Lecture 18 - BCFW

Tuesday, March 23, 2021 14:32

today • FW for SVMstruct & FCFW
• BCFW

continue FW for SVMstruct

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

$\alpha^{(t)} \rightsquigarrow w^{(t)} = A\alpha^{(t)}$ $\qquad \gamma^{(FW)} = \gamma_t^{(\text{subgradient})}$

recall: subgradient method converges $O(\frac{1}{t})$ when step-size $\gamma_t \leq \frac{1}{\mu} O(\frac{1}{t})$

FW method with fixed step-size $\gamma_t = \frac{2}{t+2}$ gives $d(\alpha^*) - d(\alpha^{(t)}) \leq \frac{2C_f}{t+2}$

* FW gap: (for SVMstruct)

$$\langle -\nabla f(\alpha^{(t)}), s^{(t)} - \alpha^{(t)} \rangle$$

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

$$= p(\underbrace{w(\alpha^{(t)})}_{A\alpha^{(t)}}) - d(\alpha^{(t)}) \quad \text{[Lagrangian gap of lecture 15]}$$

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

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

* recall that

$$\min_{s \leq t} g_s \leq \frac{3 \cdot 2C_f}{t+2}$$

(FW gap)

$\Rightarrow$ guarantees on $\underbrace{p(w(\alpha^{(t)})) - p(w^*)}_{\text{primal subopt.}} + \underbrace{d(\alpha^*) - d(\alpha^{(t)})}_{\text{dual subopt.}}$

also (turns out) $g^{FW}(\hat{x}_{WA}^{(t)}) \leq \frac{3 \cdot 2C_f}{t+2}$

$\hat{x}_{WA}^{(t)}$: weighted avg.

when $g^{FW}(\cdot)$ is convex [this is case when f is

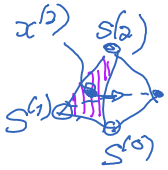
quadratic

$p(w(\alpha^{(0)})) = p(0) =: \hat{\xi} \geq \max_{y} \sum_{i=1}^{n} \ell_i(y)$

FCFW "fully corrective FW" variant

FW corners

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



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

note: could use AFW to do correction step approx.

(see "barrier FW")

[special case: min. norm point alg. MNP] → sequence of affine projections + line search to approximate the correction step

→ state of the art alg. for submodular opt.

* turns out that (batch) FCFW on dual struct is equivalent to the constraint generation / cutting plane alg on the 1-slack formulation (primal)

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

$\alpha \in \text{conv}\left(\{s^{(u)}\}_{u=0}^{t}\right) \quad \alpha = \sum_{u=1}^{t} \tilde{\alpha}_u s^{(u)}$

Solving QP in primal for cutting plane ⇔ correction step on dual for FCFW

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

to summarize:

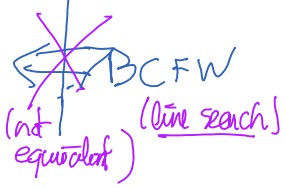
	primal problem		dual problem
$O(n)$ oracle calls per iteration	batch subgradient $\beta_t = \frac{\partial f}{\lambda}$	⇔	batch FW
	1-slack cutting plane	⇔	FCFW

complexity of alg → loss augmented decoding in # oracle calls

$O\left(n \frac{R^2}{\lambda \varepsilon}\right)$ vs $O\left(n n' \log\left(\frac{1}{\varepsilon}\right)\right)$ $\nu \geq 1$? (open problem) related to

plane

$O(1)$ { stochastic subgradient method (no line search)



$O\left(\frac{1}{\lambda\varepsilon}\right)$ (open problem) related to pyramidal widths of marginal polytope

15h35

block-coordinate optimization

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

proposed: randomized block-coordinate projected gradient method

setup: $\min f(x)$

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

$x = (x_1, \dots, x_n)$ (blocks)

algorithm: pick a random i_t

then let $\begin{cases} x_{i_t}^{(t+1)} = \text{Proj}_{M_{i_t}}\left(x_{i_t}^{(t)} - \frac{1}{L_{i_t}} \nabla_{i_t} f(x^{(t)})\right) \\ x_j^{(t+1)} = x_j^{(t)} \quad \forall j \neq i_t \end{cases}$

L_{i_t}: Lipschitz constant for $\nabla_{i_t} f(x^{(t)})$

[only updating block i_t at iteration t]

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

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

block-coordinate FW (BCFW): idea is to do a FW step on block i_t

alg.: for $t = 0, \dots$

pick i unif. at random from 1 to n

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

notation $s_{[i]} = (0, \dots, 0, s_i, 0, \dots, 0)$ ← block i

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

$\gamma_t = \begin{cases} \text{line search } \underset{\gamma \in [0,1]}{\text{argmin}} f(x^{(t)} + \gamma(s_{[i]}^{(t)} - x_{[i]}^{(t)})) \\ \frac{2n}{t+2n} \quad (n = \#\text{ of blocks}) \end{cases}$

... using ...

$\frac{2R}{t+2n}$

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

$\triangleq g_i(x)$ "block FW gap"

$$g^{FW}(x) = \sum_{i=1}^n g_i^{FW}(x)$$

↳ motivated "gap sampling" variant

as before, you can show that $g_i(x) \geq f(x) - \min_y f(y)$ s.t. $y_i = x_i \ \forall i$

[Osokin et al. ICML 2016]

Convergence of BCFW

$C_f^{(i)} \leq L_i \operatorname{diam}(\mathcal{M}_i)^2$ $\qquad C_f^{\otimes} \triangleq \sum_{i=1}^n C_f^{(i)}$

$$\underbrace{\mathbb{E} f(x^{(t)}) - f^*}_{\triangleq \varepsilon_t} \leq \frac{2n}{t+2n} \left[C_f^{\otimes} + f(x^{(1)}) - f^* \right] \quad \text{for } \gamma_t = \frac{2n}{t+2n}$$

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

$t_0 \leq n \log \frac{2(f(x^{(0)}) - f^*)}{C_f^{\otimes}}$

↑ time to ensure that $\varepsilon_t \leq C_f^{\otimes}$

batch FW: $\varepsilon_t \leq \frac{2C_f}{t+2}$

BCFW: $\varepsilon_t \leq \frac{2n C_f^{\otimes}}{t - t_0 + 2n}$

one can show that $C_f^{\otimes} \leq C_f$ for quadratic functions

C_f vs. $n C_f^{\otimes}$

but BCFW is n times cheaper than batch FW

complexity (# of oracle calls): FW $\to O(n C_f / \varepsilon)$

$\Rightarrow$ BCFW is "never" slower than batch FW (when not using parallelization)

BCFW $O(n C_f^{\otimes} / \varepsilon)$

but for SUM-struct you have $n C_f^{\otimes}$ is actually n times smaller than C_f !?

extensions to BCFW:

- non-uniform sampling e.g. $\{ g_i(x) \ / \ C_f^{(i)} \}$ — ICML 2016

- Non-uniform sampling eg- $\{g_i(x)$ / $G^{(i)}$
- Using away-Step, etc. to "get" linear convergence

} ICML 2016