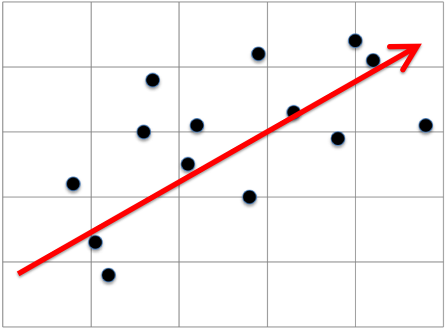
Tradeoffs in Large Scale Learning: Statistical Accuracy vs. Numerical Precision

Sham M. Kakade

Microsoft Research, New England

Statistical estimation involves optimization



Problem:

- Find the minimizer w_* of

$$L(\vec{w}) = \mathbb{E}[\text{loss}(\vec{w}, \text{point})]$$

- You only get n samples.

Example: Estimate a linear relationship with n points in d dimensions?

- Costly on large problems: $O(nd^2 + d^3)$ runtime, $O(d^2)$ memory
- How should we approximate our solution?

Statistics vs. Computation

Stochastic approximation	Numerical analysis
e.g. stochastic gradient descent	e.g. (batch) gradient descent
obtain poor accuracy, quickly?	obtain high accuracy, slowly?
simple to implement	more complicated

What would you do?

Machine learning at scale!

Vowpal Wabbit

The Vowpal Wabbit (VW) project is a fast out-of-core learning system sponsored by Microsoft Research and (previously) Yahoo! Research. Support is available through the mailing list.

torch
A SCIENTIFIC COMPUTING FRAMEWORK FOR LUAJIT

Sibyl

Caffe

Deep learning framework developed by Yangqing Jia / BVLC

ML base

Can we provide libraries to precisely do statistical estimation at scale?

Analogous to what was done with our linear algebra libraries? (LAPACK/BLAS)?

The Stochastic Optimization Problem

$$\min_w L(w) \text{ where } \quad L(w) = \mathbb{E}_{\text{point}\sim\mathcal{D}}[\text{loss}(w, \text{point})]$$

- With N sampled points from $\mathcal{D}$,

$$p_1, p_2, \ldots p_N$$

  how do you estimate w_*, the minima of P?
- Your expected error/excess risk/regret is:

$$\mathbb{E}[L(\widehat{w}_N) - L(w_*)]$$

Goal: Do well statistically. Do it quickly.

Without Computational Constraints...

What would you like to do?

Compute the empirical risk minimizer /M-estimator:

$$\widehat{w}_N^{\mathrm{ERM}} \in \operatorname*{argmin}_w \frac{1}{N} \sum_{i=1}^{N} \mathrm{loss}(w, p_i).$$

Consider the ratio:

$$\frac{\mathbb{E}[L(\widehat{w}_N^{\mathrm{ERM}}) - L(w_*)]}{\mathbb{E}[L(\widehat{w}_N) - L(w_*)]}\,.$$

Can you compete with the ERM on every problem efficiently?

This Talk

Theorem

For linear/logistic regression, generalized linear models, M-estimation, (i.e. assume "strong convexity" + "smoothness"), we provide a streaming algorithm which:

Computationally:

- *single pass; memory is $O(\textit{one sample})$*
- *trivially parallelizable*

Statistically:

- *achieves the statistical rate of the best fit on every problem (even considering constant factors)*
- *(super)-polynomially decreases the initial error*

Related work: Juditsky & Polyak (1992); Dieuleveut & Bach (2014);

Outline

1. Statistics:
   the statistical rate of the ERM
2. Computation:
   optimizing sums of convex functions
3. Computation + Statistics:
   combine ideas

Precisely, what is the error of $\widehat{w}_N^{\mathrm{ERM}}$?

$$\sigma^2 := \frac{1}{2}\mathbb{E}\left[\|\nabla \mathrm{loss}(w_*, p)\|^2_{(\nabla^2 L(w_*))^{-1}}\right]$$

Thm: (e.g. van der Vaart (2000)), Under regularity conditions, *e.g.*

- loss is convex (almost surely)
- loss is smooth (almost surely)
- $\nabla^2 L(w_*)$ exists and is positive definite.

we have,

$$\lim_{N\to\infty} \frac{\mathbb{E}[L(\widehat{w}_N^{\mathrm{ERM}}) - L(w_*)]}{\sigma^2/N} = 1$$

(just) Computation

- optimizing sums of convex functions

$$\min_w L(w) \text{ where } \quad L(w) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{loss}(w, p_i)$$

- Assume:
  - $L(w)$ is μ strongly convex
  - loss is L-smooth
  - $\kappa = \mathrm{L}/\mu$ is the effective condition number

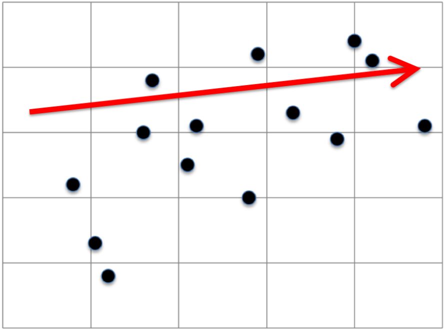
(just) Computation

- optimizing sums of convex functions

$$\min_w L(w) \text{ where } \quad L(w) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{loss}(w, p_i)$$

- Assume:
  - $L(w)$ is μ strongly convex
  - loss is L-smooth
  - $\kappa = \mathrm{L}/\mu$ is the effective condition number
- Stochastic Gradient Descent: (Robbins & Monro, ’51)
- Linear convergence:
  Strohmer & Vershynin (2009), Yu & Nesterov (2010),
  Le Roux, Schmidt, Bach (2012), Shalev-Shwartz & Zhang, (2013),
  (SVRG) Johnson & Zhang (2013)

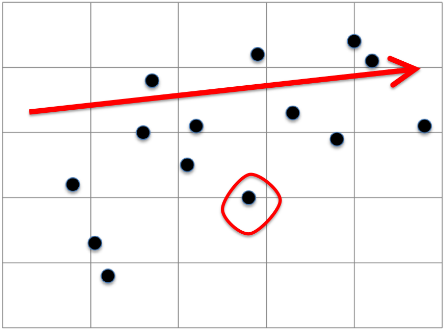
Stochastic Gradient Descent (SGD)



- SGD update rule: at each time t,

$$\text{sample a point } p$$
$$w \leftarrow w - \eta \nabla \mathrm{loss}(w, p)$$

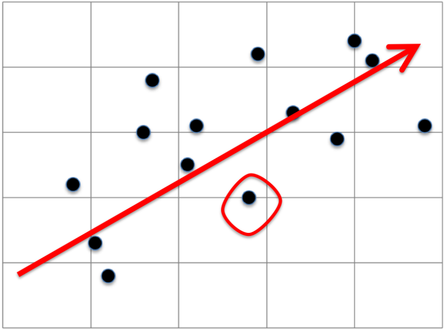
Stochastic Gradient Descent (SGD)



- SGD update rule: at each time t,

$$\text{sample a point } p$$
$$w \leftarrow w - \eta \nabla \text{loss}(w, p)$$

Stochastic Gradient Descent (SGD)



- SGD update rule: at each time t,

$$\text{sample a point } p$$
$$w \leftarrow w - \eta \nabla \text{loss}(w, p)$$

- Problem: even if $w = w_*$, the update changes w.

How do you fix this?

Stochastic Variance Reduced Gradient (SVRG)

1. exact gradient computation: at stage s, using $\widetilde{w}_s$, compute:

$$\nabla L(\widetilde{w}_s) = \frac{1}{N}\sum_{i=1}^{N} \nabla \mathrm{loss}(\widetilde{w}_s, p_i)$$

2. corrected SGD: initialize $w \leftarrow \widetilde{w}_s$. for m steps,

$$\begin{aligned}&\text{sample a point } p\\ &w \leftarrow w - \eta \left(\nabla \mathrm{loss}(w, p) - \nabla \mathrm{loss}(\widetilde{w}_s, p) + \nabla L(\widetilde{w}_s)\right)\end{aligned}$$

3. update and repeat: $\widetilde{w}_{s+1} \leftarrow w$.

Stochastic Variance Reduced Gradient (SVRG)

1. exact gradient computation: at stage s, using $\widetilde{w}_s$, compute:

$$\nabla L(\widetilde{w}_s) = \frac{1}{N}\sum_{i=1}^{N} \nabla \mathrm{loss}(\widetilde{w}_s, p_i)$$

2. corrected SGD: initialize $w \leftarrow \widetilde{w}_s$. for m steps,

$$\text{sample a point } p$$
$$w \leftarrow w - \eta \left(\nabla \mathrm{loss}(w, p) - \nabla \mathrm{loss}(\widetilde{w}_s, p) + \nabla L(\widetilde{w}_s) \right)$$

3. update and repeat: $\widetilde{w}_{s+1} \leftarrow w$.

Two ideas:

- If $\widetilde{w} = w_*$, then no update.
- unbiased updates: blue term is mean 0.

SVRG has linear convergence

- Thm: (Johnson & Zhang, '13) SVRG has linear convergence, for fixed η.

$$\mathbb{E}[L(\widetilde{w}_s) - L(w_*)] \leq e^{-s} \cdot (L(\widetilde{w}_0) - L(w_*))$$

- many recent algorithms with similar guarantees
  Yu & Nesterov '10; Shalev-Shwartz & Zhang '13
- Issues: must store dataset, requires many passes

What about the statistical rate?

Computation + Statistics

Our problem:

$$\min_{w} L(w) \text{ where } \quad L(w) = \mathbb{E}[\mathrm{loss}(w, \mathrm{point})]$$

(Streaming model) We obtain one sample at a time.

Our algo: streaming SVRG

1. estimate the gradient: at stage s, using $\widetilde{w}_s$,

$$\text{with } k_s \text{ fresh samples, estimate } \widehat{\nabla L}(\widetilde{w}_s)$$

2. corrected SGD: initialize $w \leftarrow \widetilde{w}_s$. for m steps:

$$\text{sample a point}$$
$$w \leftarrow w - \eta \left(\nabla \text{loss}(w, p) - \nabla \text{loss}(\widetilde{w}_s, p) + \widehat{\nabla L}(\widetilde{w}_s) \right)$$

3. update and repeat: $\widetilde{w}_{s+1} \leftarrow w$

single pass; memory of $O(\text{one parameter})$; parallelizable

Single Pass Least Squares Estimation

Theorem (Frostig, Ge, Kakade, & Sidford '14)

- *κ effective condition number*
- *choose $p > 2$*
- *schedule: increasing batch size $k_s = 2k_{s-1}$. fixed m and $\eta = \frac{1}{2^p}$.*

If total sample size N is larger than multiple of κ (depends on p), then

$$\mathbb{E}[L(\widehat{w}_N) - L(w_*)] \leq 1.5\frac{\sigma^2}{N} + \frac{L(\widehat{w}_0) - L(w_*)}{\left(\frac{N}{\kappa}\right)^p}$$

σ^2/N is the ERM rate.

- general case: use self-concordance

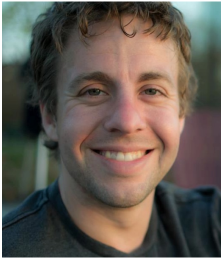
Thanks!

- We can obtain (nearly) the same rate as the ERM in a single pass.

Collaborators:

R. Frostig

R. Ge



A. Sidford