Statistics of Big Data, Fall 2021-22

Class notes 1

What is Big Data?

In the traditional view a statistical modeling or estimation problem has n observations and p parameters that have to be estimated based on this data, for example:

- Estimating mean based on a random sample from a population: n i.i.d observations, $p = 1$ parameter
- Linear regression model: n pairs $(X_1, Y_1), \ldots, (X_n, Y_n)$ with $X \in \mathbb{R}^p$, drawn i.i.d from $Pr(x, y)$, we assume
$$\mathbb{E}(Y|X) = X^T\beta + \beta_0.$$
This model has $p+1$ parameters $\beta \in \mathbb{R}^p, \beta_0 \in \mathbb{R}$, we can estimate them from the data getting $\hat{\beta}, \hat{\beta}_0$.
Given new observation X we predict it $\hat{Y}(X) = X^T\hat{\beta} + \hat{\beta}_0$.

In more complex modeling situations (such as network modeling we will discuss later in the semester) it is not always clear what are n and p, but we will try to maintain this view: if n or p are big we have a big data problem.

Tall vs Wide data and the components of error

Keeping the generic view above, we can think of two types of big data:

- **Tall data** where n is big and p is small: we have lots of data to estimate a low dimensional object. These problems can be hard in terms of computation, storage etc., leading to solutions like Map-Reduce and Spark. However they are usually "statistically easy", meaning we have enough information to estimate the models well (in the extreme, we have the entire population and so "estimation" is simply calculation).
- **Wide data** where p is big, and n is not huge. Such problems are often very hard statistically, since we may not have enough information to estimate the model well. This means that in this setting, in addition to potential computational difficulties, we have to deal with the statistical difficulties. This is often addressed by dimensionality reduction (reducing p) or adding regularization (simplifying the problem statistically without explicitly reducing p). These are topics we will discuss quite extensively.

The distinction between computational difficulties in big data (which are not the main focus of our course, though we will touch on some aspects of them) and statistical difficulties (which are our main focus) will be central to many of the topics we discuss.

Every modeling or estimation problem can be thought of as combining two elements of statistical difficulty:

- **Approximation error**: is our approach and model appropriate for the problem at hand? In the context of linear regression this can be thought of as the question whether the hypothesized model $\mathbb{E}(Y|X) = X^T\beta$ is in fact correct? If not, how far away is the true mean from being a linear function? If the true model is far from linear but we insist on fitting a linear model, we are bound to suffer *bias* or more generally *approximation error* in applying our model to the world, for example for predicting new observations.

- **Estimation error**: Do we have enough data for estimating the p parameters in our model? In the linear regression approach, even if we assume the model $\mathbb{E}(Y|X) = X^T\beta$ is correct, it does not mean that our estimates $\hat{\beta}$ will be close to the true β, and this may be because n is not big and/or because the measurement errors $\epsilon_i = Y_i - X_i^T\beta$ in the data are big. This inaccuracy due to data finiteness and noise is usually referred to in the context of linear regression as *variance* or more generally *estimation error*.

A very generic view of the behavior of these components in big data problems is:

- In tall data (big n, small p) the estimation error is usually small, while the approximation error will often be large (unless we are "lucky" in "guessing" a good model with few parameters).

- In wide data the estimation error usually dominates, as the large number of parameters allows flexible models that can reduce approximation error.

Generic rules of thumb following this discussion:

- In wide / high dimensional problems (big p) — critical to control estimation error, otherwise no useful models can be built.

- Modern methods in machine learning, predictive modeling, big data modeling aim to combine flexibility and high dimension (controlling approximation error) with regularization (controlling estimation error) in diverse and creative ways.

Big Data in action: Some Google stories

Google is a company whose initial success and rise to prominence (and arguably also its ongoing dominance) was primarily fueled by solving a big data problem of how to effectively search the web. Their *PageRank* algorithm was essentially a new statistical approach to this problem, as we will discuss. Over the years Google became a producer of huge data and has also carried out many big data modeling projects on this data, some for internal purposes and others for "public good". We will critically review one such project and as time permits also discuss the original PageRank story.

Google Flu Trends

In the time when the worst pandemic in public view was the yearly Flu outbreak, the main metric used to evaluate the Flu situation was the percentage of doctor visits related to *Influenza-like illness* or *ILI*. In usual times it is 1%-2% while during outbreaks it can rise to 10%. The Center for Disease Control (CDC) traditionally reports weekly ILI in about two weeks delay (the time it takes to collect and process the information).

Around 2009 Google had the cool idea that their big data on search can be used to estimate the ILI in real time (or even ahead of time) based on what people search for — for example, searching for Flu symptoms when self-diagnosing. This can allow obtaining information on Flu outbreaks earlier and taking the needed steps, so potentially is very important for public health policy (again, this became clearer during the COVID outbreak, where information is needed to guide public policy). They applied their approach to create a system called Google Flu Trends (GFT) and described it in a paper published in 2009 in the journal Nature (one of the two most important scientific journals).

The specific approach GFT took:

1. Collected CDC information on ILI from 128 weeks between $9/2003 - 3/2007$ in 9 regions of the US, as a training data-set of size 128×9. 42 additional weeks until 5/2008 were kept as a test set for final evaluation of their model.

2. They identified the 50 million most common search terms on Google during that period. There was no processing, so for example "flu cough" is a different term than "cough flu". Their actual data is the % of all searches that searched each term in each week (tiny numbers on order of $1/(5 \times 10^7)$.). Their training set is thus with $n = 128$ and $p = 5 \times 10^7$, a very wide problem (unclear whether the 9 regions are part of n or p...).

3. On the training data, they ranked the terms according to their "association" with ILI, as follows:

   (a) For each word, they divided the 128 weeks into 4 groups (folds) of 32 each, remember they also have 9 regions. Denote q_{ij} the vector of length 32 of % searches for this word in fold $i = 1, \ldots, 4$, region $j = 1 \ldots, 9$.

   (b) They performed 4-fold cross validation for each region separately:

      i. Denote for fold $i = 1, \ldots, 4,\ j = 1, \ldots, 9$:

      $$Y_{ij} = \text{logit}(ILI_{ij}) \in \mathbb{R}^{32}\ Q_{ij} = \text{logit}(q_{ij}) \in \mathbb{R}^{32}.$$

      Reminder: $\text{logit}(p) = \frac{\log(p)}{\log(1-p)}$ for $p \in [0, 1]$.

      ii. Assuming the model $Y = \alpha Q + \epsilon$ perform 4-fold cross-validated linear regression:

      $$\begin{aligned}&\text{Training linear regression model: } Y_{-i,j} \approx \hat{\alpha}_{ij} Q_{-i,j}\\ &\text{Holdout prediction: } \hat{Y}_{i,j} = \hat{\alpha}_{ij} Q_{i,j}\\ &\text{Evaluation: } r_{ij} = \text{cor}(Y_{ij}, \hat{Y}_{ij})\end{aligned}$$

      iii. This gives $4 \times 9 = 36$ numbers r_{ij}, calculate Fisher's transformation (why?):

      $$Z_{ij} = 0.5 \log\left(\frac{1 + r_{ij}}{1 - r_{ij}}\right),$$

and the final evaluation of how good this word is for "predicting" current ILI is:

$$\bar{Z} = \frac{\sum_{ij} Z_{ij}}{36}.$$

(c) Overall this requires estimating $5 \times 10^7 \times 36 \approx 2 \times 10^9$ linear regression models and correlations, but each one is pretty easy to calculate.

(d) This yields a ranked list of search terms: $\bar{Z}^{(1)} \geq \bar{Z}^{(2)} \geq \ldots \geq \bar{Z}^{(5\times 10^7)}$.

4. Now they define:

$$X_j^{(K)} = \text{logit}\left(\sum_{k=1}^{K} q_j^{(k)}\right),$$

the logit of the % of the queries in the top-ranked K out of all words.

5. They seek to find $\hat{K}$, the best number of terms to include in their model:

(a) For every K fit a single regression model on the 128×9 points in the training set: $Y \approx \hat{\alpha}^{(K)} X^{(K)}$.

(b) Select $\hat{K}$ as maximizing the correlation on the 42×9 observations on the test set (not yet used!):

$$\hat{K} = \arg\max_K \text{cor}\left(Y_{test}, \hat{\alpha}^{(k)} X_{test}^{(K)}\right).$$

This leads to $\hat{K} = 45$ as shown in Figure 1 of their paper.

6. In Figures 2,3 they demonstrate the prediction success of their model within the evaluation period (Fig. 2) and in real time prediction (Fig. 3).

The problems with their solution can be divided into several categories:

1. Conceptual and general approach issues, e.g.:

- Is this approach likely to work in the long term? Why or why not?
- Is it properly dealing with the huge dimensionality and the bias-variance issues that come with it?
- Where are comparisons to simpler standard benchmark approaches?

2. Specific statistical problems related to implementing the approach they chose, e.g.:

- How they perform and cross-validate their univariate models in the first stage?
- How they combine them using Fisher Z scores
- How they evaluate the overall model and choose $\hat{K}$

In the homework you will deal with the second category first, then consider also the first category. For this you may find it useful to read and think about the later paper published in 2014 in the journal Science (the other top scientific journal) and severely criticizing their approach and the validity of their results.

The Science paper reports that starting 1/2011 GFT predicted ILI too high in 100 out of 108 weeks, with a peak of two-fold too big (meaning GFT says there's a major Flu outbreak, while in fact there is regular Flu season). This paper insults the GFT paper extensively, though in my view it also suffers from being quite vague and not mathematically concrete. For example it uses the term "algorithm dynamics" as a major flaw without properly defining what it means. It does point out a concrete problem in not comparing to standard auto-regression models which they claim perform just as well. Your task in the second item of the homework is to figure out what they are trying to say and rephrase it in more accurate and concrete terms.

PageRank

In 1998 Google founders came up with a "new" algorithm for ranking importance of internet pages that revolutionized search.

Think of the internet as a graph, with directed edges being the links between sites. In this view, it is clear that important pages are ones that are linked by either other important pages or many other pages. The PageRank approach can be thought of as considering the internet as a stochastic process (random walk) where all outgoing links from a site have an equal probability/rate. Typically it is viewed in discrete time, where at each time point, given the current location with k outgoing links, we have a probability of $(1-\delta)/k$ of moving to each one, and δ of moving to a randomly chosen site on the internet. See illustration here.

For a three-node network with outgoing links from 1 to 2 and 3, and from 2 to 3, this gives transition matrix:

$$\mathbf{M} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \left\| \begin{array}{ccc} \frac{\delta}{3} & \frac{\delta}{3} & \frac{1}{3} \\ \frac{1-\delta}{2} + \frac{\delta}{3} & \frac{\delta}{3} & \frac{1}{3} \\ \frac{1-\delta}{2} + \frac{\delta}{3} & (1-\delta) + \frac{\delta}{3} & \frac{1}{3} \end{array} \right\| \end{array}$$

The stationary distribution is a probability vector $R \in \Delta^3$ such that: $R = MR$, which always exists since M is a left stochastic matrix. In this low dimension we can easily find it analytically, for example with $\delta = 0.1$ we get $R = (0.19, 0.28, 0.53)^t$.

On the internet scale it is typical to run an iterative algorithm essentially emulating the Markov process and approximate R, and this is the ranking the original PageRank used to revolutionize search...