ISyE 6416: Computational Statistics
Spring 2017

Lecture 1: Introduction

Prof. Yao Xie

H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

What this course is about

- Interface between **statistics** and **computer science**
- Closely related to **machine learning**, **data mining**, and **data analytics**
- Aim at the design of algorithm for implementing statistical methods on computers

Major components

- Optimization tools for statistics
  - First order and second order methods for likelihood
  - Expectation-maximization methods
- Parametric methods
  - Gaussian mixture model (GMM)
  - Hidden Markov model (HMM)
  - Model selection and cross validation
- Non-parametric methods
  - Principle component analysis and low-rank models
  - splines and approximation of functions
  - Bootstrap and resampling
  - Monte Carlo methods

Statistics

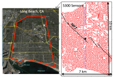
data: images, video, audio, text, etc. sensor networks, social networks, internet, genome.

statistics provide tools to

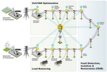
- **model** data
  e.g. distributions, Gaussian mixture models, hidden Markov models
- **formulate** problems or ask questions
  e.g. maximum likelihood, Bayesian methods, point estimators, hypothesis tests, how to design experiments

physical sensors

Engine prognostics



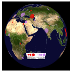
Geophysical, environmental sensor array



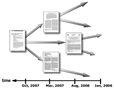
Power system

social "sensors"

Social networks



GDELT event streams



Citation networks

Statistics needs computing

- once the problem has been formulated, we have to solve and problem and this relies on **computing**
- the forms of the mathematical problem does not relate to how to solve it
- computing: find efficient algorithms to solve them
  e.g. maximum likelihood requires finding maximum of a cost function
- “Before there were computers, there were **algorithms**. But now that there are computers, there are even more algorithms, and algorithms lie at the heart of computing. ”

Algorithm

(loosely speaking) a method or a set of instructions for doing something...
A program is a set of computer instructions that implement the algorithm.

computational statistics vs. optimization

- choosing decision parameter value to minimize the decision risk

Example: linear regression

(x_i, y_i), $i = 1, \ldots, n$.
Risk function: $R(a, b) = \sum_{i=1}^{n}(y_i - (ax_i + b)^2$

$$(\hat{a}, \hat{b}) = \arg\min_{a,b} R(a, b)$$

- choosing parameter value according to maximum likelihood

Example: maximum likelihood

- θ: parameter, x: data
- log-likelihood function $\ell(\theta|x) \triangleq \log f(x|\theta)$

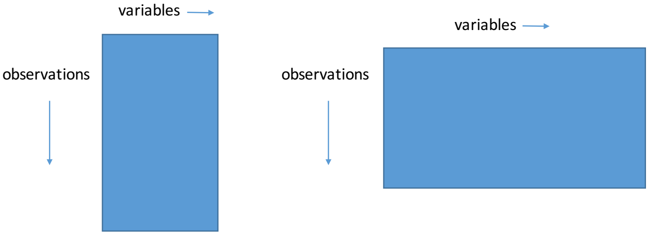
$$\hat{\theta}_{\mathrm{ML}} = \arg\max_{\theta} \ell(\theta|x)$$

- drop dependence on x, but remember that $\ell(\theta)$ is a function of data x
- Simplest setting: maximize the log-likelihood function by setting $\frac{d\ell(\theta)}{d\theta} = 0$

How to find a solution to the optimization problem? Is there is a global solution, or there are many local solutions?

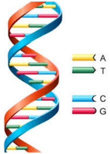
computational statistics vs. linear algebra

- A common data structure for statistical analysis is the rectangular array: a matrix
- the property of the matrix says a lot about the structure of the data



Common statistics..

High-dimensional statistics



How to solve large linear systems

$$y = Ax$$

- linear regression: A data matrix; y vector of response variables, we need to solve $(A^\intercal A)^{-1} A^\intercal y$
- directly compute matrix inverse may not be practical
- needs various regularization to obtain good solution

Example: big data challenge

The Human Genome Project has made great progress toward the goals of identifying all the 100,000 genes in human DNA. With 10 patients, A is of size 10 by 100,000.

TABLE 1.1

Comparison Between Traditional Statistics and Computational Statistics [Wegman, 1988]. Reprinted with permission from the *Journal of the Washington Academy of Sciences.*

Traditional Statistics	Computational Statistics
Small to moderate sample size	Large to very large sample size
Independent, identically distributed data sets	Nonhomogeneous data sets
One or low dimensional	High dimensional
Manually computational	Computationally intensive
Mathematically tractable	Numerically tractable
Well focused questions	Imprecise questions
Strong unverifiable assumptions: Relationships (linearity, additivity) Error structures (normality)	Weak or no assumptions: Relationships (nonlinearity) Error structures (distribution free)
Statistical inference	Structural inference
Predominantly closed form algorithms	Iterative algorithms possible
Statistical optimality	Statistical robustness

Statistics needs computing - II

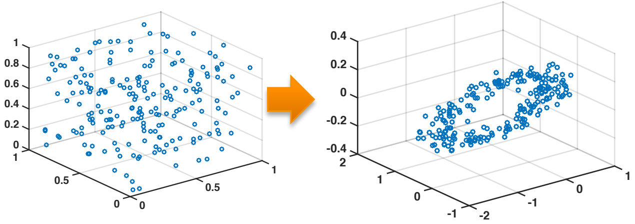
- many realistic models are not as mathematically tractable, we may use computationally intensive methods involving simulation, resampling of data etc.

Example

simple Bayesian inference
$x \sim \mathcal{N}(\mu, \sigma^2)$, $\mu \sim \mathcal{N}(\theta, \tau^2)$
posterior distribution $\mu | x \sim \mathcal{N}(\frac{\tau^2}{\tau^2+\sigma^2}x + \frac{\sigma^2}{\sigma^2+\tau^2}\theta, \frac{\sigma^2\tau^2}{\sigma^2+\tau^2})$
But in other case
$x \sim \mathcal{N}(\mu, \sigma^2)$, $\mu \sim$ Unif$[0, 1]$, posterior distribution $\mu | x$ is not any known distribution

Statistics needs computing - III

- to discover structure in the data: gaps, gaps, clusters, principle components, rank, linear relationship between variables, etc.



full rank $\Rightarrow$ rank ≈ 2

Example: Netflix Problem

- Netflix database: About 1,000,000 users and $25,000$ movies
- Quantized moving ratings (e.g, 1,2,3,4,5)
- Observe a subset of entries (sparsely sampled)

Guess the missing ratings?

observed

$$\begin{array}{cc} & \text{users} \\ \text{movie} & \begin{bmatrix} 3 & & 1 & 5 \\ & 1 & & \\ 2 & & & \\ & & 5 & 3 \end{bmatrix} \end{array}$$

true preference

$$\begin{array}{cc} & \text{users} \\ \text{movie} & \begin{bmatrix} 3.5 & & 1.3 & 4.43 \\ & 1.01 & & \\ 2.1 & & & \\ & & 4.9 & 3.5 \end{bmatrix} \end{array}$$

$$\begin{bmatrix} \times & ? & ? & ? & \times & ? \\ ? & ? & \times & \times & ? & ? \\ \times & ? & ? & \times & ? & ? \\ ? & ? & \times & ? & ? & \times \\ \times & ? & ? & ? & ? & ? \\ ? & ? & \times & \times & ? & ? \end{bmatrix}$$

Regularized maximum-likelihood estimator

- log-likelihood function for categorical matrix completion

$$F_{\Omega,Y}(X) \triangleq \sum_{(i,j)\in\Omega} \sum_{k=1}^{K} \mathbb{I}_{[Y_{ij}=a_k]} \log(f_k(X_{ij})).$$

- Nuclear norm regularization likelihood function

$$\widehat{M} = \arg\max_{X\in\mathcal{S}} F_{\Omega,Y}(X),$$

$$\mathcal{S} \triangleq \Big\{ X \in \mathbb{R}_{+}^{d_1\times d_2} : \|X\|_* \leq \alpha\sqrt{r d_1 d_2}, \\ -\alpha \leq X_{ij} \leq \alpha, \forall (i,j) \in [d_1]\times[d_2] \Big\},$$

Optimization problem

- non-convex optimization problem

$$\min_{M \in \Gamma} f(M) + \lambda \|M\|_*$$

matrix completion $f(M) = -\sum_{(ij)\in\Omega} \log p(Y_{ij}|M_{ij})$
Γ: set of feasible estimators

exact algorithm:	approximate algorithm:
Semidefinite program (SDP)	**singular value thresholding**
$\mathcal{O}(d^4)$	$\mathcal{O}(d^3)$

Algorithm 1 PMLSVT for Poisson matrix recovery and completion

1: Initialize: The maximum number of iterations K, parameters α, β, η, and t.
$X \leftarrow \mathcal{P}(\sum_{i=1}^{m} y_i A_i)$ {matrix recovery}
$[X]_{ij} \leftarrow Y_{ij}$ for $(i, j) \in \Omega$ and $[X]_{ij} \leftarrow (\alpha+\beta)/2$ otherwise {matrix completion}
2: **for** $k = 1, 2, \ldots K$ **do**
3: $\quad C \leftarrow X - (1/t)\nabla f(X)$
4: $\quad C = U\Sigma V^{\intercal}$ {singular value decomposition}
5: $\quad [\Sigma]_{ii} \leftarrow ([\Sigma]_{ii} - \lambda/t)^{+}$, $i = 1, \ldots, d$
6: $\quad X' \leftarrow X$ {record previous step}
7: $\quad X \leftarrow \mathcal{P}\left(U\Sigma V^{\intercal}\right)$ {matrix recovery}
$\quad X \leftarrow \Pi_{\Gamma_1}\left(U\Sigma V^{\intercal}\right)$ {matrix completion}
8: $\quad$ If $f(X) > Q_t(X, X')$ then $t \leftarrow \eta t$, go to 4.
9: $\quad$ If $|f(X) - Q_t(X, X')| < 0.5/K$ then exit;
10: **end for**

Another example: HMM algorithm

- Let each spoken word to represented by a sequence of speech signals

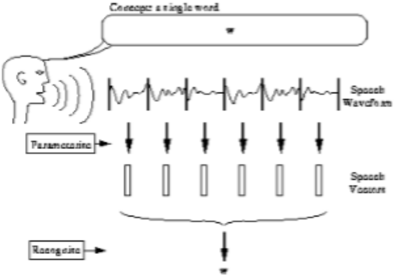
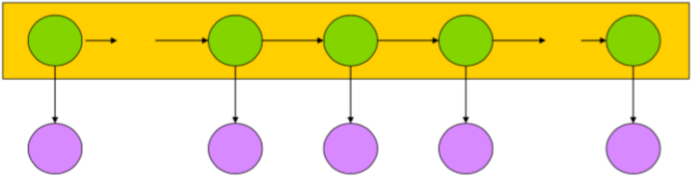


Fig. 1.2 Isolated Word Problem

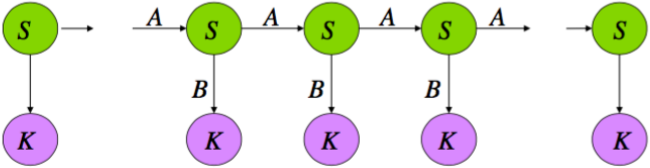
Hidden Markov Model



- Green circles are ***hidden states***
- Dependent only on the previous state
- "The past is independent of the future given the present."

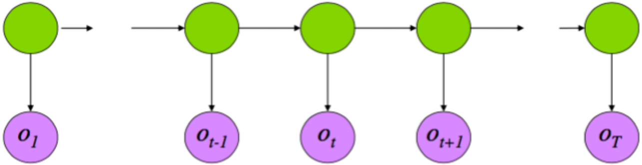
Formalism

HMM Formalism



- $\{S, K, \Pi, A, B\}$
- $\Pi = \{\pi_i\}$ are the initial state probabilities
- $A = \{a_{ij}\}$ are the state transition probabilities
- $B = \{b_{ik}\}$ are the observation state probabilities

Decoding



Given an observation sequence and a model, compute the probability of the observation sequence

$$O = (o_1 \ldots o_T),\ \mu = (A, B, \Pi)$$

Compute $P(O \mid \mu)$

Viterbi algorithm

computing needs statistics

Spectrum: Do we currently have the tools to provide those error bars?

Michael Jordan: We are just getting this engineering science assembled. We have many ideas that come from hundreds of years of statistics and computer science. And we're working on putting them together, making them scalable. A lot of the ideas for controlling what are called familywise errors, where I have many hypotheses and want to know my error rate, have emerged over the last 30 years. But many of them haven't been studied computationally. It's hard mathematics and engineering to work all this out, and it will take time.

It's not a year or two. It will take decades to get right. We are still learning how to do big data well.

The age of big data

Machine-Learning Maestro Michael Jordan on the Delusions of Big Data and Other Huge Engineering Efforts

Big-data boondoggles and brain-inspired chips are just two of the things we're really getting wrong

By Lee Gomes
Posted 20 Oct 2014 | 19:37 GMT

Share | Email | Print

Photo-Illustration: Randi Klett

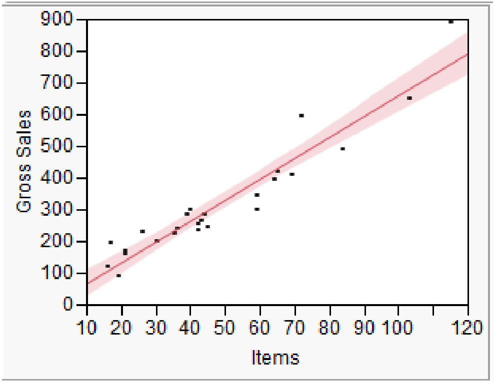
“danger” of big data

Michael Jordan: I like to use the analogy of building bridges. If I have no principles, and I build thousands of bridges without any actual science, lots of them will fall down, and great disasters will occur.

Similarly here, if people use data and inferences they can make with the data without any concern about error bars, about heterogeneity, about noisy data, about the sampling pattern, about all the kinds of things that you have to be serious about if you're an engineer and a statistician—then you will make lots of predictions, and there's a good chance that you will occasionally solve some rea interesting problems. But you will occasionally have some disastrously bad decisions. And you won't know the difference a priori. You will just produce these outputs and hope for the best.

Uncertainty quantification for algorithms

- many machine learning algorithms, little tools for uncertainty quantification ("error bars")
- Many open research problems



Example: bootstrap

- idea: in statistics, we learn about characteristics of the population by taking samples.
- bootstrapping learns about the sample characteristics by taking **resamples** and use the information to infer to the population
- **resample**: we retake samples from the original samples
- calculate the standard error of an estimator, construct confidence intervals, and many other uses