Lecture 14 - scribbles - BCFW

Wednesday, February 22, 2017
10:29

today: · BCFW
· start theory?

Block-coordinate optimization:

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

proposed: block-coordinate projected gradient method

optimization setup: $\min f(x)$ s.t. $x \in \sum_{i=1}^{n} M_i$

ie. $x = (x_1, \dots, x_n)$ ← blocks

$\nabla = \begin{pmatrix} \nabla_1 \\ \vdots \\ \nabla_n \end{pmatrix}$

→ pick i at random

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

(only update update block i at iteration t)

Nesterov should get $\mathbb{E} f(x^{(t)}) - f(x^*) \leq \frac{2}{t+4} \left[\frac{1}{n} \sum_i L_i\right] \|x_0 - x^*\|^2$ (convex f)

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

algorithm: for $t = 0, \dots,$

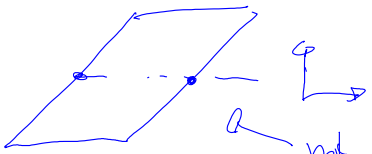
pick i at random

let $s_i^{(t)} = \underset{s_i \in M_i}{\text{argmin}} \langle \nabla_i f(x^{(t)}), s_i \rangle$ (FW corner for block i) $\quad S_{[i]} = \begin{pmatrix} 0 \\ s_i \\ 0 \end{pmatrix}$

$x_i^{(t+1)} = x_i^{(t)}(1-\gamma_t) + \gamma_t s_i^{(t)}$

$x_j^{(t+1)} = x_j^{(t)}$ for $j \neq i$

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



not a product space?

$2n$ → # of blocks

$\epsilon + 2n$

$\prod_{i=1}^{n} M_i$ looks like

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

$\overset{\text{product structure}}{=} \sum_i \max_{s_i \in M_i} \langle -\nabla_i f(x), s_i - x \rangle$

$g_i(x)$ "block FW gap"

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

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

convergence result: define $C_f^{(i)} \triangleq \sup_{\substack{\gamma \in [0,1] \\ s_i \in M_i \\ x \in M}} \frac{2}{\gamma^2}\left(f(x_\gamma) - f(x) - \gamma \langle \nabla f_i(x), s_i - x_i \rangle\right)$

$x_\gamma \triangleq x + \gamma (s_{[i]} - x_{[i]})$ (only one block updated)

$C_f^{(i)} \leq L_i \, \text{diam}(M_i)^2$

"block descent lemma"

during BCFW, if block i is updated,

$$f(x_\gamma) \leq f(x) + \gamma \underbrace{\langle \nabla f(x), s_i - x_i \rangle}_{-g_i(x)} + \frac{\gamma^2}{2} C_f^{(i)}$$

consider γ to be fixed γ_t

$$\mathbb{E}\left[f(x^{(t+1)}) \mid x^{(t)}\right] \leq f(x^{(t)}) - \gamma \underbrace{\mathbb{E}[g_i(x) \mid x]}_{\text{FW}} + \frac{\gamma^2}{2} \underbrace{\mathbb{E}[C_f^{(i)}]}_{\text{"}}$$

$\Rightarrow \quad \varepsilon_{t+1} \leq (1-\frac{\gamma}{n})\varepsilon_t + \frac{\gamma^2}{2n} C_f^{(\infty)}$ $\quad \frac{1}{n}\sum_i g^{(i)}(x) \quad \frac{1}{n}\sum_i C_f^{(i)} \to C_f^{(\infty)}$

$\rightsquigarrow$ do induction ...

$$\mathbb{E}[f(x^{(t)})] - f(x^*) \leq \frac{2n}{t+2n}\left[C_f^{(\infty)} + f(x^{(0)}) - f(x^*)\right] \qquad \text{for } \gamma_t = \frac{2n}{t+2n}$$

If you use line search:

$$\mathbb{E} f(x^{(t)}) - f(x^*) \leq \frac{2nC_f^{(\infty)}}{t - t_0 + 2n} \quad \text{for } t \geq t_0$$

where $t_0 \leq n \log \frac{2\varepsilon_0}{C_f^{(\infty)}}$ # times to ensure that $\varepsilon_t \leq C_f^{(\infty)}$

using from $\varepsilon_t \leq \varepsilon_0 \exp(-t/n) + \frac{C_f^{(\infty)}}{2}$ when $\varepsilon_t \geq C_f^{(\infty)}$

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

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

vs BCFW with line search $\to \frac{2nC_f^{(\infty)}}{t-t_0+2n}$ (say $t_0 = 0$ for simplicity)

BCFW update is n times cheaper than batch FW (i.e. # of block oracle call to reach ε-error)

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

FW: $O\left(\frac{nC_f}{\varepsilon}\right)$

BCFW: $O\left(\frac{nC_f^{(\infty)}}{\varepsilon}\right)$

Extensions: • non-uniform sampling e.g. using $\{C_f^{(i)}$ / $\{g^i(x)\}$ } ICML 2016

• using away steps etc.. to "get" linear convergence

Applications to SVM struct:

- getting s_i is one loss-augmented decoding call on example i
- $w = A\alpha = \sum A_i \alpha_i$

$\dots$ i $\underbrace{\quad}_{\triangleq w_i}$

need to store in memory

when you do a BCFW step,

you update $w_i^{(t+1)} = w_i^{(t)} + \gamma(w_s - w_i^{(t)})$

for SVMstruct $C_f^{(i)} = \frac{4B^2}{\lambda n^2} \Rightarrow C_f^{(k)} = \frac{4B^2}{\lambda n} = \frac{C_f}{n}$

Theory basics:

decision theory setup:

estimate: $h_w : \mathcal{X} \to \mathcal{Y}$

task loss

generalization error $= L_P(w) \triangleq \mathbb{E}_{(x,y)\sim P}[\ell(y, h_w(x))]$

ultimate goal is find $w^* = \arg\min_{w \in \mathcal{W}} L_P(w)$

problem is do not know P (distribution on (x,y))

suppose $\underbrace{(x^{(i)}, y^{(i)})_{i=1}^n}_{D_n} \overset{iid}{\sim} P$ training data $\rightsquigarrow$ we can look at $\hat{L}_n(w) = \frac{1}{n}\sum_{i=1}^n \ell(y^{(i)}, h_w(x^{(i)}))$

learning algorithm $\hat{w}_n = \overset{\text{algorithm}}{A}(D_n)$

from statistics/probs

$\hat{L}_n(w) \overset{a.s.}{\to} L(w) \quad \forall w$

note that this is much weaker that $\sup_w |\hat{L}_n(w) - L(w)| \overset{n\to\infty}{\to} 0$

note that minimizing training error gives not much guarantees in general

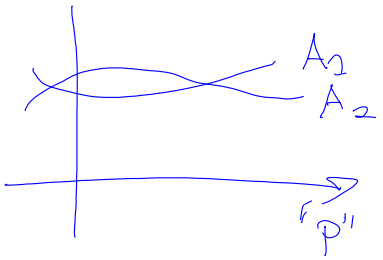
regression

for n points, can get zero training error with polynomial of degree $\geq n-1$

suppose you have no noise
real fct. is quadratic;

learning theory, we want to understand properties of learning algorithms
in particular, what can I say about $L_p(A(D_n))$?

"frequentist risk" $\mathbb{E}_{D_n \sim p^n}[L_p(A(D_n))] = R^F_p(A)$
D_n is random



PAC framework
("Probably Approximately Correct")
$P\{ L_p(A(D_n)) > \text{some bound} \} \leq \delta$

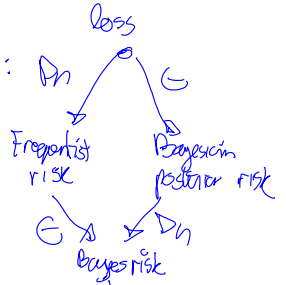
[which means $L_p(A(D_n)) \leq$ bound with prob bigger that $1-\delta$ on random D_n]

generalization error bound

"Bayesian posterior risk" $R^{post}(w|D_n) = \mathbb{E}_{\theta \sim p(\theta|D_n)}[L_{p_\theta}(w)]$ prior $p(\theta)$ over distributions
posterior $p(\theta|D_n)$

Bayesian estimate $\hat{w}_n = \operatorname{argmin}_w R^{post}(w|D_n)$

could analyze the property of $R^F_p(A^{Baye}(D_n))$ i.e. $R^F_p(\hat{w}_n)$

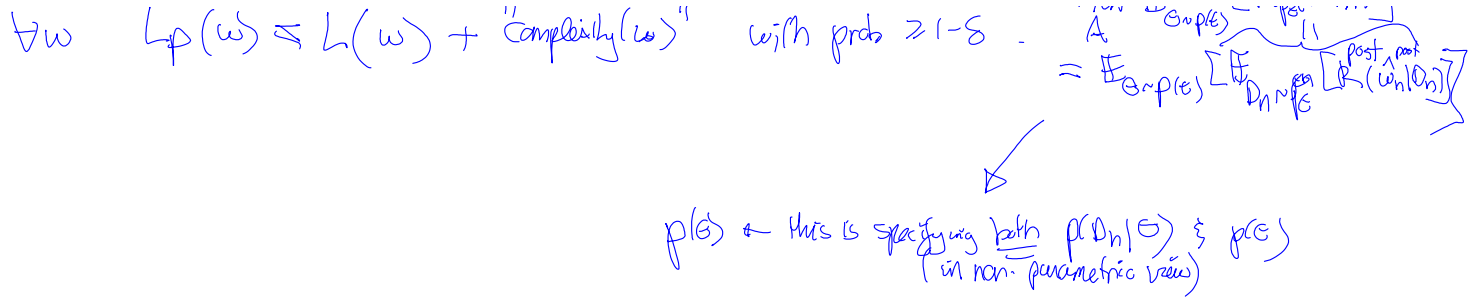


$\min_A \mathbb{E}_{\theta \sim p(\theta)}[R^F_{p_\theta}(A(D_n))]$
$= \mathbb{E}_{D_n \sim p^n}[\mathbb{E}_{\theta \sim p(\theta|D_n)}[R^{post}_{p_\theta}(\hat{w}_n|D_n)]]$

next time:

Occam's bound:

$\forall w \quad L_p(w) \leq \hat{L}(w) + \text{"complexity}(w)\text{"}$ with prob $\geq 1-\delta$.



Pointers:

- Nesterov coordinate method:
  - Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems, Y. Nesterov, SIAM J. Optim., 22(2), 341–362. 2012
    epubs.siam.org/doi/pdf/10.1137/100802001
- BCFW in all its gory details:
  - Block-Coordinate Frank-Wolfe Optimization for Structural SVMs, S. Lacoste-Julien, M. Jaggi, M. Schmidt and P. Pletscher, ICML 2013
    http://jmlr.csail.mit.edu/proceedings/papers/v28/lacoste-julien13-supp.pdf
    code: https://github.com/ppletscher/BCFWstruct
- Improvements of BCFW for SVMstruct (non-uniform sampling, away steps, etc.):
  - Minding the Gaps for Block Frank-Wolfe Optimization of Structured SVMs, A. Osokin, J.-B. Alayrac, I. Lukasewitz, P. Dokania and S. Lacoste-Julien, ICML 2016
    https://arxiv.org/abs/1605.09346
    code: http://www.di.ens.fr/sierra/research/gapBCFW/