Lectures I & II: The Mathematics of Data

(for folks in partial differential equations, fluid dynamics, scientific computing)

Lek-Heng Lim

University of California, Berkeley

December 21, 2009

Fundamental Problem

Problem

Learn a function

$$f : X \to Y$$

from partial information on f.

Data: Know f on a (very small) subset $\Omega \subseteq X$, i.e. know

$$\{(\omega, f(\omega)) \mid \omega \in \Omega\} \subseteq X \times Y.$$

Model: Know that f belongs to some class of functions $\mathcal{F}(X, Y) \subseteq Y^X$.

Fundamental Objective

Objective

Want graph of f, i.e. want $(x, f(x))$ for all $x \in X$.

Prediction: Given $x \notin \Omega$, want $f(x)$.

Approximation: Y has some measure of nearness, want $\hat{f}$ such that $d(\hat{f}(x), f(x))$ is small.

Classification: Y no intrinsic measure of nearness, want $\hat{f}$ such that $\Pr\{\hat{f}(x) \neq f(x) \mid x \notin \Omega\}$ is small.

Familiar Example: Dirichlet Problem

Problem: Want $f : X \to Y$ where $X \subseteq \mathbb{R}^n$, $Y = \mathbb{R}$.

Data: Know f on ∂X, boundary value/initial value.

Model: f satisfies

$$\Delta f = \varphi$$

for some given φ (say, fluid potential).

Objective: Want f or an approximation $\hat{f}$ on X, i.e. solve PDE analytically or numerically.

Another Example: Spam Filter

Problem: Want $f : X \to Y$ where $X \subseteq \texttt{emails}$, $Y = \{\texttt{spam}, \texttt{ham}\}$.

Data: Know f on $T \subseteq X$, training set, i.e. for $\texttt{email} \in T$, we know whether $f(\texttt{email}) = \texttt{spam}$ or $f(\texttt{email}) = \texttt{ham}$.

Model: What equations do f satisfies? What class of functions should it belong to?

Objective: Want f or an approximation $\hat{f}$ on X, i.e. design a spam filter.

One Major Difference

PDE: We have a physical law of nature describing how f behaves:

$$\Delta f = \varphi.$$

Spam: No law of nature — the 'fundamental laws of emails' too numerous and imprecise to list.

- How to get a reasonable $\mathcal{F}(X, Y)$ for spam filters?
- Use Green functions, just like in PDE (cf. Lecture II).

More Examples

Problems of the latter type increasingly common.

Collaborative filtering: $f : \texttt{movies} \times \texttt{viewers} \to \texttt{ratings}$.

Computer vision: $f : \texttt{handwritten digits} \to \{0, 1, 2, \dots, 9\}$

Machine translation: $f : \texttt{French} \to \texttt{Japanese}$.

Cancer genetics: $f : \texttt{SNPs} \to [0, 1]$; $f =$ likelihood of cancer.

Cancer metabonomics: $f : \texttt{metabolytes} \to \{\texttt{cancer}, \texttt{healthy}\}$.

Modern Massive Data Sets

Characteristics of modern data sets: complex, high-dimensional, massive, nonlinear, non-Gaussian.

- Human-generated data
  - digitization of the entire collections of libraries, medical records of a country;
  - user information collected by data centers of Facebook, Google, Twitter, etc.
- Scientific data
  - genome $\to$ proteome $\to$ transcriptome $\to$ metabolome $\to$ physiome [P. Hunter];
  - sequencing entire ecosystem with high-speed sequencers [C. Venter].
- Plug: `http://mmds.stanford.edu`.

Trouble with Massive Data Sets

- Traditional statistical tools may not work.
- Take example of ranking.
  - Statistics:
    - order statistics,
    - rank statistics,
    - beautiful work of Diaconis with Fourier analysis on $\mathfrak{S}_n$.
  - Problems:
    - combinatorial in nature,
    - $|\mathfrak{S}_n| = n!$,
    - Kemeny optimal is NP-hard.
  - OK if $n = 7$:
    - Number of political parties in Japan.
  - Not OK if $n = 1{,}000{,}000{,}000{,}000$:
    - Unique URLs indexed by Google (July 2008).

Continuum Approximation for Massive Data Sets?

Some examples that we will discuss in these four lectures.

Heat flow: Web search (PageRank).

Green's functions: Spam filtering (Kernel Learning).

Helmholtz decomposition: Product recommendations (HodgeRank).

Elasticity: Cancer metabonomics (Higher-order Tensors).

Web Search

- Suppose you type in a term, say, ‘iPod’ in Google. What happens next?
- Essentially two things:

Retrieval: Find all webpages (inverted index) containing or concerning the term ‘iPod’ and return them.

Ranking: Order the results and present them to you through your browser.

- Second step particularly important.
- Sets modern search engines apart from older ones:
  - Ask.com, Baidu, Bing, Google, Yahoo!
  - Alta Vista, Excite, HotBot, Infoseek, Lycos

Web Search (2009)

Question: How does Google rank its search results nowadays?

Short Answer: No one (not even Google folks) really knows.

Longer Answer: From reliable sources,

- PageRank accounts for about 70% of its ranking methodology.
- Remaining 30% accounted for by about 100 other factors:
  - click-through rate,
  - immediacy,
  - term document analysis,
  - training by human test users,
  - ...
- These factors are used to tweak the PageRank result.
- Seeks to maximize **happiness index**, i.e. the likelihood that what you want is the first result/among the first five results/in the first screen full of results returned.

Web Search (1999)

Question: How did Google rank its search results in 1999?

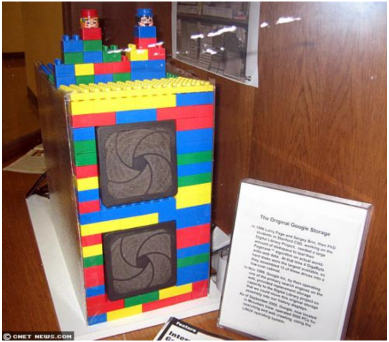

Figure: Original Google site `http://google.stanford.edu`

Answer: PageRank.

The Web as a Directed Graph

- $G_{\texttt{www}} = (V, E)$:
  - nodes $i \in V$ are webpages,
  - directed edges $(i, j) \in E$ are hyperlinks,
  - $n = |V|$.
- Adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$,

$$a_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

- Stochastic adjacency matrix $P = [p_{ij}] \in \mathbb{R}^{n \times n}$,

$$p_{ij} = \begin{cases} 1/\deg(i) & \text{if } (i, j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

PageRank

- Proposed by Larry Page, 1998.
- Used by Google, eigenfactor.org (new ISI impact factor).
- Intuition: a webpage is important if it is pointed to by other important webpages:

$$\left[\alpha P^{\top} + \frac{(1-\alpha)}{n}\mathbf{e}\mathbf{e}^{\top}\right]\mathbf{x} = \mathbf{x}.$$

- **Random surfer model:** $\mathbf{e} = [1, \dots, 1]^{\top}$, $\alpha = 0.85$.
- Matrix is irreducible.
- Perron-Frobenius theorem guarantees existence of $\mathbf{x} > 0$.
- x_i = PageRank of webpage i.

HITS

- Proposed by Jon Kleinberg, 1999.
- Used by Ask.com, Teoma.
- Each webpage i has a **hub score** v_i and an **authority score** u_i.
- Intuition: a good authority is pointed to by may good hubs and a good hub points to many good authorities:

$$u_i' = \sum_{j:(j,i)\in E} v_j, \quad v_i' = \sum_{j:(i,j)\in E} u_j; \quad u_i = u_i'/\|\mathbf{u}'\|, \quad v_i = v_i'/\|\mathbf{y}'\|.$$

- Singular values and singular vectors:

$$\mathbf{u}' = A^\top \mathbf{v}, \quad \mathbf{v}' = A\mathbf{u}; \quad \mathbf{u} = \mathbf{u}'/\|\mathbf{u}'\|, \quad \mathbf{v} = \mathbf{v}'/\|\mathbf{v}'\|.$$

- $u_i =$ authority score of i, $v_i =$ hub score of i.

Diffusion Geometry

- Ronald Coifman's generalization, 2006.
- Graph replaced by data set X. $(X, \mathcal{A}, \mu)$ measure space.
- Kernel $K : X \times X \to \mathbb{R}$ continuous, $K(x, y) = K(y, x)$, and $K(x, y) \geq 0$.
- Degree replaced by volume $d(x) = \int_X K(x, y)\, d\mu(y)$.
- Transition matrix replaced by transition kernel $p(x, y) = K(x, y)/d(x)$. Note that $\int_X p(x, y)\, d\mu(y) = 1$.
- Markov chain replaced by diffusion operator

$$Pf(x) = \int_X p(x, y) f(y) d\mu(y).$$

- Random surfer model becomes random walk on data set X.
- Connections with Fokker-Plank diffusion, Neumann heat kernel.

Mercer Kernels

- Stronger condition on K: for any $n \in \mathbb{N}$, $x_1, \ldots, x_n \in X$, want $[K(x_i, x_j)]_{i,j=1}^n \in \mathbb{R}^{n \times n}$ to be positive definite.
- Canonical example: Gaussian $K(x, y) = \exp(-\|x - y\|^2 / 2\sigma^2)$.
- Integral transform L is compact operator on $L^2(X, \mu)$ (clearly self-adjoint)
$$Lf(x) = \int_X K(x, y) f(y) d\mu(y).$$
- Spectral Theorem: λ_k, φ_k kth eigenvalue/function of L
$$K(x, y) = \sum\nolimits_{k=1}^{\infty} \lambda_k \varphi_k(x) \varphi_k(y),$$
absolutely for any (x, y), uniformly on X (assumed compact).
- What I meant by 'Green functions' earlier (cheated a bit).

Reproducing Kernel Hilbert Space

- Given Mercer kernel K, there is unique Hilbert space $\mathcal{H}_K$ with
  1. $K(x, \cdot) \in \mathcal{H}_K$;
  2. $\operatorname{span}\{K(x, \cdot) \mid x \in X\}$ dense in $\mathcal{H}_K$;
  3. $f(x) = \langle K(x, \cdot), f \rangle_K$ for all $f \in \mathcal{H}_K$.
- Furthermore $\Phi : X \to \ell^2(\mathbb{N})$, $x \mapsto (\sqrt{\lambda_k}\varphi_k(x))_{k \in \mathbb{N}}$ well-defined, continuous, and

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle.$$

- Earlier question revisited. What class of function to use for spam filter? Answer:

$$\mathcal{F}(X, \mathbb{R}) = \mathcal{H}_K$$

for appropriate K.

How to Design a Spam Filter

- $f : X \to Y$ where $X \subseteq \texttt{emails}$, $Y = \{\texttt{spam}, \texttt{ham}\}$.
- Pick kernel K, Galerkin approach:

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x)$$

  where $x_j \in T$, training set.
- Since we know $f(x_j) = y_j$, may solve linear system

$$f(x_j) = \sum_{i=1}^{n} \alpha_i K(x_i, x_j), \qquad j = 1, \dots, n,$$

  for coefficients $\alpha_1, \dots, \alpha_n$.
- Finite element method without PDE!

Classification and Regression

- In practice, need to approximate. E.g. regularized least squares:

$$\min \frac{1}{n} \sum_{j=1}^{n} (f(x_j) - y_j)^2 + \lambda \|f\|_K^2.$$

- Other loss functions possible. E.g. support vector machines use $V(y, f(x)) = (1 - yf(x))_+$ in place of $(f(x) - y)^2$.
- Given $x \notin T$, $f(x) > 0 \Rightarrow x$ is ham, $f(x) < 0 \Rightarrow x$ is spam.
- Applies to other problems as well: collaborative filtering, computer vision, machine translation, cancer genetics.

References I

- R. Coifman, S. Lafon, "Diffusion maps," *Appl. Comput. Harmon. Anal.*, **21** (2006), no. 1, pp. 5–30.
- R. Coifman, I. Kevrekidis, S. Lafon, M. Maggioni, B. Nadler, "Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems," *Multiscale Model. Simul.*, **7** (2008), no. 2, pp. 842–864.
- S.-I. Amari, H. Nagaoka, *Methods of Information Geometry*, AMS, 2000.
- B. Croft, D. Metzler, T. Strohman, *Search Engines: Information retrieval in practice*, Addison-Wesley, 2010.
- C. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval*, Cambridge, 2008.

References II

- T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, 2nd edition, Springer, 2009.
- C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- F. Cucker, D.-X. Zhou, *Learning Theory: An Approximation Theory Viewpoint*, Cambridge, 2007.
- F. Cucker, S. Smale, "On the mathematical foundations of learning," *Bull. Amer. Math. Soc.*, **39** (2002), no. 1, pp. 1–49.
- S. Mallat, *A Wavelet Tour of Signal Processing*, 3rd edition, Academic Press, 2008.
- N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani, *Algorithmic Game Theory*, Cambridge, 2007.