Primal-Dual Block Generalized Frank-Wolfe

Qi Lei*, Jiacheng Zhuo*, Constantine Caramanis*, Inderjit S. Dhillon*,†
and Alexandros G. Dimakis*.

* University of Texas at Austin
† Amazon

Problem Setup

Convex-concave saddle point Problem (with constraints):

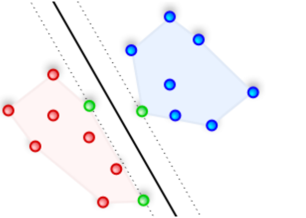
$$\min_{\boldsymbol{x} \in C \subset \mathbb{R}^d} \max_{\boldsymbol{y} \in \mathbb{R}^n} \left\{ L(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x}) + \boldsymbol{y}^\top A\boldsymbol{x} - g(\boldsymbol{y}) \right\}$$

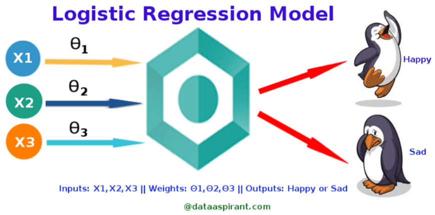
Why is this formulation important?

1. Many machine learning applications

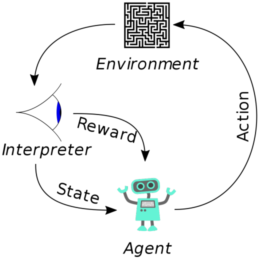
Machine Learning Applications with Convex-Concave Formulations

Empirical Risk Minimization

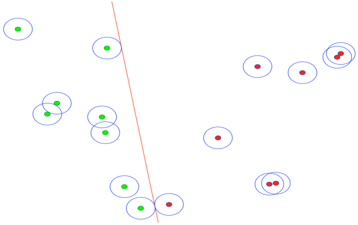




Reinforcement Learning



Robust Optimization



Problem Setup

Convex-Concave Saddle Point Problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \max_{\boldsymbol{y}\in\mathbb{R}^n} \left\{ L(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x}) + \boldsymbol{y}^\top A\boldsymbol{x} - g(\boldsymbol{y}) \right\}$$

Why is this formulation important?

1. Many machine learning applications
2. To exploit special structure induced by the constraints

Observations and Challenges On Frank-Wolfe algorithm

Lessons from simple constrained minimization problems:

Observations.

Frank-Wolfe conducts *partial updates:*

1. For ℓ_1 ball constraint, FW conducts **1-sparse** update
2. For nuclear norm ball constraint, FW conducts **rank-1** update

Challenges to get full benefits from FW and the partial updates.

1. FW yield **sublinear convergence** even for strongly convex problems
2. Even with partial updates, FW requires to compute the full gradient.

(For big data setting, **per iteration complexity is the same with projected gradient descent**.)

Tackle challenge 1: To achieve linear convergence

- Continue to look at simple minimization problems:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d, \|\boldsymbol{x}\|_1\leq\tau} \{f(\boldsymbol{x})\}$$

Method	iteration complexity	# update per iteration
Projected GD	$\kappa \log \frac{1}{e}$	d (feature dimension)
Frank-Wolfe	$\frac{1}{e}$	1
Ours	$\kappa \log \frac{1}{e}$	s (optimal sparsity)

Tackle challenge 1: block Frank-Wolfe

Input: Data matrix $A \in \mathbb{R}^{n \times d}$, label matrix b, iteration T.

Initialize: $x_1 \leftarrow 0$.

for $t = 1, 2, \cdots, T-1$ **do**

$$\text{ProjectedGD: } \Delta x_t \leftarrow \operatorname*{argmin}_{\|\Delta x\|_1 \le \tau} \{\langle \nabla f(x_t), \Delta x \rangle + \frac{\beta}{2}\eta \|\Delta x - x_t\|_2^2\}$$

$$x_{t+1} \leftarrow (1-\eta)x_t + \eta \Delta x_t$$

end for

Output: x_T

Tackle challenge 1: block Frank-Wolfe

Input: Data matrix $A \in \mathbb{R}^{n\times d}$, label matrix b, iteration T.

Initialize: $x_1 \leftarrow 0$.

for $t = 1, 2, \cdots, T-1$ **do**

$$\text{ProjectedGD: } \Delta x_t \leftarrow \underset{\|\Delta x\|_1 \le \tau}{\operatorname{argmin}} \left\{ \langle \nabla f(x_t), \Delta x \rangle + \frac{\beta}{2}\eta \|\Delta x - x_t\|_2^2 \right\}$$

$$\text{FW: } \Delta x_t \leftarrow \underset{\|\Delta x\|_1 \le \tau}{\operatorname{argmin}} \left\{ \langle \nabla f(x_t), \Delta x \rangle \right\}$$

$$x_{t+1} \leftarrow (1-\eta)x_t + \eta \Delta x_t$$

end for

Output: x_T

Tackle challenge 1: block Frank-Wolfe

Input: Data matrix $A \in \mathbb{R}^{n \times d}$, label matrix b, iteration T.

Initialize: $x_1 \leftarrow 0$.

for $t = 1, 2, \cdots, T-1$ **do**

$$\text{ProjectedGD: } \Delta x_t \leftarrow \operatorname*{argmin}_{\|\Delta x\|_1 \leq \tau} \{\langle \nabla f(x_t), \Delta x \rangle + \frac{\beta}{2}\eta \|\Delta x - x_t\|_2^2\}$$

$$\text{FW: } \Delta x_t \leftarrow \operatorname*{argmin}_{\|\Delta x\|_1 \leq \tau} \{\langle \nabla f(x_t), \Delta x \rangle\}$$

$$\text{ours: } \Delta x_t \leftarrow \operatorname*{argmin}_{\|\Delta x\|_1 \leq \tau, \|\Delta x\|_0 \leq s} \{\langle \nabla f(x_t), \Delta x \rangle + \frac{\beta}{2}\eta \|\Delta x - x_t\|_2^2\}$$

$$x_{t+1} \leftarrow (1-\eta)x_t + \eta \Delta x_t$$

end for

Output: x_T

Tackle challenge 2: reduce iteration complexity from partial updates

$$\min_{\boldsymbol{x}\in C\subset\mathbb{R}^d}\max_{\boldsymbol{y}\in\mathbb{R}^n}\left\{L(\boldsymbol{x},\boldsymbol{y}) = f(\boldsymbol{x}) + y^\top A\boldsymbol{x} - g(\boldsymbol{y})\right\}$$

Write $\boldsymbol{w} = A\boldsymbol{x}$ and $\boldsymbol{z} = A^\top\boldsymbol{y}$.

For each iteration,

Operation	cost
Compute full gradient $\partial_{\boldsymbol{x}} L = A^\top\boldsymbol{y} + f'(\boldsymbol{x})$	$\mathcal{O}(nd)$
Conduct BlockFW on $\boldsymbol{x}$ to find s-sparse update $\Delta\boldsymbol{x}$	$\mathcal{O}(d)$
$\boldsymbol{x}^+ \leftarrow (1-\eta)\boldsymbol{x} + \eta\Delta\boldsymbol{x}$	$\mathcal{O}(d)$
Greedy block-k coordinate ascent for $\boldsymbol{y}$	$\mathcal{O}(nd)$

Remark 1: take $k = ns/d$ the iteration complexity is $\mathcal{O}(sn)$.

Remark 2: the advantage comes from the fact that gradient could be maintained with the bilinear form

Tackle challenge 2: reduce iteration complexity from partial updates

$$\min_{\boldsymbol{x}\in C\subset\mathbb{R}^d}\max_{\boldsymbol{y}\in\mathbb{R}^n}\left\{L(\boldsymbol{x},\boldsymbol{y})=f(\boldsymbol{x})+y^\top A\boldsymbol{x}-g(\boldsymbol{y})\right\}$$

Maintain $\boldsymbol{w}=A\boldsymbol{x}$ and $\boldsymbol{z}=A^\top\boldsymbol{y}$.

For each iteration,

Operation	cost
Compute full gradient $\partial_{\boldsymbol{x}}L=\boldsymbol{z}+f'(\boldsymbol{x})$	$\mathcal{O}(d)$
Conduct BlockFW on $\boldsymbol{x}$ to find s-sparse update $\Delta\boldsymbol{x}$	$\mathcal{O}(d)$
$\boldsymbol{x}^+\leftarrow(1-\eta)\boldsymbol{x}+\eta\Delta\boldsymbol{x}$	$\mathcal{O}(d)$
$\boldsymbol{w}^+\leftarrow(1-\eta)\boldsymbol{w}+\eta A\Delta\boldsymbol{x}$	$\mathcal{O}(sn)$
Greedy block-k coordinate ascent for $\boldsymbol{y}$ and $\boldsymbol{z}$	$\mathcal{O}(kd)$

Remark 1: take $k=ns/d$ the iteration complexity is $\mathcal{O}(sn)$.

Remark 2: the advantage comes from the fact that gradient could be maintained with the bilinear form

Time complexity comparisons

Algorithm	Per Iteration Cost	Iteration Complexity
Frank Wolfe	$\mathcal{O}(nd)$	$\mathcal{O}(\frac{1}{\epsilon})$
Accelerated PGD (Nesterov et al. 2013)	$\mathcal{O}(nd)$	$\mathcal{O}(\sqrt{\kappa}\log\frac{1}{\epsilon})$
SVRG (Rie et al. 2013)	$\mathcal{O}(nd)$	$\mathcal{O}((1+\kappa/n)\log\frac{1}{\epsilon})$
SCGS (Lan et al. 2016)	$\mathcal{O}(\kappa^2\frac{\#\text{iter}^3}{\epsilon^2}d)$	$\mathcal{O}(\frac{1}{\epsilon})$
STORC (Hazan et al. 2016)	$\mathcal{O}(\kappa^2 d + nd)$	$\mathcal{O}(\log\frac{1}{\epsilon})$
Primal Dual FW (ours)	$\mathcal{O}(ns)$	$\mathcal{O}((1+\kappa/n)\log\frac{1}{\epsilon})$

Remark 1: s is the sparsity of primal optimal induced by ℓ_1 constraint.
Remark 2: for algorithm and complexity for nuclear norm constraints, refer to our paper to details.

Experiments

Compared methods: (1) Accelerated ProjectedGradient Descent (Acc PG) (2) Frank-Wolfe algorithm (FW) (3) Stochastic Variance ReducedGradient (SVRG) (4) Stochastic Conditional Gradient Sliding (SCGS) and (5) StochasticVariance-Reduced Conditional Gradient Sliding (STORC)