Optimization in the “Big Data” Regime: Averaging and Statistics

Sham M. Kakade

Machine Learning for Big Data
CSE547/STAT548

University of Washington

Tradeoffs in Large Scale Learning.

Tradeoffs in Large Scale Learning.

- Many issues sources of “error”
- approximation error: our choice of a hypothesis class
- estimation error: we only have n samples
- optimization error: computing exact (or near-exact) minimizers can be costly.
- How do we think about these issues?

The true objective

- hypothesis map $x \in \mathcal{X}$ to $y \in \mathcal{Y}$.
- have n training examples $(x_1, y_1), \ldots (x_n, y_n)$ sampled i.i.d. from $\mathcal{D}$.
- Training objective: have a set of parametric predictors $\{h(x, w) : w \in \mathcal{W}\}$,

$$\min_{w \in \mathcal{W}} \hat{L}_n(w) \text{ where } \hat{L}_n(w) = \frac{1}{n} \sum_{i=1}^{n} \text{loss}(h(x_i, w), y_i)$$

- True objective: to generalize to $\mathcal{D}$,

$$\min_{w \in \mathcal{W}} L(w) \text{ where } L(w) = \mathbb{E}_{(X,Y)\sim\mathcal{D}} \text{loss}(h(X, w), Y)$$

Optimization: Can we obtain linear time algorithms to find an ϵ-accurate solution? i.e. find $\hat{h}$ so that

$$L(\hat{w}) - \min_{w \in \mathcal{W}} L(w) \leq \epsilon$$

Definitions

- Let h^* is the *Bayes optimal hypothesis*, over all functions from $\mathcal{X} \to \mathcal{Y}$.
$$h^* \in \text{argmin}_h L(h)$$
- Let w^* is the *best in class hypothesis*
$$w^* \in \text{argmin}_{w \in \mathcal{W}} L(w)$$
- Let w_n be the *empirical risk minimizer:*
$$w_n \in \text{argmin}_{w \in \mathcal{W}} \hat{L}_n(w)$$
- Let $\tilde{w}_n$ be what our algorithm returns.

Loss decomposition

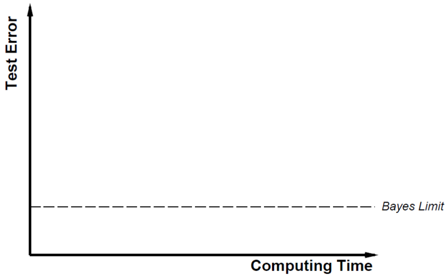
- Observe:

$$
\begin{aligned}
L(\tilde{w}_n) - L(h^*) = \quad & L(w^*) - L(h^*) && \text{Approximation error} \\
& + L(w_n) - L(w^*) && \text{Estimation error} \\
& + L(\tilde{w}_n) - L(w_n) && \text{Optimization error}
\end{aligned}
$$

- Three parts which determine our performance.
- Optimization algorithms with “best” accuracy dependencies on $\hat{L}_n$ may not be best.
  Forcing one error to decrease much faster may be wasteful.

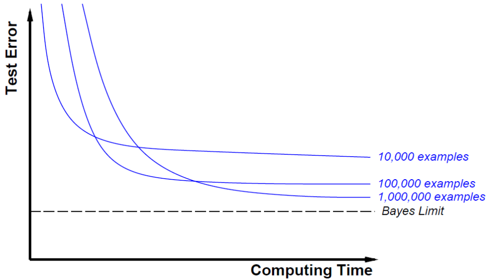
Time to a fixed accuracy

test error versus training time



Comparing sample sizes

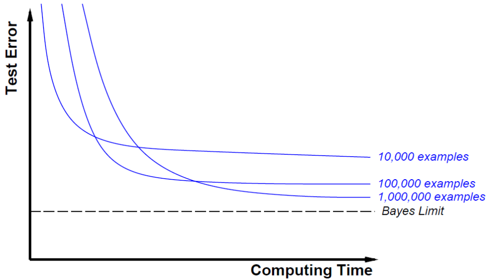
test error versus training time



- Vary the number of examples

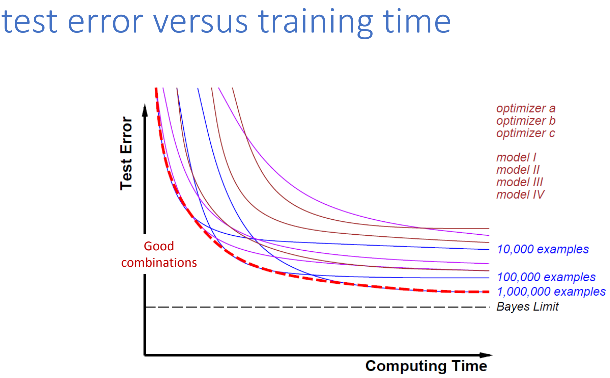
Comparing sample sizes and models

test error versus training time



- Vary the number of examples

Optimal choices



- Optimal combination depends on training time budget.

Estimation error: simplest case

- Measuring a mean:
$$L(\mu) = \mathbb{E}(\mu - y)^2$$
The minima is at $\mu = \mathbb{E}[y]$.
- With n samples, the Bayes optimal estimator is the sample mean: $\hat{\mu}_n = \frac{1}{n}\sum_i y_i$.
- The error is:
$$\mathbb{E}[L(\hat{\mu}_n)] - L(\mathbb{E}[y]) = \frac{\sigma^2}{n}$$
σ^2 is the variance and the expectation is with respect to the n samples.
- How many samples do we need for ϵ error?

Let's compare:

- SGD: Is $O(1/\epsilon)$ reasonable?
- GD: Is $\log 1/eps$ needed?
- SDCA/SVRG: These are also $\log 1/eps$ but much faster than GD (for large n).

Best in class error

- Fix a class $\mathcal{W}$. What is the best estimator of w^* for this model?
- For a wide class of models (linear regression, logistic regression, etc), the ERM, w_n, is (in the limit) the best estimator:

$$w_n \in \operatorname{argmin}_{w \in \mathcal{W}} \hat{L}_n(w)$$

1. What is the generalization error of best estimator w_n?
2. How well can we do? Note:

$$\begin{aligned} L(\tilde{w}_n) - L(w^*) = \quad & + L(w_n) - L(w^*) \quad \text{Estimation error} \\ & + L(\tilde{w}_n) - L(w_n) \quad \text{Optimization error} \end{aligned}$$

- Can we generalize as well as the sample minimizer, w_n? (without computing it exactly)

Statistical Optimality

- Can generalize as well as the sample minimizer, w_n?
(without computing it exactly)
- For a wide class of models (linear regression, logistic regression, etc), we have that the estimation error is:

$$\mathbb{E}[L(w_n)] - L(w^*) \overset{n\to\infty}{=} \frac{\sigma^2_{\mathrm{opt}}}{n}$$

where σ^2_{opt} is an (optimal) problem dependent constant.
- This is the *best* possible statistical rate.
(Can quantify the non-asymptotic "burn-in").
- What is the computational cost of achieving exactly this rate? say for large n?

Averaged SGD

- SGD:

$$w_{t+1} \leftarrow w_t - \eta_t \nabla \mathrm{loss}(h(x, w_t), y)$$

- An (asymptotically) optimal algo:
  - Have η_t go to 0 (sufficiently slowly)
  - (iterate averaging) Maintain the a running average:

$$\overline{w_n} = \frac{1}{n} \sum_{t \le n} w_t$$

  - (Polyak & Juditsky, 1992) for large enough n and with one pass of SGD over the dataset:

$$\mathbb{E}[L(\overline{w_n})] - L(w^*) \overset{n \to \infty}{=} \frac{\sigma^2_{\mathrm{opt}}}{n}$$

Acknowledgements

Some slides from “Large-scale machine learning revisited”, Leon Bottou 2013.