Frank-Wolfe Algorithms for Saddle Point problems

author: Gauthier Gidel,
Supervisors: Simon Lacoste-Julien & Tony Jebara
INRIA Paris, Sierra Team & Columbia University

September 15^{th} 2016

Overview

- Machine Learning needs to tackle complicated optimization problems $\Rightarrow$ ML needs optimization.
- Frank-Wolfe algorithm (FW) gained in popularity in the last couple of years.
- It is a convex optimization algorithm solving constrained problems.
- We tried to extend FW to saddle point optimization which is non trivial (we partially answered a 30 years old conjecture).

Motivations: Games

Zero-sum games with two players:

- Player 1 has actions $\{1, \dots, I\}$ available.
- Player 2 has actions $\{1, \dots, J\}$ available.
- If action i and action j, implies a reward M_{ij} for Player 1
- Two players play randomly, $\mathbf{x} \in \Delta(|I|), \mathbf{y} \in \Delta(|J|)$,

$$\mathbb{E}[M_{ij}] = \mathbf{x}^\top M \mathbf{y}$$

Nash equilibrium: $(\mathbf{x}^*, \mathbf{y}^*) \in \mathcal{X} \times \mathcal{Y}$,

$$\forall (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y} \qquad (\mathbf{x}^*)^\top M \mathbf{y} \leq (\mathbf{x}^*)^\top M \mathbf{y}^* \leq \mathbf{x}^\top M \mathbf{y}^*$$

Saddle point setting

Let $\mathcal{L} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $\mathcal{X}$ and $\mathcal{Y}$ are convex and compact.

- Intuition from two players games:
  - $\mathcal{L}$ is a *score* function.
  - P1 chooses action in $\mathcal{X}$ and want to minimize the score.
  - P2 chooses action in $\mathcal{Y}$ and want to maximize the score.
  - The *saddle point* is the couple of best choice for each player.
- $\mathcal{L}$ is said to be *convex-concave* if:
  1. $\forall\, \mathbf{y} \in \mathcal{Y}$, $\mathbf{x} \mapsto \mathcal{L}(\mathbf{x}, \mathbf{y})$ is convex.
  2. $\forall\, \mathbf{x} \in \mathcal{X}$, $\mathbf{y} \mapsto \mathcal{L}(\mathbf{x}, \mathbf{y})$ is concave.
- A *saddle point* is a couple $(\mathbf{x}^*, \mathbf{y}^*)$ such that, $\forall (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$,

$$\mathcal{L}(\mathbf{x}^*, \mathbf{y}) \leq \mathcal{L}(\mathbf{x}^*, \mathbf{y}^*) \leq \mathcal{L}(\mathbf{x}, \mathbf{y}^*)$$

Motivations: mores applications

Robust learning:[1] We want to learn

$$\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell\left(f_\theta(x_i), y_i\right) + \lambda \Omega(\theta) \tag{1}$$

with an uncertainty regarding the data:

$$\min_{\theta \in \Theta} \max_{w \in \Delta_n} \sum_{i=1}^{n} \omega_i \ell\left(f_\theta(x_i), y_i\right) + \lambda \Omega(\theta) \tag{2}$$

[1] Junfeng Wen, Chun-Nam Yu, and Russell Greiner. "Robust Learning under Uncertain Test Distributions: Relating Covariate Shift to Model Misspecification." In: *ICML*. 2014, pp. 631–639.

Standard approaches in literature

The standard algorithm to solve Saddle point optimization is the projected gradient algorithm.

$$\mathbf{x}^{(t+1)} = P_{\mathcal{X}}(\mathbf{x}^{(t)} - \eta \nabla_x \mathcal{L}(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}))$$
$$\mathbf{y}^{(t+1)} = P_{\mathcal{Y}}(\mathbf{y}^{(t)} + \eta \nabla_y \mathcal{L}(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}))$$

When the gradient is uniformly bounded,

$$\frac{1}{T} \sum_{t=1}^{T} \left(\mathbf{x}^{(t)}, \mathbf{y}^{(t)} \right) \underset{T \to \infty}{\longrightarrow} (\mathbf{x}^*, \mathbf{y}^*) \tag{3}$$

The FW algorithm

Initialize $\mathbf{x}^{(0)}$.

For $t = 0, \ldots, T$ do

- Compute:

  $\mathbf{s}^{(t)} := \operatorname*{argmin}_{\mathbf{s} \in \mathcal{X}} \left\langle \mathbf{s}, \nabla f(\mathbf{x}^{(t)}) \right\rangle.$

- Let $\gamma_t = \frac{2}{2+t}$.
- Update:

  $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \gamma_t(\mathbf{s}^{(t)} - \mathbf{x}^{(t)})$

end

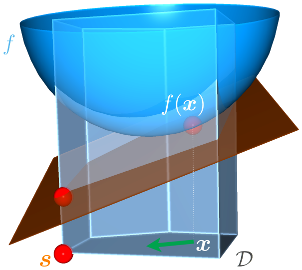

Figure: One step of the FW algorithm

SPFW

Then a Saddle point version of Frank Wolfe algorithm is

- Let $\mathbf{z}^{(0)} = (\mathbf{x}^{(0)}, \mathbf{y}^{(0)}) \in \mathcal{X} \times \mathcal{Y}$
- For $t = 0 \ldots T$
  - Compute $G = \begin{pmatrix} \nabla_x \mathcal{L}(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}) \\ -\nabla_y \mathcal{L}(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}) \end{pmatrix}$
  - Compute $\mathbf{s}^{(t)} := \underset{\mathbf{s} \in \mathcal{X} \times \mathcal{Y}}{\operatorname{argmin}} \langle \mathbf{s}, G \rangle$
  - Let $\gamma_t = \frac{2}{2+t}$
  - Update $\mathbf{z}^{(t+1)} := (1 - \gamma_t)\mathbf{z}^{(t)} + \gamma_t \mathbf{s}^{(t)}$
- **return** $(\mathbf{x}^{(T)}, \mathbf{y}^{(T)})$

Advantages of SP-FW

Why would we use SP-FW ?

- Only a LMO (linear oracle).
- Gap certificate for free.
- Simplicity of implementation.
- Universal step size $\frac{2}{2+k}$, adaptive step size $\frac{g_t}{2C_{\mathcal{L}}}, \ldots$
- *Sparsity* of the solution.
- Lots of improvement easily available. Block-coordinate, Away Step...

When the constraint set is a "complicated" polytope the projection can be super hard whereas the LMO might be tractable.

Problems with Hard projection

The structured SVM:

$$\min_{\omega} \lambda\Omega(\omega) + \frac{1}{n}\sum_{i=1}^{n} \tilde{H}_i(\omega)$$

where $\tilde{H}_i(\omega) = \max_{y \in \mathcal{Y}_i} L_i(y) - \langle \omega, \phi_i(y) \rangle$ is the structured hinge loss. Then we can rewrite the problem as

$$\min_{\Omega(\omega) \leq R} \frac{1}{n}\sum_{i=1}^{n} \left(\max_{\mathbf{y}_i \in \mathcal{Y}_i} \; L_i^\top \mathbf{y}_i - \omega^\top M_i \mathbf{y}_i \right)$$

but as the function is bilinear

$$\min_{\Omega(\omega) \leq \beta} \max_{\alpha \in \Delta(|\mathcal{Y}|)} \; b^T \alpha - \omega^T M \alpha$$

If $\Omega(\cdot)$ is a group lasso norm with overlapping projection is hard. Projecting on $\mathcal{Y}$ is intractable.

Problems with hard projection

University game:

1. Game between two universities (A and B).
2. Admitting d students and have to assign pairs of students into dorms.
3. The game has a payoff matrix M belonging to $\mathbb{R}^{(d(d-1)/2)^2}$.
4. $M_{ij,kl}$ is the expected tuition that B gets (or A gives up) if A pairs student i with j and B pairs student k with l.
5. Here the actions are both in the *marginal polytope* of all perfect *unipartite matchings*.

Hard to project on this polytope whereas the LMO can be solved efficiently with the blossom algorithm[2].

[2] J. Edmonds. "Paths, trees and flowers". In: *Canadian Journal of Mathematics* (1965).

Our contributions

Theoretical contributions:

- Introduced a SP extension of FW with *away step* and proved its convergence over a polytope under some conditions (strong convexity of the function big enough). Partially answering a **30 years old conjecture**[3].
- With step size $\gamma_t \sim g_t$

$$h_t = O\left((1-\rho)^{t/3}\right). \tag{4}$$

[3] Janice H Hammond. "Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms". PhD thesis. Massachusetts Institute of Technology, 1984.

Toy experiments

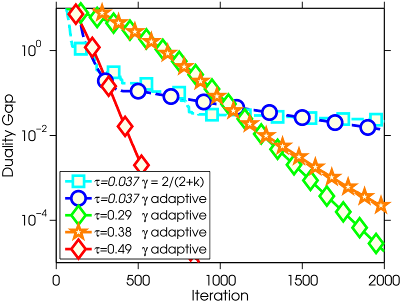

Figure: SP-AFW on a toy example $d = 30$

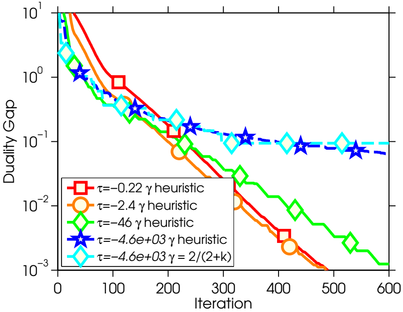

Figure: SP-AFW on a toy example $d = 30$ with heuristic step-size

Experiments

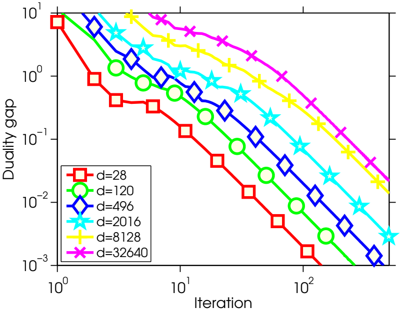

Figure: SP-FW on the University game.

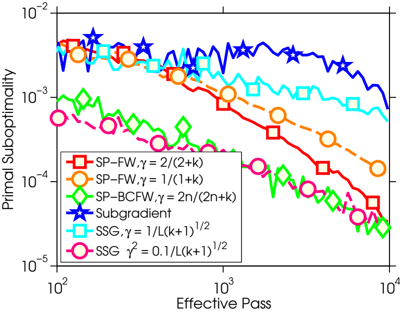

Figure: Structural SVM with OCR dataset (highly regularized).

Conclusion

- There already exist a lot a saddle point problem in the machine learning literature and they are most of the time solved by a trick.
- There exist a few number of algorithm to solve SP problems directly ! (and they are not well known)
- SP-FW work on SPs and is the only algorithm existing able to solve some of these problem.

Thank You !