Review of Frank Wolfe and its variants

William Saint-Arnaud WILLIAM.ST-ARNAUD@UMONTREAL.CA
Elyes Lamouchi ELYESLAMOUCHI@GMAIL.COM
Frederic Boileau FREDERIC.BOILEAU@UMONTREAL.CA

Abstract

Due to the combinatorial nature of multilabel outputs, predicting structured data typically comes with an exponentially large number of constraints, which makes the problem inefficient or intractable in practice. There has been a lot of research focused on providing a solution to that issue. In the structured SVM setting, conditional gradient methods a.k.a Frank-Wolfe type algorithms have become a method of choice.

This paper's aim is to synthesize the recent advances, starting from the classical F-W to the more sophisticated variants, while motivating this with the problems each variant addresses. Finally, we will discuss the pitfalls of some variants and their intrinsic trade-offs. We will then evaluate the performance of the methods proposed to see how reasonable are the assumptions (providing theoretical guarantees) on synthetic data, and to get an idea whether each variant's trade-off is worth it.

1. Structured SVM context

Given a training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ where $y \in \mathcal{Y}$ is a multi-label output, and a feature map $\phi : \mathcal{X} \times \mathcal{Y} \longrightarrow \mathbb{R}$, which encodes a similarity measure between $\mathcal{X}$ and $\mathcal{Y}$, such that if y_i is the ground truth (target) for an input x_i, then

$$\forall y \in \mathcal{Y} \backslash \{y_i\} \quad \textit{we have} \quad \psi_i(y) = \phi(x_i, y_i) - \phi(x_i, y) > 0$$

The aim is to construct an accurate linear classifyer, $h_w(x) = \underset{y \in \mathcal{Y}(x)}{\textit{argmax}} \langle w, \phi(x, y) \rangle$.
To learn w, consider the task loss $L : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}_+$, where $L(y, y\prime) = 0 \Longleftrightarrow y = y\prime$.

$$\begin{aligned} &\underset{w,\xi}{max} \quad \frac{\lambda}{2}||w||^2 + \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i \\ &\textit{s.t.} \quad \langle w, \psi_i(y) \rangle \geq L(y_i, y) - \varepsilon_i, \quad \forall i, \forall y \in \mathcal{Y}(x) = \mathcal{Y}_i \end{aligned}$$

Problems: (1) The zero-one loss is not differentiable and (2) we have an exponential number of constraints.
Solutions: (1) Minimizing an upper bound to the task loss gives us a worst case guarantee.

Consider the **max oracle**, $\tilde{H} = \underset{y \in \mathcal{Y}_i}{max} \underbrace{L_i(y) - \langle w, \psi_i(y) \rangle}_{=H_i(y,w) \quad \textit{the hinge loss}}$.
(2) The exponential number of constraints are replaced by n piecewise linear ones.

Proposition. The max oracle is a convex upper bound to the task loss.
Proof. The maximum of two convex (linear) functions is convex, and

$$\begin{aligned} L(y_i, h_w(x_i)) &\leq L(y_i, h_w(x_i)) + \underbrace{\langle w, \psi_i(y) \rangle}_{\geq 0 \textit{ by definition}} \\ &\leq \underset{y \in \mathcal{Y}_i}{max} L_i(y) - \langle w, \psi_i(y) \rangle \end{aligned}$$

Thus learning w amounts to the unconstrained problem,

$$\underset{w}{max} \quad \frac{\lambda}{2}||w||^2 + \frac{1}{n}\sum_{i=1}^{n} \tilde{H}_i(w)$$

2. From classical Frank-Wolfe to more sophisticated variants

Consider the problem of minimizing a convex continuously differentiable objective function f over a compact set $\mathcal{M}$. Due to the exponential number of dual variables in the structured SVM setting, classical algorithms, like projected gradient are intractable.
Stochastic subgradient methods, on the other hand, achieve a linear convergence rate while only requiring a single call to the maximization oracle every step. They are nonetheless very sensitive to the sequence of stepsizes and it is unclear when to terminate the iterations.

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

Algorithm 1 Frank-Wolfe

Let $\alpha \in \mathcal{M}$
for $k = 0$ **to** K **do**
  Compute $s = argmin_{s \in \mathcal{M}} \langle s, \nabla f(\alpha^k) \rangle$
  Let $\gamma = \frac{2}{k+2}$, or optimize γ by line search
  Update $\alpha^{k+1} = (1-\gamma)\alpha^k + \gamma s$
end for

Conditional gradient a.k.a **Frank Wolfe**, addresses this problem by giving an adaptive stepsize $\gamma = \frac{2}{k+2}$ and a duality gap while still retaining a linear rate of convergence.
The Frank Wolfe algorithm goes as follows: We definie the duality gap $g(\alpha^k) = \max_{s \in \mathcal{M}} \langle \alpha^k - s, \nabla f(\alpha^k) \rangle$.
By first order convexity of the objective, we have

$$f(s) \geq f(\alpha^k) + \langle \alpha^k - s, \nabla f(\alpha^k) \rangle$$
$$\implies g(\alpha^k) = -\min_{s \in \mathcal{M}} \langle \alpha^k - s, \nabla f(\alpha^k) \rangle \geq f(\alpha^k) - f^*$$

We can thus see that the duality gap gives us an optimality guarantee.
Note. The main idea here, is that the linear subproblem in Frank-Wolfe and the loss augmented decoding of the structured SVM are equivalent.
Proof of the equivalence. The objective function being differentiable and convex, if we are at a point α such that $f(\alpha)$ is minimized along each coordinate axis, then α is a global minimizer. Therefore,

$$\min_{s \in \mathcal{M}} \langle s, \nabla f(\alpha) \rangle = \sum_i \min_{s_i \in \Delta_{|\mathcal{Y}_i|}} \langle s_i, \nabla_i f(\alpha) \rangle$$

Moreover, with

$$w = A\alpha, A = \Big[\frac{1}{n\lambda}\psi_1(y) ... \frac{1}{n\lambda}\psi_{\sum_i |\mathcal{Y}_i|}(y)\Big]$$
$$\textit{and} \quad b = \Big(\frac{1}{n}L_i(y)\Big)_{i \in [n], y \in \mathcal{Y}_i}$$

The gradient of the dual would be,

$$\nabla f(\alpha) = \nabla\Big[\frac{\lambda}{2}||A\alpha||^2 - b^T\alpha\Big] = \lambda A^T A\alpha - b$$
$$= \lambda A^T w - b = \frac{1}{n}H_i(y, w)$$
$$\max_{y_i \in \mathcal{Y}_i} \tilde{H}_i = -\min_{y_i \in \mathcal{Y}_i} \tilde{H}_i = \min_{y_i \in \mathcal{Y}_i} L_i - \langle w, \psi_i \rangle$$
$$= \min_{s_i \in \Delta_{|\mathcal{Y}_i|}} \langle s_i, \nabla_i f(\alpha) \rangle$$

Thus we can see that, if $n =$ size of the training data, one Frank-Wolfe step is equivalent to n calls to the maximization oracle. Unlike stochastic subgradient and stochastic methods in general, classical Frank-Wolfe requires one call for each training example at each iteration. For large datasets, this can get unpractical.
Hence the stochastic variant of Frank Wolfe, **Block Coordinate Frank Wolfe (BCFW)**.

Algorithm 2 Batch Primal-Dual Frank-Wolfe

Let $\alpha \in \mathcal{M}$
Let $w^0 = 0, l^0 = 0$
for $k = 0, \ldots, K$ **do**
  for $i = 1, \ldots, n$ **do**
    Solve $y_i^* = \max_{y_i \in \mathcal{Y}_i} H_i(y, w^k)$//
    Let $w_s = \sum_{i=1}^n \frac{1}{n\lambda}\psi_i(y_i^*)$, and $l_s = \frac{1}{n}\sum_{i=1}^n L_i(y_i^*)$
    Let $\gamma = \frac{\lambda(w^k - w_s)^T w^k - l^k + l_s}{\lambda||w^k - w_s||^2}$, and clip to $[0, 1]$
    Update $w^{k+1} = (1-\gamma)w^k + \gamma w_s$, and $l^{k+1} = (1-\gamma)l^k + \gamma l_s$
  end for
end for

Algorithm 3 Block-Coordinate Frank-Wolfe

Let $w^0 = w_i^0 = \overline{w}^0 = 0, l^0 = l_i^0 = 0$
for $k = 0 ... K$ **do**
  Pick i at random in $\{1, ..., n\}$
  Solve $y_i^* = \max_{y_i \in \mathcal{Y}_i} H_i(y, w^k)$
  Let $w_s = \frac{1}{n\lambda}\psi_i(y_i^*)$, and $l_s = \frac{1}{n}L_i(y_i^*)$
  Let $\gamma = \frac{\lambda(w_i^k - w_s)^T w^k - l_i^k + l_s}{\lambda||w_i^k - w_s||^2}$, and clip to $[0, 1]$
  Update $w_i^{k+1} = (1-\gamma)w_i^k + \gamma w_s$, and $l_i^{k+1} = (1-\gamma)l_i^k + \gamma l_s$
  Update $w^{k+1} = w^k + w_i^{k+1} - w_i^k$, and $l_i^{k+1} = (1-\gamma)l_i^k + \gamma l_s$
end for

Theorem. Given a convex, differentiable objective $f : \mathcal{M}^1 \times ... \times \mathcal{M}^n \to \mathbb{R}$, where $\forall i \in \{1..n\}$, each factor $\mathcal{M}^i \subseteq \mathbb{R}^n$ is convex and compact, if we are at a point x such that $f(x)$ is minimized along each coordinate axis, then x is a global minimum.

As in coordinate descent, we minimize the objective function one coordinate (block) at a time. At each iteration, BCFW picks the i^{th} block (from n) uniformly at random and updates the i^{th} coordinate of the corresponding weight, by calling the maximization oracle on the chosen block.

Convergence Results

Definition. The curvature constant C_f is given by the maximum relative deviation of the objective function f from its

linear aaporximations, over the domain $\mathcal{M}$,

$$C_f = \sup_{\substack{x,s\in\mathcal{M} \\ \gamma\in[0,1], y=x+\gamma(s-x)}} \frac{2}{\gamma^2}\Big(f(y) - f(x) - \langle y-x, \nabla f(x)\rangle\Big)$$

Intuitively, the curvature constant can be seen as a measure of how flat the objective function is. For example, if the objective is linear, say $f(x) = ax + b$ and $x \in [e, f]$ then $\nabla f(x) = a$ and $C_f = \frac{2}{\gamma^2}\Big(ay + b - ax - b + (-ay + ax)\Big) = 0$.
Moreover $s = \underset{s\in[e,f]}{argmin}\langle s, a\rangle = \frac{e}{a}$. Thus we reach the minimum in one F-W step.
Thus, we can observe that for flatter functions, that is with smaller curvature constants, Frank-Wolfe should converge faster.

Definition. Over each coordinate block $\mathcal{M}^i$, let the curvature be given by,

$$C_f^{(i)} = \sup_{\substack{x\in\mathcal{M}, s_i\in\mathcal{M}^i \\ y=x+\gamma(s_{[i]}-x_{[i]}) \\ \gamma\in[0,1]}} \frac{2}{\gamma^2}\Big(f(y) - f(x) - \langle y_i - x_i, \nabla_i f(x)\rangle\Big)$$

Where $x_{[i]}$ refers to the zero-padding of i^{th} coordinate of x. And let the global *product curvature constant* be,

$$C_f^{\otimes} = \sum_{i=1}^{n} C_f^{(i)}$$

Theorem. For the dual structural SVM objective function over the domain $\mathcal{M} = \Delta_{|\mathcal{Y}_1|} \times ... \times \Delta_{|\mathcal{Y}_n|}$, the total curvature constant $C_f^{\otimes}$, on the product domain $\mathcal{M}$, is upper bounded by,

$$C_f^{\otimes} \geq \frac{4R^2}{\lambda n} \quad \textit{where} \quad R = \max_{i\in[n], y\in\mathcal{Y}_i} ||\psi_i(y)||_2$$

Proof. By second order convexity on f at y, we have

$$\begin{aligned} f(y) &\leq f(x) + \langle y_i - x_i, \nabla_i f(x)\rangle \\ &\quad + (y-x)^T\nabla^2 f(x)(y-x) \\ f(y) - f(x) - \langle y_i - x_i, \nabla_i f(x)\rangle &\\ \leq (y-x)^T\nabla^2 f(x)(y-x) & \end{aligned}$$

$$C_f^{(i)} \leq \sup_{\substack{x\in\mathcal{M}, s_i\in\mathcal{M}^i \\ y=x+\gamma(s_{[i]}-x_{[i]}) \\ \gamma\in[0,1]}} \Big(f(y) - f(x) - \langle y_i - x_i, \nabla_i f(x)\rangle\Big)$$

$$\leq \sup_{\substack{x,y\in\mathcal{M}, (y-x)\in\mathcal{M}^{[i]} \\ z\in[x,y]\subseteq\mathcal{M}}} (y-x)^T\nabla^2 f(z)(y-x)$$

Moreover

$$\sup_{\substack{x,y\in\mathcal{M}, (y-x)\in\mathcal{M}^{[i]} \\ z\in[x,y]\subseteq\mathcal{M}}} (y-x)^T\nabla^2 f(z)(y-x)$$

$$= \lambda \sup_{x,y\in\mathcal{M}, (y-x)\in\mathcal{M}^{[i]}} (A(y-x))^T\nabla^2 f(z)(A(y-x))$$

$$C_f^{(i)} \leq \lambda \sup_{v,w\in A\mathcal{M}^{(i)}} ||v-w||_2^2 \leq \lambda \sup_{v\in A\mathcal{M}^{(i)}} ||2v||_2^2$$

Where $\forall v \in A\mathcal{M}^{(i)}$, v is a convex combination of the feature vectors corresponding to the possible labelings for the i^{th} example of the training data, such that $||v||_2 \leq$ *the longest column of* $A = \frac{1}{n\lambda}R$. Therefore,

$$C_f^{\otimes} = \sum_{i=1}^{n} C_f^{(i)} \leq 4\lambda \sum_{i=1}^{n} \Big(\frac{1}{n\lambda}R\Big)^2 = \frac{4}{n\lambda}R^2$$

Which completes the proof. □

First, we observe that the curvature constant for BCFW is n times smaller than that of batch Frank Wolfe which is $\leq \frac{4}{\lambda}R^2$. Hence the n times faster convergence rate of BCFW.

Definition. Let $||.||$ be a norm on $\mathbb{R}^n$. The associated dual norm, denoted $||.||_*$ is defined as,

$$||z||_* = \underset{z}{sup}\{z^T x| \quad ||x|| \leq 1\}$$

We denote the dual norm of l_p by l_q. For $p = 2$ we have $q = 2$ and for $p = 1$, $q = \infty$.

Definition. A function f has Lipschitz continuous gradient if:

$$\forall x, y \in dom(f), \exists L > 0 \quad \textit{such that} \quad ||\nabla f(x) - \nabla f(y)||^2 \leq L||x-y||^2$$

We say that a function is L-smooth if its gradient is Lipschitz continuous for some $L > 0$.

The Fundamental Descent Lemma. If f is L-smooth then

$$f(y) \leq f(x) + \langle \nabla f(x), y-x\rangle + \frac{L}{2}||y-x||^2$$

Proof. Let g be such that: $g(\alpha) \triangleq f(x + \alpha(y-x)) \implies$
$f(y) - f(x) = g(1) - g(0)$
$= \int_0^1 \frac{dg}{d\alpha}(\alpha)d\alpha = \int_0^1 \frac{df}{d\alpha}(x + \alpha(y-x))d\alpha$
$= \int_0^1 (y-x)\nabla f(x + \alpha(y-x))d\alpha$
$= \int_0^1 (y-x)\Big[\nabla f(x + \alpha(y-x)) - \nabla f(x)\Big]d\alpha + \int_0^1 (y-x)\nabla f(x)d\alpha$
$\leq \Big|\Big| \int_0^1 (y-x)\nabla f(x+\alpha(y-x))d\alpha\Big|\Big| + \int_0^1 (y-x)\nabla f(x)d\alpha$
$\underset{\Delta \text{ inequality}}{\leq} \int_0^1 \Big|\Big|(y-x)\nabla f(x + \alpha(y-x))\Big|\Big|d\alpha + \int_0^1 (y-x)\nabla f(x)d\alpha$
$\underset{\text{Cauchy-Schwartz}}{\leq} \int_0^1 ||y-x||.||\nabla f(x + \alpha(y-x))||d\alpha + (y-x)\Delta f(x)$

$$\underset{\text{Lipschitz continuity of } f}{\leq} ||y-x|| \int_0^1 L\alpha||y-x||d\alpha + (y-x)\Delta f(x)$$
$$= \langle \nabla f(x), y-x \rangle + \frac{L}{2}||y-x||^2 \quad \square.$$

Theorem. If a convex function f on C has Lipschitz gradient, i.e $||\nabla f(x) - \nabla f(y)||_p \leq L_q ||x-y||_p, \quad \forall x, y \in C$, then

$$C_f \leq L_q . \text{diam}_p^2(C)$$

Proof. f has Lipschitz gradient therefore by the fundamental descent lemma we have,

$$f(y) - f(x) - \langle y-x, \nabla f(x) \rangle \leq \frac{L_q}{2}||y-x||_p^2$$
$$C_f \leq \max_{\substack{y=(1-\gamma)x+\gamma s \\ x,s,y \in C}} \frac{2}{\gamma^2}\frac{L_q}{2} \underbrace{||y-x||_p^2}_{=\gamma^2||x-s||_p^2}$$
$$C_f \leq L_q \underbrace{\max_{x,s \in C} ||x-s||_p^2}_{\triangleq diam_p^2(C)} \quad \square.$$

Problem. For $p = q = 2$ we get $diam_2^2(C) = 2n$, and the Lipschitz constant L_q is the largest eigenvalue of the hessian.

$$\lambda A^T A = \frac{1}{n^2 \lambda}\Big(\langle \psi_i(y) - \psi_j(y\prime) \rangle\Big)_{(i,y),(j,y\prime)}$$

And say $\langle \psi_i(y) - \psi_j(y\prime) \rangle \approx 1$ *for a lot of outputs, we get:*

$$\mathbb{1}^T \mathbb{1} \approx \mathbb{1} \underbrace{diam_2(C)}_{=\sqrt{2n}}$$

Hence the largest eigenvalue the hessian, and therefore the Lipschitz constant, can scale with the dimension of $A^T A$, i.e exponentially with the size of the training data, rendering the bound above very loose, and thus of little practical use.

Solution. Taking $p = 1$ and therefore $q = \infty$, we get $Ldiam^2(C) \approx \frac{4}{\lambda} R^2$.

3. Extra-Gradient

3.1. Overview

In a typical setting