Lecture 18 - BCFW

Thursday, March 14, 2019 13:37

today:
- FW & FCFW on SVMstruct
- BCFW

Continue FW for SVMstruct

FW on dual is equivalent to batch subgradient update on primal with step-size relationship

$\alpha^{(t)} \longrightarrow w^{(t)} = A\alpha^{(t)}$

$\gamma^{(FW)} = \lambda \beta^{(subgradient)}$

recall: subgradient method converges at $O(\frac{1}{t})$

when step-size $\beta_t \leq \frac{1}{\mu} O(\frac{1}{t})$

FW method with fixed step-size schedule

$\gamma_t = \frac{2}{t+2}$

gives $d(\alpha^*) - d(\alpha^{(t)}) \leq \frac{2C_f}{t+2}$

$d(\alpha^{(t)}) \quad p(w^*) = d(\alpha^*) \quad p(w^{(t)})$

dual subopt. — primal subo.

* FW-gap

$$< -\nabla f(\alpha^{(t)}), s^{(t)} - \alpha^{(t)} >$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left[H_i(\hat{y}_i^{(t)}; w^{(t)}) - \sum_{\tilde{y} \in \mathcal{Y}_i} \alpha_i^{(t)}(\tilde{y}) H_i(\tilde{y}; w^{(t)}) \right]$$

$$= p(\underbrace{w(\alpha^{(t)})}_{A\alpha^{(t)}}) - d(\alpha^{(t)})$$ (Lagrangian gap of lecture xx)

here, FW-gap = Lagrangian gap on $(w(\alpha^{(t)}), \alpha^{(t)})$

* recall that FW gap ...

[note: FW gap is not always Lagrangian gap]

* recall, that $\min_{s \le t} \overset{\text{FW gap}}{g_s} \le 3 \cdot \frac{2C_f}{t+2}$

[note: FW gap is ~~not~~ always Lagrangian gap
e.g. CRF objective (dual)]

$\Rightarrow$ guarantees on $p(w(\alpha^{(t)})) - p(w^*)$?

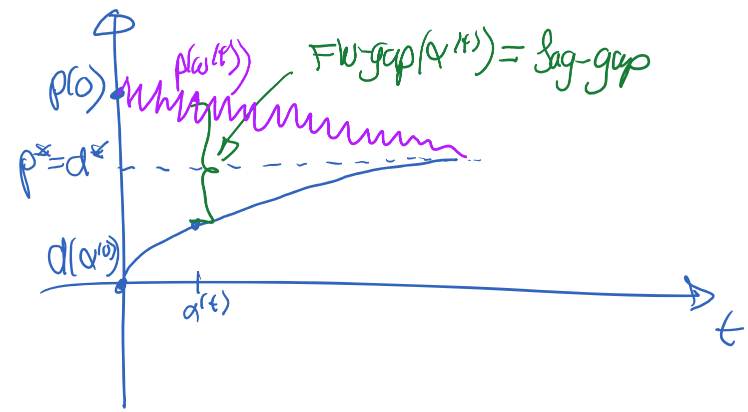
also, $g^{FW}(\hat{x}^{(t)}_{wA}) \le 3 \cdot \frac{2C_f}{t+2}$

weighted average

when $g(\cdot)$ is convex [this is the case when f is quadratic]

$p(0) = \frac{1}{n} \sum_{i=1}^{n} \max_{\tilde{y}} \ell_i(\tilde{y}) = L_{max}$

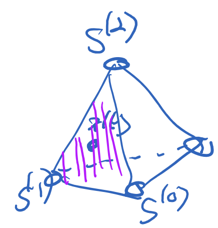
$d(\alpha^{(0)}) = 0$



FCFW "fully corrective FW" variant:

algorithm: re-optimize f over conv-hull$\left(\{s^{(u)}\}_{u=0}^{t}\right)$

FW corners



$\rightarrow$ think of it as doing "fancy line-search" on the correction polytope

note: could use AFW ...

→ think of it as doing "fancy line-search" on the correction polytope)

note: could use AFW to do correction step

(see "barrier FW" paper)

[special case: min norm point alg. (MNP)]
→ state of art alg. for submodular opt
↳ sequence of affine projections + line search

turns out that (batch) FCFW on dual of SVMstruct
is equivalent to the constraint generation / cutting plane approach on 1-slack formulation

why? every $s^{(t)}$ corresponds $\{\hat{y}_i^{(t)}\}_{i=1}^n$

$\alpha \in \mathrm{conv}\left(\{s^{(u)}\}_{u \le t}\right)$ $\quad \alpha = \sum_{u \le t} \tilde{\alpha}_u s^{(u)}$

$$w = A\alpha = \sum_{u \le t} \tilde{\alpha}_u \left(\frac{1}{\lambda n} \sum_{i=1}^n \psi_i(\hat{y}_i^{(u)}) \right)$$

to summarize:

	primal problem	dual problem	complexity of alg. in # of oracle calls
$O(n)$ oracle calls per iteration	batch subgradient $\beta_t = \frac{\gamma_t}{\lambda}$	batch FW ⊛ can do line search to get γ_t^*	$O\left(n \frac{R^2}{\lambda\varepsilon}\right)$
	1-slack cutting plane	FCFW	$O\left(n^{\nu} \log\left(\frac{1}{\varepsilon}\right)\right)$ $\nu \geq 1$? (open problem) related to pyramidal width of marginal polytope
1 oracle call per iteration	stochastic subgradient method (no line search)	BCFW (line search)	$O\left(1 \frac{R^2}{\lambda\varepsilon}\right)$

(not equivalent)

14h44

Block-coordinate optimization

"huge scale" optimization → [Nesterov 2010-2012]

proposed: randomized block-coordinate projected gradient method

setup: $\min f(x)$

s.t. $x \in \times_{i=1}^{n} M_i$

$x = (\underbrace{x_1}_{\text{block}}, \ldots, x_n)$

algorithm: pick at random i

then let $\begin{cases} x_i^{(t+1)} = \text{Proj}_{M_i}\left(x_i^{(t)} - \frac{1}{L_i} \nabla_i f(x^{(t)})\right) \\ x_j^{(t+1)} = x_j^{(t)} \quad \forall j \neq i \end{cases}$

Lipschitz constant of $\nabla_i f(x^{(t)})$

$x = (\underbrace{x_1}_{\text{block}}, \ldots, x_n)$

$\begin{cases} \ldots \\ x_j^{(t+1)} = x_j^{(t)} \quad \forall j \neq i \end{cases}$ L_i

[only update block i at iteration t]

Nesterov showed (uniform sampling) $\mathbb{E} f(x^{(t)}) - f(x^*) \leq \frac{2}{t+4}\left[\frac{1}{n}\sum_{i=1}^{n} L_i\right] \|x_0 - x^*\|^2$

(for convex f with L_i-Lipschitz gradient per block)

(*) aside: in constrained optimization, $-\frac{1}{\gamma_t}\left(x^{(t+1)} - x^{(t)}\right)$ plays the role of $\nabla f(x^{(t)})$

Block-coordinate FW (BCFW): idea: do a FW step on block i

alg: for $t = 0, \ldots$

pick i uniformly at random

let $s_i^{(t)} = \underset{s_i \in M_i}{\arg\min} \langle s_i, \nabla_i f(x^{(t)}) \rangle$ [FW corner for block i]

$$\begin{cases} x_i^{(t+1)} = x_i^{(t)}(1-\gamma_t) + \gamma_t s_i^{(t)} \\ x_j^{(t+1)} = x_j^{(t)} \text{ for } j \neq i \end{cases}$$

$$S_{[i]} = \begin{pmatrix} 0 \\ 0 \\ s_i \\ 0 \\ 0 \end{pmatrix}$$

$\gamma_t =$ line search: $\underset{\gamma \in [0,1]}{\arg\min} f\left(x^{(t)} + \gamma\left(S_{[i]}^{(t)} - x_{[i]}^{(t)}\right)\right)$

$\frac{2n}{t+2n}$ (n: # of blocks)

(*) an important property: FW-gap $= \max_{s \in M} \langle -\nabla f(x), s - x \rangle \overset{\text{product structure}}{=} \sum_i \max \langle -\nabla_i f(x), s_i - x_i \rangle$

(*) an important property: FW-gap $= \max_{s \in M} \langle -\nabla f(x), s-x \rangle \overset{\downarrow}{=} \sum_i \underbrace{\max_{s_i \in M_i} \langle -\nabla_i f(x), s_i - x_i \rangle}_{g_i(x) \text{ "block FW gap"}}$

$$\boxed{g^{FW}(x) = \sum_i g_i^{FW}(x)}$$

↳ motivated "gap sampling variant" [Osokin et al. ICML 2016]

as before, can show $g_i(x) \geq f(x) - \min_{\substack{y \\ \text{s.t. } y_j = x_j \, \forall j \neq i}} f(y)$

Convergence result for BCFW:

$C_f^{(i)} \leq L_i \operatorname{diam}(M_i)^2$ $\quad C_f^{(\times)} \triangleq \sum_{i=1}^{n} C_f^{(i)}$

$$\boxed{\mathbb{E}[\underbrace{f(x^{(t)}) - f(x^*)}_{\triangleq \varepsilon_t}] \leq \frac{2n}{t+2n}\left[C_f^{(\times)} + f(x^{(0)}) - f(x^*)\right] \text{ for } \gamma_t = \frac{2n}{t+2n}}$$

if you use line search: $\varepsilon_t \leq \frac{2n C_f^{(\times)}}{t - t_0 + 2n}$ for $t \geq t_0$

where $t_0 \leq n \log 2 \left(\frac{f(x^{(0)}) - f(x^*)}{C_f^{(\times)}}\right)$

time to ensure $\varepsilon_t \leq C_f^{(\times)}$

one can show that $C_f^{(\times)} \leq C_f$ for quadratic functions

compare batch FW $\longrightarrow \frac{2C_f}{t+2}$

vs.

BCFW with line search $\longrightarrow \frac{2n C_f^{(\times)}}{\cdots}$

vs.

BCFW with line search $\longrightarrow \dfrac{2n\, C_f^{\otimes}}{t - t_0 + 2n}$

Complexity

\# of oracle calls to reach ε-error

FW: $O\left(\dfrac{n C_f}{\varepsilon}\right)$

BCFW: $O\left(\dfrac{n C_f^{\otimes}}{\varepsilon}\right)$

but BCFW is often n times cheaper than batch FW

$\Rightarrow$ BCFW is "never" slower than batch FW