Lecture 20 - variance reduction

Thursday, March 30, 2023 1:29 PM

today: • variance reduction perspective
• application CRF

variance reduction idea:

X & Y are R.Vs,

goal: estimate $\mathbb{E}X$ using M.C. samples

suppose: $\mathbb{E}Y$ is cheap to compute and Y is correlated with X

consider estimator:

$\alpha \in [0,1]$ ← convex combo coeff. between $\mathbb{E}X$ & $\mathbb{E}Y$

$\theta_\alpha \triangleq \alpha(X-Y) + \mathbb{E}Y$ to approximate $\mathbb{E}X$

properties: $\mathbb{E}\theta_\alpha = \alpha\mathbb{E}X + (1-\alpha)\mathbb{E}Y$ → unbiased (i.e. $\mathbb{E}\theta_\alpha = \mathbb{E}X$)

if $\mathbb{E}X = \mathbb{E}Y$ [not interesting]
$\alpha = 1$

variance: $\mathrm{var}(\theta_\alpha) = \alpha^2[\mathrm{Var}X + \mathrm{Var}Y - 2\mathrm{cov}(X,Y)]$

variance reduction?

for $\alpha = 1$ (unbiased setting) $\theta_\alpha = X + (\mathbb{E}Y - Y)$
correction

SGD setting:

X is $\nabla f_i(x_t)$; $\mathbb{E}X$ = batch gradient

SAG/SAGA algorithm: Y is g_i [past stored gradient]

$$\mathbb{E}Y = \frac{1}{n}\sum_{i=1}^{n} g_i$$

SAG alg.: $\alpha = \frac{1}{n}$ (biased)

SAGA alg.: $\alpha = 1$ ⇒ unbiased

$\alpha(X-Y) + \mathbb{E}Y$

$\alpha(X-Y) + \mathbb{E}Y$

SAG: $x_{t+1} = x_t - \gamma\left[\frac{1}{n}\left[\nabla_{i_t} f(x_t) - g_{i_t}^{(t)}\right] + \frac{1}{n}\sum_{i=1}^{n} g_i^{(t)}\right]$ (biased)

SAGA: $x_{t+1} = x_t - \gamma\left[1\left[\quad\|\quad - \quad\|\quad\right] + \quad\|\quad\right]$ (unbiased)

SVRG: $x_{t+1} = x_t - \gamma\left[1\left[\nabla f_i(x_t) - \nabla f_i(x_{old})\right] + \frac{1}{n}\sum_i \nabla f_i(x_{old})\right]$ (unbiased)

(stochastic variance reduced gradient)

↳ x_{old} is updated from outer loop

SVRG algorithm:

for $k = 0, \ldots$ (outer loop)

compute $g_{ref} \triangleq \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x^{(k)})$

for $t = 0, \ldots, T_{max}$

sample i_t

$x_{t+1}^{(k)} = x_t^{(k)} - \gamma\left[\nabla f_{i_t}(x_t^{(k)}) - \nabla f_{i_t}(x^{(k)}) + g_{ref}\right]$

end

$x^{(k+1)} = x_{T_{max}}^{(k)}$

end

Questions:

- what is T_{max}?
- what is γ?

original SVRG convergence result: need $\gamma \leq \approx \frac{1}{L}$

$T_{max} \geq \approx \frac{L}{\mu} = \kappa$ → to run alg., need to know κ

⇒ not adaptive to local strong convexity

tweak of SVRG (now called "loopless")

[Hoffmann & al. NeurIPS 2015] $T_{max} \sim \text{Geom}(\_)$

[at inner loop, do a batch gradient comp. with prob $\frac{1}{n}$]

↳ then get same convergence as SAGA

↳ comp. cost: size of inner loop $\mathbb{E}[T_{max}] = n$

overall cost of SVRG $\approx$ 3(SGD cost) for n updates

SAG / SAGA / loopless SVRG, convergence for convex fct. ($\mu=0$) get $\min_{t \le T} \{ f(x_t) - f^* \} = O(\frac{1}{T})$

[contrast this with $O(\frac{1}{\sqrt{T}})$ for SGD]

(*) note: interpolation regime: $\|\nabla f_i(x^*)\| = 0 \ \forall i$

[vs. just. $\|\frac{1}{n}\sum_i \nabla f_i(x^*)\| = 0$]

↳ get similar rate as SVRG/SAGA with SGD

CRF optimization

primal $\to \max_{\tilde{y}} \ell_i(\tilde{y}) - w^T \psi_i(\tilde{y})$ (margin) dual

SVM struct

$$\min_w \frac{\lambda\|w\|^2}{2} + \frac{1}{n}\sum_{i=1}^n H_i(w) \qquad \max_{\alpha_i \in \Delta_{|\mathcal{Y}_i|}} -\frac{\lambda}{2}\|w(\alpha)\|^2 + \frac{1}{n}\sum_{i=1}^n \ell_i^T \alpha_i$$

CRF

$$\min_w \frac{\lambda\|w\|^2}{2} + \frac{1}{n}\sum_{i=1}^n -\log p(y^{(i)} | x^{(i)}; w) \qquad \max_{\alpha_i \in \Delta_{|\mathcal{Y}_i|}} -\frac{\lambda}{2}\|w(\alpha)\|^2 + \frac{1}{n}\sum_{i=1}^n H_i(\alpha_i)$$

$-\log p(y^{(i)} | x^{(i)}; w) = \log\left(\sum_{\tilde{y}} \exp(-w^T \psi_i(\tilde{y}))\right)$

entropy $H_i(\alpha_i) \triangleq -\sum_{\tilde{y}} \alpha_i(\tilde{y}) \log \alpha_i(\tilde{y})$

KKT $\to$ $w(\alpha) = \frac{1}{\lambda n}\sum_i \sum_{\tilde{y}} \alpha_i(\tilde{y}) \psi_i(\tilde{y})$

$= \frac{1}{\lambda n}\sum_i \sum_{c \in \mathcal{C}} \sum_{\tilde{y}_c} \mu_{i,c}(\tilde{y}_c) \psi_{i,c}(\tilde{y}_c)$

μ from MRF

$p(\tilde{y} | x; w) \propto \exp(\langle w, \varphi(x, \tilde{y}) \rangle)$

(*) at optimality

$\alpha_i^*(\tilde{y}) = p(\tilde{y} | x^{(i)}; w(\alpha^*))$

$\alpha_i^* \in$ interior of $\Delta_{|\mathcal{Y}_i|}$

unlike sparse solution in SVM struct

14h33

CRF optimization

- primal is smooth [vs. non-smooth for SVM struct]

- for a while, batch L-BFGS was method of choice [batch ⇒ slow for large n]
- [Collins & al. JMLR 2008]: online exponentiated gradient (OEG)

  block-coordinate method on dual; exponentiated gradient step on block

  $$\alpha_i(\tilde{y})^{(t+1)} \propto \alpha_i(\tilde{y})^{(t)} \exp\left(-\gamma_t \nabla_{\alpha_i} D(\alpha^{(t)})\right)$$ (dual obj.)

  EG alg → proximal gradient step using $KL(\alpha \| \alpha^{(t)})$ as a Bregman divergence for prox term

  → get linear convergence rate with cheap $O(1)$ updates (like SGD) [vs. $O(n)$ for batch method]

  [can think of it as a variance reduced method as well?]

- SAGA for CRF [Schmidt & al. AISTATS 2015]

$$w^{(t+1)} = (1-\lambda\gamma_t) w^{(t)} - \gamma_t \left[\nabla f_i(w^{(t)}) - g_i^{(t)} + \frac{1}{n}\sum_j g_j^{(t)}\right]$$

- SDCA (stochastic dual coordinat ascent) → SOTA for CRF [Le Priol & al UAI 2015] thanks line search

$$\alpha_i^{(t+1)}(\tilde{y}) = (1-\gamma_t)\alpha_i^{(t)}(\tilde{y}) + \gamma_t \tilde{s}_i(\tilde{y}), \quad \gamma_t \in [0,1] \rightsquigarrow \text{stochastique}$$

as a relaxed fixed pt. iteration

$$\alpha_i(\tilde{y}) = p(y \mid x_i; w(\alpha^{(t)}))\ \forall i$$

$p(\tilde{y} \mid x_i; w(\alpha^{(t)})) \equiv q_i(w)$

related to subgradient ascent

[note: BCFW is a special case of SDCA on SVM-struct obj.]

proximal gradient method

↳ generalization of projected gradient method to other non-smooth fct.

composite framework $F(w) \triangleq f(w) + \Omega(w)$ where f is convex, L-smooth

- constrained opt. $\Omega(w) = \delta_M(w) \triangleq \begin{cases} 0 & \text{if } w \in M \\ +\infty & \text{o.w.} \end{cases}$ ("indicator fct." on M)

  Ω "nice" but not necess. smooth

- ℓ_1-regularization $\Omega(w) = \|w\|_1$

proximal gradient update:

$$w_{t+1} = \operatorname{argmin}_w \underbrace{f(w_t) + \langle \nabla f(w_t), w - w_t \rangle + \frac{1}{2\gamma_t} \|w - w_t\|_2^2}_{\triangleq B_t(w)} + \Omega(w)$$

- if $\gamma \leq \frac{1}{L}$, then $f(w) \leq B_t(w)$ $\forall w$
- we can rewrite $B_t(w) = \frac{1}{2\gamma_t} \| w - [w_t - \gamma_t \nabla f(w_t)] \|_2^2 + \text{cst.}$ (by completing square)

$\Rightarrow$ if $\Omega(w) = \delta_\mu(w)$; we get projected gradient alg.

$$w_{t+1} = \operatorname{prox}_{\gamma_t}^{\Omega}(w_t - \gamma_t \nabla f(w_t))$$

↳ "proximal operator"

$$\operatorname{prox}_\gamma^\Omega(z) \triangleq \operatorname{argmin}_w \left\{ \Omega(w) + \frac{1}{2\gamma} \|w - z\|^2 \right\}$$

could replace by Bregman divergence to get other generalization (e.g. OEG)

⊗ like projection, prox operator is non-expansive (i.e. 1-Lipschitz)

i.e. $\|\operatorname{prox}(w) - \operatorname{prox}(w')\|_2 \leq \|w - w'\|_2$ (recall lecture 11 landscape of rates)

$\Rightarrow$ convergence rate for prox gradient method on $F = f + \Omega$ are same as unconstrained gradient descent on f

* to be useful, need $\operatorname{prox}_\gamma^\Omega$ to be efficiently computable

$$\operatorname{prox}_\gamma^{\|\cdot\|_1}(z) = \operatorname{argmin}_w \|w\|_1 + \frac{1}{2\gamma}\|w - z\|_2^2$$

"soft-thresholding" (component wise) $\triangleq \begin{cases} \operatorname{sgn}(z_i)[|z_i| - \gamma] & \text{if } |z_i| \geq \gamma \\ 0 & \text{o.w.} \end{cases}$

used e.g. for lasso: ℓ_1-reg. least-square

FISTA → accelerated prox-gradient method

↳ SOTA for batch lasso

$\min_w \frac{1}{n}\sum_i f_i(w) + \Omega(w)$

* scikit-learn → use (prox) SAGA for lasso & ℓ_1-reg log reg.

prox SAGA $w_{t+1} = \operatorname{prox}_{\gamma\delta}^{\Omega}\left[w_t - \gamma_t \left[\nabla f_{i_t}(w_t) - g_{i_t}^{(t)} + \frac{1}{n}\sum_j g_j^{(t)} \right] \right]$

prox SAGA $w_{t+1} = \text{prox}^{\Omega}_{\gamma\delta}\left[w_t - \gamma\epsilon\left[\nabla f_{i_t}(w_t) - g_{i_t}^{(t)} + \frac{1}{n}\sum_j g_j^{(t)}\right]\right]$

Could accelerate prox SAGA using "catalyst" [see next class]