The Tradeoffs of Large Scale Learning

Leon Bottou, Olivier Bousquet

Arthur Gretton's notes

January 8, 2015

What the paper is about

When you have big data (tm) and not much time, what is the best way to learn?

- Gradient descent (first or second order)?
- Online (stochastic gradient descent, first or second order)?

"It is known" that for large-scale problems, stochastic methods are better. This paper proves why this should occur (for linear functions $f = w^\top x$).

What the paper is about

When you have big data (tm) and not much time, what is the best way to learn?

- Gradient descent (first or second order)?
- Online (stochastic gradient descent, first or second order)?

"It is known" that for large-scale problems, stochastic methods are better. This paper proves why this should occur (for linear functions $f = w^\top x$).

Outline:

- How well *can* we learn with n samples?
- How do we trade off *time spent optimizing* and generalization performance?

What is a learning problem?

Expected risk:

$$E(f) = \int \ell(f(x), y) dP(x, y)$$

for loss $\ell(f(x), y)$. Best possible function:

$$f^*(x) := \arg\min_{\hat{y}} \mathbb{E}\left[\ell(\hat{y}, y)|x\right].$$

If we're constrained to smaller function class $\mathcal{F}$, best answer:

$$f^*_{\mathcal{F}} := \arg\min_{f \in \mathcal{F}} E(f)$$

What is a learning problem?

Expected risk:

$$E(f) = \int \ell(f(x), y) dP(x, y)$$

for loss $\ell(f(x), y)$. Best possible function:

$$f^*(x) := \arg\min_{\hat{y}} \mathbb{E}\left[\ell(\hat{y}, y)|x\right].$$

If we're constrained to smaller function class $\mathcal{F}$, best answer:

$$f^*_{\mathcal{F}} := \arg\min_{f \in \mathcal{F}} E(f)$$

Empirical risk:

$$E_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i).$$

Empirical risk minimizer:

$$f_n := \arg\min_{f \in \mathcal{F}} E_n(f).$$

How well have we learned?

The excess error tells us how well we learned:

$$\begin{aligned}\mathcal{E} &:= \mathbb{E}\left[E(f_n) - E(f^*)\right] \\ &= \underbrace{\mathbb{E}\left[E(f^*_{\mathcal{F}}) - E(f^*)\right]}_{\mathcal{E}_{\text{app}}} + \underbrace{\mathbb{E}\left[E(f_n) - E(f^*_{\mathcal{F}})\right]}_{\mathcal{E}_{\text{est}}}.\end{aligned}$$

- $\mathcal{E}_{\text{app}}$: approximation error (small for rich $\mathcal{F}$)
- $\mathcal{E}_{\text{est}}$: estimation error (small for simple $\mathcal{F}$)
- $\mathbb{E}$: expectation over n-sample (relevant in $\mathcal{E}_{\text{est}}$)

How well have we learned?

The excess error tells us how well we learned:

$$\begin{aligned}\mathcal{E} &:= \mathbb{E}\left[E(f_n) - E(f^*)\right] \\ &= \underbrace{\mathbb{E}\left[E(f_{\mathcal{F}}^*) - E(f^*)\right]}_{\mathcal{E}_{\text{app}}} + \underbrace{\mathbb{E}\left[E(f_n) - E(f_{\mathcal{F}}^*)\right]}_{\mathcal{E}_{\text{est}}}.\end{aligned}$$

- $\mathcal{E}_{\text{app}}$: approximation error (small for rich $\mathcal{F}$)
- $\mathcal{E}_{\text{est}}$: estimation error (small for simple $\mathcal{F}$)
- $\mathbb{E}$: expectation over n-sample (relevant in $\mathcal{E}_{\text{est}}$)

What if we learn only to some precision?

$$E_n(\tilde{f}_n) \leq E_n(f_n) + \rho$$

Additional term

$$\mathcal{E}_{\text{opt}} = \mathbb{E}\left[E(\tilde{f}_n) - E(f_n)\right]$$

How well have we learned?

The excess error tells us how well we learned:

$$\begin{aligned}\mathcal{E} &:= \mathbb{E}\left[E(f_n) - E(f^*)\right] \\ &= \underbrace{\mathbb{E}\left[E(f_{\mathcal{F}}^*) - E(f^*)\right]}_{\mathcal{E}_{\text{app}}} + \underbrace{\mathbb{E}\left[E(f_n) - E(f_{\mathcal{F}}^*)\right]}_{\mathcal{E}_{\text{est}}}.\end{aligned}$$

- $\mathcal{E}_{\text{app}}$: approximation error (small for rich $\mathcal{F}$)
- $\mathcal{E}_{\text{est}}$: estimation error (small for simple $\mathcal{F}$)
- $\mathbb{E}$: expectation over n-sample (relevant in $\mathcal{E}_{\text{est}}$)

New excess error

$$\mathcal{E} = \underbrace{\mathbb{E}\left[E(f_{\mathcal{F}}^*) - E(f^*)\right]}_{\mathcal{E}_{\text{app}}} + \underbrace{\mathbb{E}\left[E(f_n) - E(f_{\mathcal{F}}^*)\right]}_{\mathcal{E}_{\text{est}}} + \underbrace{\mathbb{E}\left[E(\tilde{f}_n) - E(f_n)\right]}_{\mathcal{E}_{\text{opt}}}.$$

How well *can* we learn?

Our setting: $f := w^\top x$ for $w \in \mathbb{R}^d$, and x, y, ℓ bounded.
Best bounds

$$\mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}} \leq c \left(\mathcal{E}_{\text{app}} + \left(\frac{d}{n} \log \frac{n}{d} \right)^{\alpha} \right) \qquad \frac{1}{2} \leq \alpha \leq 1,$$

if we are allowed to assume variance condition

$$\forall f \in \mathcal{F}, \quad \mathbb{E}\left(\ell(f(x), y) - \ell(f^*_{\mathcal{F}}(x), y)\right)^2 \leq c \left(E(f) - E(f^*_{\mathcal{F}})\right)^{2-\alpha^{-1}}$$

(large α is easier).[1]

[1] Not very intuitive: for *classification*, clearer condition is Tsybakov noise condition, $\exists \mu > 0, \beta \in (0, \infty)$ s.t. $\forall \varepsilon > 0$, $\mathbb{P}\left(|\eta(x) - 1/2| \leq \epsilon\right) \leq \mu \epsilon^{\beta}$.

How well *can* we learn?

Our setting: $f := w^\top x$ for $w \in \mathbb{R}^d$, and x, y, ℓ bounded.
Best bounds

$$\mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}} \leq c \left(\mathcal{E}_{\text{app}} + \left(\frac{d}{n} \log \frac{n}{d} \right)^{\alpha} \right) \qquad \frac{1}{2} \leq \alpha \leq 1,$$

if we are allowed to assume variance condition

$$\forall f \in \mathcal{F}, \quad \mathbb{E}\left(\ell(f(x), y) - \ell(f^*_{\mathcal{F}}(x), y)\right)^2 \leq c \left(E(f) - E(f^*_{\mathcal{F}})\right)^{2-\alpha^{-1}}$$

(large α is easier).[1]
Then

$$\mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}} + \mathcal{E}_{\text{opt}} \leq c \left(\mathcal{E}_{\text{app}} + \left(\frac{d}{n} \log \frac{n}{d} \right)^{\alpha} + \rho \right)$$

[1] Not very intuitive: for *classification*, clearer condition is Tsybakov noise condition, $\exists \mu > 0, \beta \in (0, \infty)$ s.t. $\forall \varepsilon > 0$, $\mathbb{P}\left(|\eta(x) - 1/2| \leq \epsilon\right) \leq \mu \epsilon^{\beta}$.

Now to optimize: first some definitions

Recall $f_w = w^\top x$. Empirical cost function

$$C = E_n(f_w).$$

Empirical optimum occurs at w_n.

Now to optimize: first some definitions

Recall $f_w = w^\top x$. Empirical cost function

$$C = E_n(f_w).$$

Empirical optimum occurs at w_n.
Hessian at optimum is

$$H = \frac{d^2 C}{dw^2}(w_n) \quad \text{eigenvalues} \in [\lambda_{\min}, \lambda_{\max}], \quad \text{condition}\#\kappa = \lambda_{\max}/\lambda_{\min}.$$

Gradient covariance at optimum

$$G = \mathbb{E}_n \left(\left(\frac{\partial \ell(f_{w_n}(x), y)}{\partial w} \right) \left(\frac{\partial \ell(f_{w_n}(x), y)}{\partial w} \right)^\top \right) \qquad \operatorname{tr}(G^{-1} H) \leq \nu.$$

(statements on eigenvalue range and bound on $\operatorname{tr}(G^{-1}H)$ are w.h.p. since quantities are empirical).

How fast to optimize to precision ρ?

Block strategies:

- Gradient descent: precision ρ with steps[2] $O(\kappa \log(1/\rho))$,

$$w(t+1) = w(t) - \eta \frac{\partial C}{\partial w}(w(t)).$$

  Time to reach ρ: $O(nd\kappa \log(1/\rho))$

- "Magical" second order gradient descent (we are given H): ρ in steps $O(\log\log(1/\rho))$,

$$w(t+1) = w(t) - H^{-1} \frac{\partial C}{\partial w}(w(t)).$$

  Time to reach ρ: $O((d^2 + nd)\log\log(1/\rho))$
  (no κ, better dependence on ρ)

[2] Given stepsize $\eta = \lambda_{\max}^{-1}$

How fast to optimize to precision ρ?

Stochastic gradient descent strategies:

- ***Stochastic*** gradient descent: precision ρ with steps[3] $\nu\kappa^2\rho^{-1} + o(1/\rho)$,

$$w(t+1) = w(t) - \frac{\eta}{t}\frac{\partial}{\partial w}\left[\ell(f_{w(t)}(x_t), y_t)\right].$$

  Time to reach ρ: $O(d\nu\kappa^2/\rho)$, (note: no n.

- "Magical" second order ***stochastic*** gradient descent: ρ in steps $\nu\rho^{-1} + o(1/\rho)$,

$$w(t+1) = w(t) - \frac{1}{t}H^{-1}\frac{\partial}{\partial w}\left[\ell(f_{w(t)}(x_t), y_t)\right].$$

  Time to reach ρ: $O(d^2\nu/\rho)$
  (no κ, same dependence on ρ)

[3] Given stepsize $\eta = \lambda_{\min}^{-1}$

Putting all the results together

What is time to get error ε above $\mathcal{E}_{\text{app}}$? (use $n \sim d\varepsilon^{-1/\alpha} \log(\alpha^{-1})$ for batch)

Time to reach $\mathcal{E} \leq c\,(\mathcal{E}_{\text{app}} + \varepsilon)$	
$\mathcal{O}\left(\frac{d^2\,\kappa}{\varepsilon^{1/\alpha}} \log^2 \frac{1}{\varepsilon}\right)$	GD
$\mathcal{O}\left(\frac{d^2}{\varepsilon^{1/\alpha}} \log \frac{1}{\varepsilon} \log\log \frac{1}{\varepsilon}\right)$	2GD
$\mathcal{O}\left(\frac{d\,\nu\,\kappa^2}{\varepsilon}\right)$	SGD
$\mathcal{O}\left(\frac{d^2\,\nu}{\varepsilon}\right)$	2SGD

- Stochastic methods have the best generalization performance, despite having the worst optimization performance on the empirical cost.
- Fast convergence in SGD bounds doesn't depend on α (but watch out for constants!).

Experiment 1: logistic loss

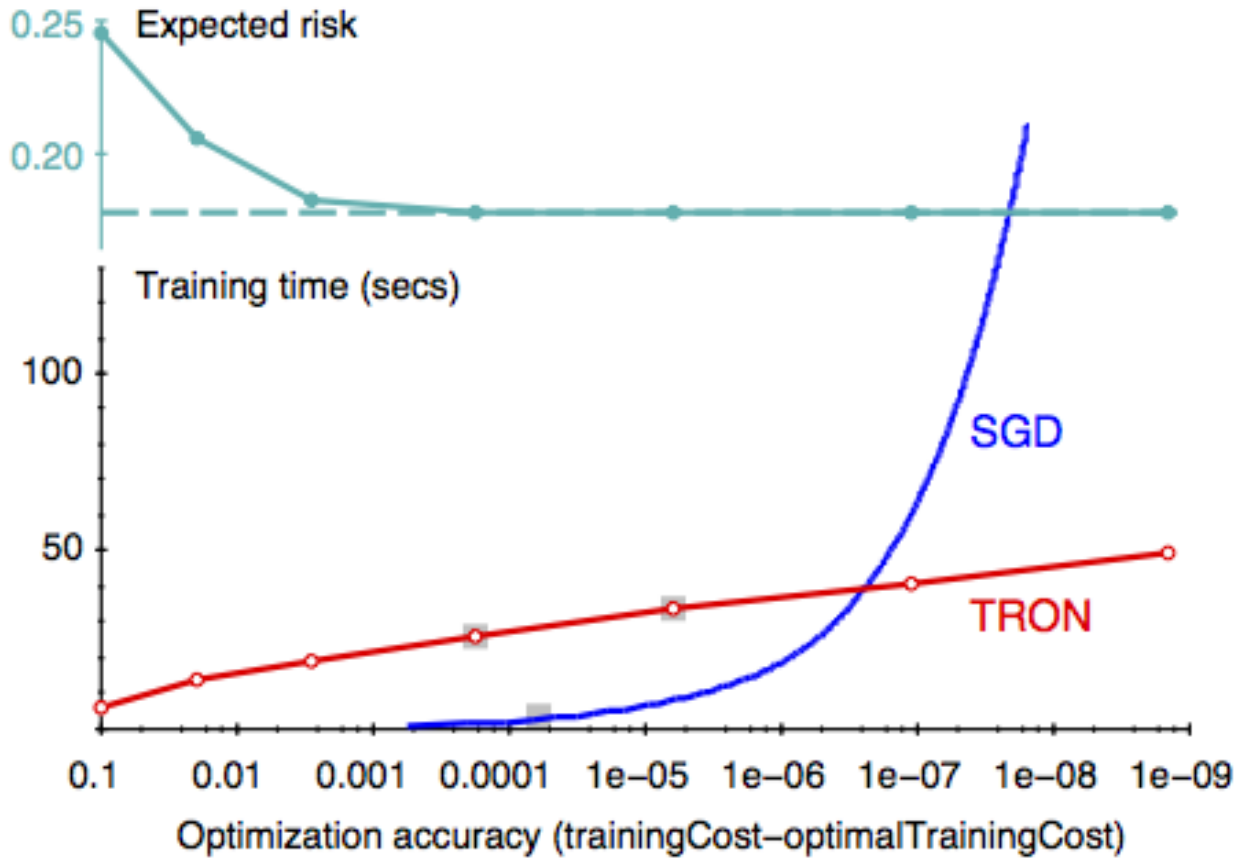

Figure: Superlinear batch method (TRON) vs SGD

Experiment 2:

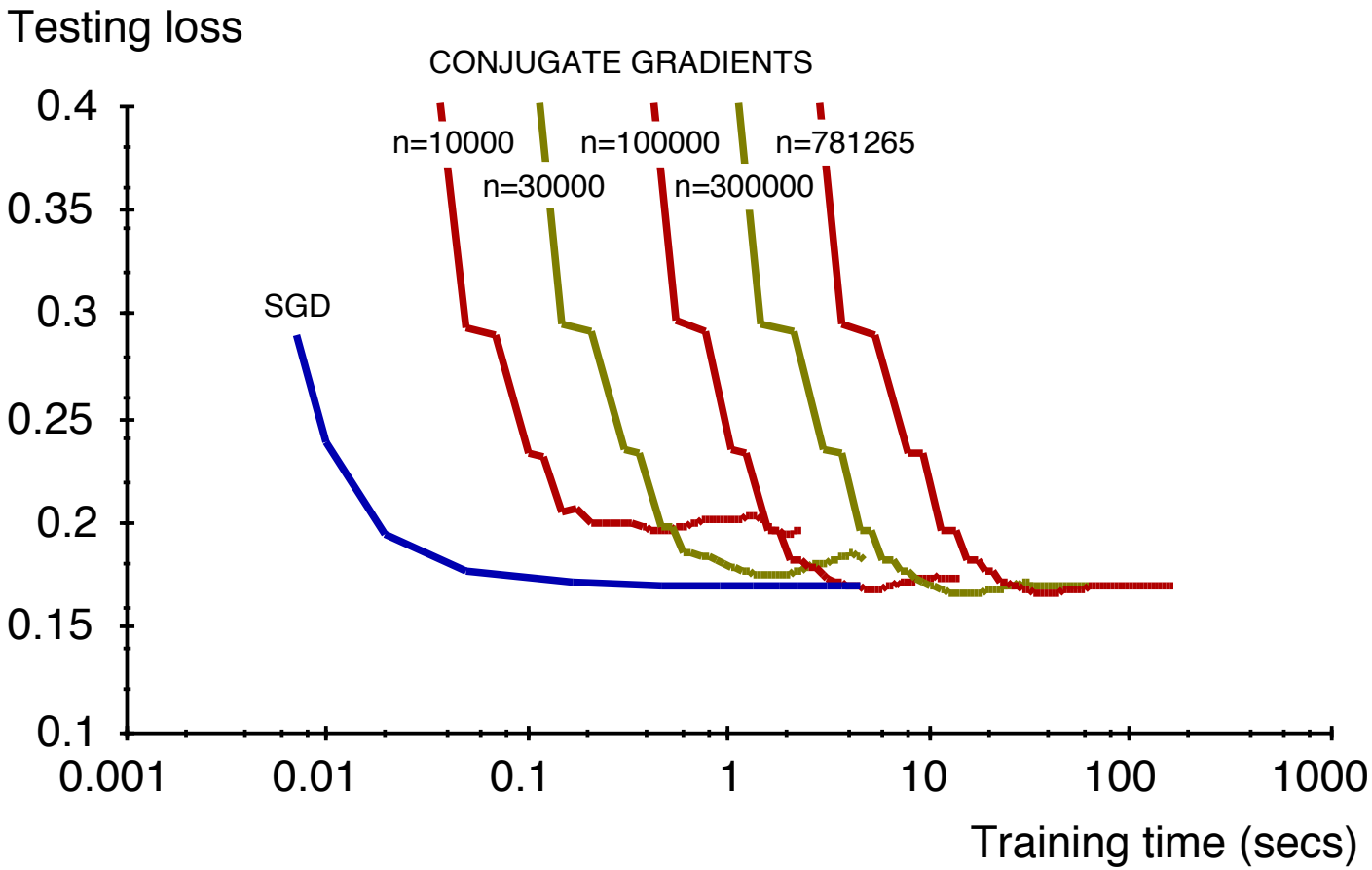

Figure: Conjugate gradients vs. SGD