
Problem:

- Projections or linear minimization oracle (LMO) over the intersection is **expensive**.
- Projection onto each individual set is **expensive**.

Our solution:

- Requires **linear minimization oracles** over **individual constraints**.
- Based on the **Augmented Lagrangian Method**.

Contributions:

- Extension of GDMM for general convex sets.
- Fix a missing part of the previous proofs.

Conclusion

Task: Minimize a function over an *intersection* of convex sets.
Problem:

- Projections or linear minimization oracle (LMO) over the intersection is **expensive**.
- Projection onto each individual set is **expensive**.

Our solution:

- Requires **linear minimization oracles** over **individual constraints**.
- Based on the **Augmented Lagrangian Method**.

Contributions:

- Extension of GDMM for general convex sets.
- Fix a missing part of the previous proofs.

Conclusion

Task: Minimize a function over an *intersection* of convex sets.
Problem:

- Projections or linear minimization oracle (LMO) over the intersection is **expensive**.
- Projection onto each individual set is **expensive**.

Our solution:

- Requires **linear minimization oracles** over **individual constraints**.
- Based on the **Augmented Lagrangian Method**.

Contributions:

- Extension of GDMM for general convex sets.
- Fix a missing part of the previous proofs.

Thank You !

Slides available on `www.di.ens.fr/~gidel` and `www-ens.iro.umontreal.ca/~gidelgau`.

Bibliography

J.-B. Alayrac, P. Bojanowski, N. Agrawal, I. Laptev, J. Sivic, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016.

X. Evangelopoulos, A. J. Brockmeier, T. Mu, and J. Y. Goulermas. A graduated non-convexity relaxation for large scale seriation. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 462–470. SIAM, 2017.

X. Huang, I. E.-H. Yen, R. Zhang, Q. Huang, P. Ravikumar, and I. Dhillon. Greedy direction method of multiplier for MAP inference of large output domain. In *AISTATS*, 2017.

G. Lancia and P. Serafini. *Compact Extended Linear Programming Models*. Springer, 2018.

E. Richard, P.-A. Savalle, and N. Vayatis. Estimation of simultaneously sparse and low rank matrices. In *ICML*, 2012.

I. Yen, X. Huang, K. Zhong, R. Zhang, P. Ravikumar, and I. Dhillon. Dual decomposed learning with factorwise oracle for structural SVM with large output domain. In *NIPS*, 2016b.

I. E.-H. Yen, X. Lin, J. Zhang, P. Ravikumar, and I. Dhillon. A convex atomic-norm approach to multiple sequence alignment and motif discovery. In *ICML*, 2016a.