Wilks Lecture

Computational Thinking, Inferential Thinking, and Data Science March 29, 2016

Lecturer: Michael Jordan Scribe: Kiran Vodrahalli

# Contents



# 1 Introduction

Princeton is an institute where optimization, statistics, and computer science have always talked to each other. That's the message of this talk!

Let me say a bit about big data phenomenon. It's more of an industry word. What is it?

## 1.1 What is the Big Data Phenomenon?

First it came in particle physics: They needed a huge amount of data to figure out whether the Higgs boson existed. It's because the models were so similar, and noise levels were hugely amplified. So to get a reliable test, you needed a huage amount of data. This was just hypothesis testing!

The next way was genomics and astronomy. Here the inference problem wasn't hypothesis testing, it was exploratory. Gather as much data you could that could drive your science. There were tons of possible hypotheses; let's find those which are worth studying. Inferentially this was very different, worry about FDR and so on at high scale.

Finally was industry. All the Silicon Valley companies started talking about data. Well, personalization was really important. They could offer personalized services to most people. This is the measurement of human activity, particularly online activity.

So how do you make inferences at hard scale with good granularity.

There were inferential issues, but also computational issues. Let's talk about that.

Here's a job description circa 2015. Boss: I need you to build a Big Data system that will replace our classical service with a personalized service. It should work reasonably well for anyone and everyone, I can tolerate only a few errors but not too many dumb ones. Not the classical statistician sort of thing, 99% isn't good enough. You have to evaluate with $\ell_\infty$ norm instead. That's just the statistical side! It needs to run also as fast as our classical service. Google did this in 2007. If this didn't happen, their business would have died. So it's got to run just as fast. Now it's really hard. Before we had only one model, now we have hundreds of thousands of personalized models!

Furthermore, it should only improve as we collect more data; in particular, things shouldn't slow down. More data means even more computational difficulty. All the algorithms go up at rate $n$! Industry does adhoc things to deal with this (subsample, throw away some of the data: but that might hurt you; what's the right rate?). Parallel distribution etc is just hoping things will speed up. Let's be serious about engineering science. There are also serious privacy concerns of course, and they vary across clients.

This is the problem of our age. Now we have to build systems that bring all these concerns together.

So Big Data analysis requires a thorough blend of computational thinking and inferential thinking. These are distinct styles of thinking and we have to respect both of them. Computational thinking means

- abstraction
- modularity
- scalability
- robustness

Inferential thinking is considering the real world phenomena behind the data, the sampling pattern, developing procedures that will go backwards from the data to the underlying phenomena. Merely computing statistics or running machine learning algorithms isn't inferential thinking. There is too much black box principles here.

The challengeies are daunting: Core statistical theory never mentions the term runtime (the whole field of computer science). Core computational theory doesn't have a place for statistical risk. We would like to talk about these things in a theoretical sense.

### 1.2 Outline

- Inference under privacy constraints
- Inference under communication constraints
- The variational perspective
- Nesterov Acceleration (just finished a paper on this)

## 2 Inference and Privacy and Data Analysis

Work with Wainwright and Duchi.

Some people hold data privately: for instance, your genome. Sometimes you may want to give up this data, sometimes you don't want to. You want to have a knob that you can turn, and this should be sensitive to the loss function: how much privacy loss can occur? This is like differential privacy.

If you're a database person, and they've collected a whole bunch of data about you, you put this query in the database. Let's call that answer $\tilde{\theta}$. That might be the maximum amount of money held by someone, or the variance, or whatever. You may want to protect identities of the people there. We will talk about differential privacy, which privatizes the database, by putting it through a stochastic operator $Q$ which perturbs the entries of the database. You get out a $\hat{\theta}$ after you have the privatized databases. Then you can say with differential privacy that $\hat{\theta} \approx \tilde{\theta}$. This is computational, how about inferential?

What if you want to say things about where the data came from? There is some population, sampled behind the database. Now you put a query into the population, and get an answer for $\theta$. Then statistical theory says over all sampling procedures and populations, $\theta \approx \tilde{\theta}$. These are two different styles of thinking. So now we want to prove that $\theta \approx \hat{\theta}$.

What's the core quantity we need to discuss? The risk. We have a family of distributions $\mathcal{P}$, a parameter $\theta(P)$ for each $P \in \mathcal{P}$, an estimator $\hat{\theta}$ and a loss $l$.

$$R_P(\hat{\theta}) := \mathbf{E}_P[l(\hat{\theta}, \theta(P))]$$

This number is a function of $P$. Wald gave the minimax principle: Let's take the worst case over all distributions:

$$\sup_{P\in\mathcal{P}}\mathbf{E}_P[l(\hat{\theta}, \theta(P))]$$

If you're a Bayesian, there's a robust interpretation. Let's bring this together with privacy. Differential privacy says that channel $Q$ is $\alpha$-differentially private if

$$\sup_{S,x\in X,x'\in X}\frac{Q(Z \in S|x)}{Q(Z \in S|x')} \leq \exp(\alpha)$$

where$x, x'$ are two different situations. You could look at over all sets $S$. If the ratio is close, then you can't tell the difference. Essentially, we have $X_i \to Q(\cdot|X_i) \to Z_i \to \hat{\theta}$, where $Z$ is our "noisy version". How do we talk about how we don't deal with statistical inference.

Now we have a family of channels which protect privacy at level $\alpha$. Then, the $\alpha$-private minimax risk is

**Definition 2.1.** $\alpha$-private minimax risk.

$$\inf_{Q\in Q_\alpha}\inf_{\hat{\theta}}\sup_{P\in\mathcal{P}}\mathbf{E}_{P,Q}[l(\hat{\theta}(Z_1^n), \theta(P))]$$

Can we get our math from before, in addition to our new parameter $\alpha$?

For instance, we want to estimate our mean $\theta(P) := \mathbf{E}_P[X] \in \mathbb{R}^d$ with errors measured in $\ell_\infty$ norm for $P_d$ which are distributions supported on $[-1, 1]^d$. For private minimax rate for $\alpha = O(1)$, we have the sample rate is

$$M_n(P_d, \|\cdot\|_\infty, \alpha) \asymp \min\{1, \frac{\sqrt{\frac{d\log d}{n\alpha^2}}}{\}}$$

Usually (for the usual definition), the bottom fraction is just $n$. It turns out that this shift in sample size reduction is generic when you do differential privacy: $n \to n\alpha^2/d$.

# 3 Inference and Compression

This is now Shannon Land, compression land.

What if you have huge amounts of data, separated geographically. How do I compress the data? We have $X_i \to Z_i \to \hat{\theta}$, where $X_i$ is the original data, and $Z_i$ is the actual message. Shannon solved it where the objective is the length of the codewords. What about when the loss function is instead statistical risk?

Now you have a new constraint: the bit rate $B$: Can't send more than $B$ bits in $Z_i$. Here we have minimax risk with $B$-bounded communication:

$$M_n(\theta(P), B) := \inf_{\pi\in\Pi_B}\inf_{\hat{\theta}}\sup_{P\in\mathcal{P}}\mathbf{E}_{P,B}[\|\hat{\theta}(Z_1^n) - \theta(P)\|_2]$$

We get a theorem which gives for $X_j \sim \mathcal{N}(\theta, \sigma^2 I_{d\times d})$.

$$\frac{d}{B \vee d} \frac{1}{\log m} \frac{\sigma^2 d}{nm} < M_n(N_d, B) < \frac{d \log m}{B \vee d} \frac{\sigma^2 d}{mn}$$

as the minimax rate with $B$-bounded communication, when each agent has a sample of size $n$.

# 4 Computation and Inference

Previously we say privacy and communication, memory together with inference. How do we put computation together with inference?

This is one of the great problems of our time. It is hard to get Turing theory to help you that much here; that's worse case lower bounds. Princeton is a hot bed for these kinds of issues.

In my group we have several ongoing groups. One early thing was tradeoffs via convex relaxations (optimization really makes its appearance here). One simple model of computation is minimize functions. Our resources will be like number of times you can see the gradient of a function. How close can you get? This is the problem from Nesterov and so on; it's a model of computation. Now how does that interact with statistical concerns.

In this paper, we tried to put geometry inbetween computation and statistics. A breakthrough on geometry relates to computation. This was only for denoising problems.

Another thing I'm enthusiastic about is concurrency control: How do you think about updating things asynchronously; how do we think about concurrency control. Bag of bootstraps (subsampling paradigm) is another thing.

What I will talk about today is something variational: A variational framework for accelerated methods in optimization with Andrew Wibisono and Ashia Wilson, who have been working on this for $2-3$ years.

Variational is my favorite word in applied mathematics. Think about physics, where this word got used. Newton had done his differential equations; they were the input to the solution. The computation became integration. A hundred years later, Lagrange came along and said you could do the same physics via optimization. I'll write down an action and get out a curve by optimizing, not doing differential equation. Get out exactly the same physics.

In statistics, everything has been based on sampling and integrals; Bayesian stats is about integrals. Very little is about calculus of variations problem. So many procedures involving relaxations involve this variational procedure. You instead solve the relaxed problem. How do you move around in the optimization? Well you can change the problem a bit, and the solution changes a bit. Every time I get into an area, I wonder if there's a variational version of the area.

This will be about gradient based optimization. You know that today it's all about getting downhill cheaply on large-scale spaces. How fast can you get downhill using gradients? Are there better ways to get downhill faster? Are there lower bounds?

Nesterov and Nemirovski found ways to get down faster with lower bounds. Our setting is unconstrained convex optimization: $\min_{x\in\mathbb{R}^d} f(x)$. The classical way to do this is gradient descent. Classical gradient descent is $\mathcal{O}(1/T)$. Is that the best you can do? It turns out there's a better algorithm: Two sequences where you have two updates, with a convex combination of the updates:

$$y_{k+1} = x_k - \beta \nabla f(x_k)$$

$$x_{k+1} = (1-\lambda_k) y_{k+1} + \lambda_k y_k$$

In fact you can get convergence rate $\mathcal{O}(1/T^2)$, and that's optimal. You can also accelerate Newton, conic optimizations, etc.

Usually there's some specific algebra and not too much intuition. There's not a general theory. Accelerated methods are not descent methods, they oscillate a bit. Moritz Hardt and Zeyuan Allen-Zhu have only strongly convex functions, or first-order methods.

We'll give a Lagrangian perspective (variational) on this. A lot of the mess in this is because we deal with discrete time. We have to use continuous time. Gradient descent is discretization of gradient flow: $X_t^{\cdot} = -\nabla f(X_t)$. And mirror descent is discretization of natural gradient flow. Su, Boyd, Candes at Stanford wrote down a second order ODE:

$$X_t^{\cdot\cdot} + \frac{3}{t} X_t^{\cdot} + \nabla f(X_t) = 0$$

Ah, this must come from some action! We will show there is a variational approach to acceleration. Critically, there is a systematic way to discretize these. This is the complicated part. These are unusual differential equations.

Define the Bregman Lagrangian:

$$L(x, x^{\cdot}, t) = e^{\gamma_t + \alpha_t} \left( D_h(x + e^{-\alpha_t} x^{\cdot}, x) - e^{\beta_t} f(x) \right)$$

where

$$D_h(y, x) = h(y) - h(x) - \langle \nabla h(x), y - x \rangle$$

is the Bregman divergence. If $h$ is the $l_2$ norm (convex distance generating function), then it looks like physics again (maybe some energy), $f$ is the convex objective function.

The message, you generate Nesterov acceleration and this is an interesting object to study.

The Bregman Lagrangian is

$$\min_X \int L(X_t, X_t^{\cdot}, t) dt$$

and the optimal curve is characterized by Euler-Lagrange:

$$(d/dt)\{dL/dx^{\cdot}(X_t, X_t^{\cdot,t})\} = \frac{\partial L}{\partial x}(X_t, X_t^{\cdot,t})$$

You can plug in the Bregman Lagrangian to get a differential equation.

**Theorem 4.1.** *Under ideal scaling, you have convergence rate*

$$f(X_t) - f(x^*) \leq \mathcal{O}(e^{-\beta_t})$$

You get polynomial convergence rates by choosing $\alpha_t = \log p - \log t, \beta_t = p \log t + \log C, \gamma_t = p \log t$. In the Euclidean case, you recover the ODE of Su et al.

## 4.1 Discretization

Now, how do we discretize Euler-Lagrange to preserve convergence rate?

You can write a system of equations, and do the forward Euler method. This doesn't work.

Is there a better way? You can introduce an auxillary sequence, like Nesterov, and then it works. You need a condition involving the dual norm of a gradient to get it to be stable. Then you can get $f(y_k) - f(x^*) \leq \mathcal{O}(1/\epsilon k^p)$.

You can also do higher-order gradient updates using higher order Taylor approximations of $f$, and get accelerated algorithms. You can get this at any order $p$. This is a generalization of Nesterov.

Bregman Lagrangian also has interesting properties (go to physics). You can even define a Hamiltonian version of this. Look at structure and properties of Bregman Lagrangian: Gauge invariance, symmetry, gradient flows as limit points, etc.

A lot of the interesting problems to me are theoretical, not empirical (all the stuff you read about is empirical). This gives us brand new kinds of theory. Some of the people in the past like von Neumann, etc. They didn't think of themselves as any one of stats, computer science, or optimization. They were all of the above! I left econ out by the way, it's also a major player. The intellectual problems of our day are really theory ones. Note that this is a local theory! So it's applicable to nonconvex optimization.

You can often generate a lot of interesting results by considering Lyapunov geometry.