Computational and Statistical Tradeoffs Via Convex Relaxation

Venkat Chandrasekaran, Michael Jordan

Presented by Carrie Wu

June 2, 2017

Statistics vs Computational Sciences

Classical Statistics

- More data leads to better inferences
- More data allows us to invoke asymptotic results in statistical theory
- Very little thought given to the computational feasibility of processing large amounts of data

Computer Science

- Data is a source of complexity (sample complexity)
- Algorithms should use data as sparingly and efficiently as possible

Algorithm Weakening

As data accumulates, one can back off to simpler algorithmic strategies to achieve the same performance.

Computation can be simplified in large datasets because of the enhanced inferential power in the data

The Framework

An algorithm belongs to the *time-data class* $\mathbb{TD}(t(p), n(p), \epsilon(p))$, where p is the dimension of the model class, if:

- runtime upper-bounded by $t(p)$
- number of i.i.d. samples processed bounded by $n(p)$
- achieves a risk bounded by $\epsilon(p)$

The (Additive) Tradeoff

For certain problems, we can divide resource allocation into 2 categories:

- Pre-processing data to create good statistical estimators (ie quantization, dimension reduction, and clustering)
  - Denote sample complexity by $n(p)$
- Optimize on pre-processed data to achieve end goal
  - Denote cost of optimization over convex set c as $f_c(p)$

Total runtime $t(p) = n(p) + f_c(p)$

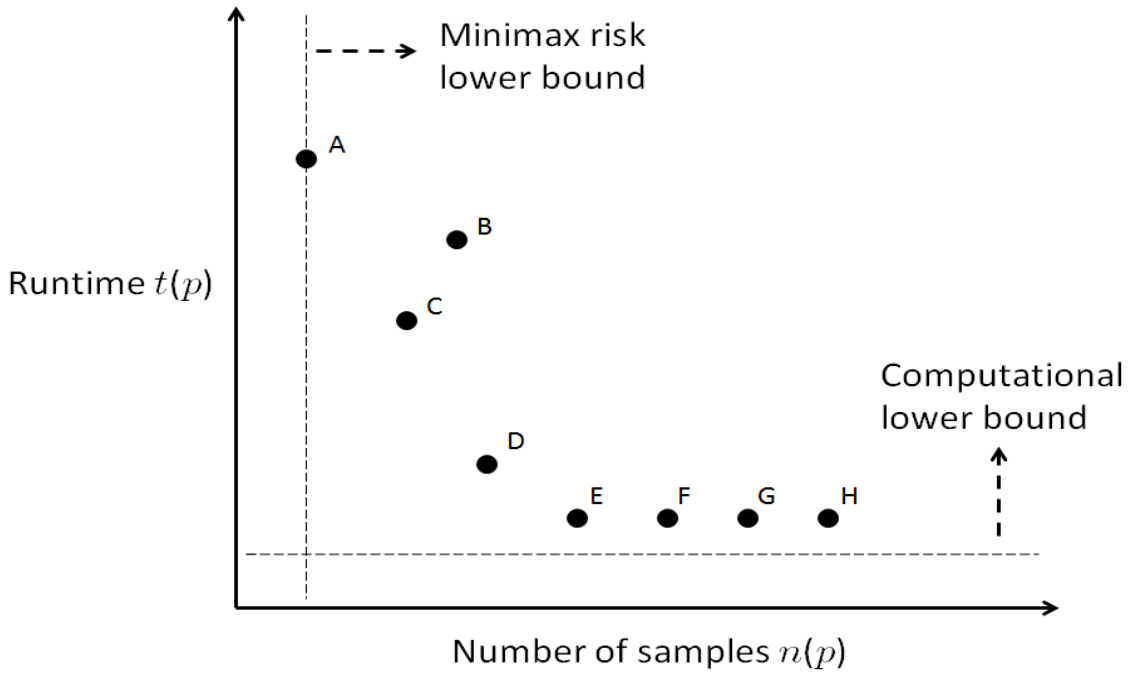

Figure : The tradeoff plot between the runtime and sample complexity in a typical parameter estimation problem. Here the risk is assumed to be fixed to some desired level, and the points in the plot refer to different algorithms that require a certain runtime and a certain number of samples in order to achieve the desired risk. The vertical and horizontal lines refer to lower bounds in sample complexity and in runtime, respectively.

Algorithm Weakening via Convex Relaxation

Consider the following denoising problem:

$$\mathbf{y} = \mathbf{x}^* + \sigma \mathbf{z}, \tag{1}$$

where $\sigma > 0$, the noise vector $\mathbf{z} \in \mathbb{R}^p$ is standard normal, and the unknown parameter $\mathbf{x}^* \in \mathcal{S} \subset \mathbb{R}^p$.

The objective is to estimate $\mathbf{x}^*$ based on n independent observations $\{\mathbf{y}_i\}_{i=1}^n$ of $\mathbf{y}$.

To estimate $\mathbf{x}^*$, consider the natural shrinkage estimator given by a projection of the sample mean $\bar{\mathbf{y}}$ onto a convex set $\mathcal{C}$ that is an outer approximation to $\mathcal{S}$, i.e., $\mathcal{S} \subset \mathcal{C}$:

$$\hat{\mathbf{x}}_n(\mathcal{C}) = \arg\min_{\mathbf{x} \in \mathbb{R}^p} \quad \frac{1}{2} \|\bar{\mathbf{y}} - \mathbf{x}\|_{\ell_2}^2 \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{C}. \tag{2}$$

We can use convex hierarchies to produce these sets:

$$S \subset C \subset C' \subset C'' \ldots$$

Main Result

Theorem (Main Result)

$$\mathbb{E}\left[\|\mathbf{x}^* - \hat{\mathbf{x}}_n(\mathcal{C})\|_{\ell_2}^2\right] \leq \frac{\sigma^2}{n} g(T_{\mathcal{C}}(\mathbf{x}^*) \cap B_{\ell_2}^p).$$

The Gaussian squared-complexity measures the "complexity" or "size" of a tangent cone.

Definition

The *Gaussian squared-complexity* of a set $\mathcal{D} \in \mathbb{R}^p$ is defined as:

$$g(\mathcal{D}) = \mathbb{E}\left[\sup_{\mathbf{a} \in \mathcal{D}} \langle \mathbf{a}, \mathbf{g} \rangle^2\right],$$

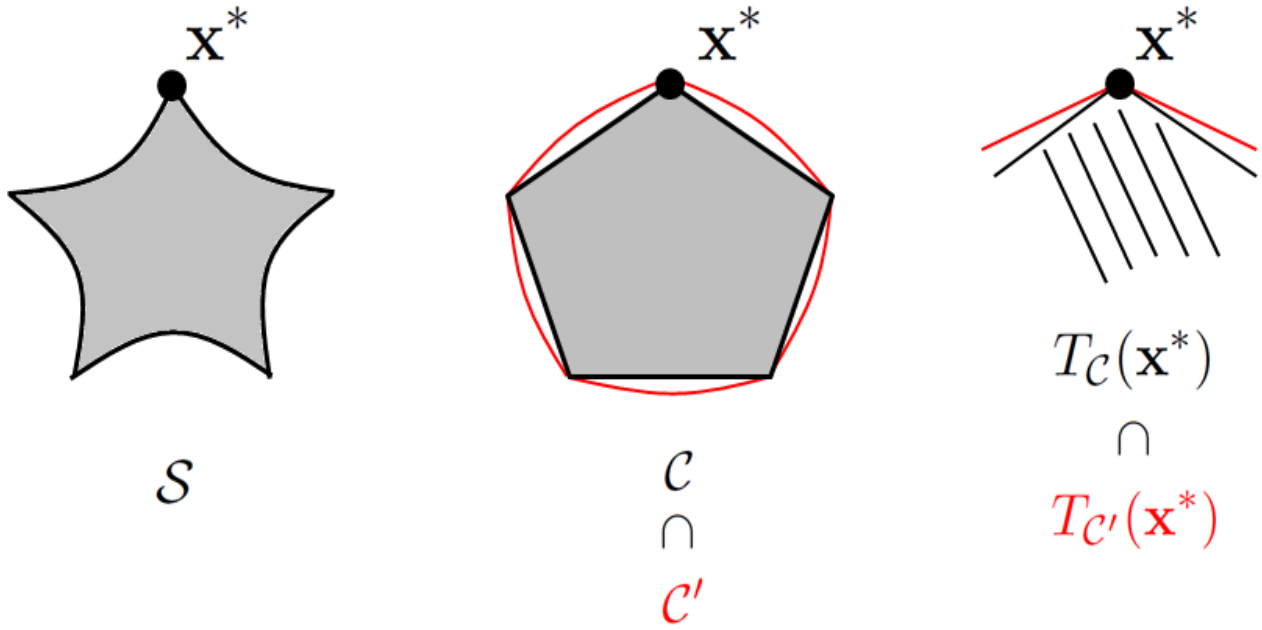

Figure : (left) A signal set $\mathcal{S}$ consisting of $\mathbf{x}^*$; (middle) Two convex constraint sets $\mathcal{C}$ and $\mathcal{C}'$, where $\mathcal{C}$ is the convex hull of $\mathcal{S}$ and $\mathcal{C}'$ is a relaxation that is more efficiently computable than $\mathcal{C}$; (right) The tangent cone $T_{\mathcal{C}}(\mathbf{x}^*)$ is contained inside the tangent cone $T_{\mathcal{C}'}(\mathbf{x}^*)$. Consequently, the Gaussian squared-complexity $g(T_{\mathcal{C}}(\mathbf{x}^*) \cap B^p_{\ell_2})$ is smaller than the complexity $g(T_{\mathcal{C}'}(\mathbf{x}^*) \cap B^p_{\ell_2})$, so that the estimator $\hat{\mathbf{x}}_n(\mathcal{C})$ requires fewer samples than the estimator $\hat{\mathbf{x}}_n(\mathcal{C}')$ for a risk of at most 1.

Conclusions

- Should think about balancing run-time and sample complexity
- Sometimes it is possible to obtain substantial speedups computationally with just a constant factor increase in the size of the dataset.
- Sometimes it is more beneficial to throw away some data and do more intense computations on this smaller dataset, if the time to process a very large dataset starts to heavily dominate the overall runtime.

Questions?