Machine Learning and Language Processing

John Lafferty

LTI & CSD
The coming century is surely the century of data. Our society is investing massively in the collection and processing of data of all kinds, on scales unimaginable until recently.”

—David Donoho (Stanford, August 2000)

- Satellite Images
- DNA sequences
- Internet portals
- Text and hypertext
- Financial transactions
- Astrophysics

...
Managing the Data

- Data is typically very high dimensional and abundant
- Increasing sophistication is available—and required
- Collaboration between computer scientists, statisticians, applications experts, etc. required
Why is Language Data Hard?

- **High dimensions**—for basic problems typically $\gg 10,000$ dimensions

- **Ambiguity**—different words mean different things in different contexts

- **Non-stationarity**—constantly changing source

- **Categorical**—lack of geometric structure limits usefulness of standard techniques; e.g. SVD as an $L^2$ technique
Why is Language Data Interesting?

Same reasons: high dimensions, ambiguity, non-stationarity, and lack of geometry force new statistical methods.

• Terabytes of data are now readily available online

• Not too difficult to get limited labeled data

• Many applications; increasingly important & relevant

• Current technology is useable, though extremely limited; partial solutions can suffice
Basic Language Processing Tasks

- **Prediction/Compression**—Playing the Shannon game; evaluate using “perplexity”
- **Classification**—standard statistical task, though often hierarchical or overlapping labels.
- **Sequence Annotation**—Markov’s original problem; e.g., marking up Web pages
- **Parsing**—annotating sentences with syntactic/semantic structure; viewed as essential for “understanding”
- **Translation**—mapping strings in one language to another; e.g., summarization
The Generation Gap

- Variations on $n$-gram models do fairly well at prediction & compression. Upper bound on entropy: 1.75 bits/char.

- Yet humans can easily detect computer-generated text with very high accuracy ($>99\%$).

- The best statistical estimation procedures have 10% error.

- Statistical modeling for text still has tremendous room for improvement
Basic Statistical Approaches

• *Generative models*—intuitive, flexible, explicit assumptions. The place to start.

• *Discriminative methods*—increasingly sophisticated, distribution free, general theory

• *Hybrid approaches*—conditional likelihood, discriminative training, etc.; compensate for (or avoid) invalid independence assumptions.
Outline

• *Discrete PCA Models* — modeling subtopic structure; alternatives to variational methods

• *Richer Sequence Models* — Conditional random fields and related techniques

• *Kernels* — based on generative models using information geometry

• *Reflections* — are we making progress?
Part I: Discrete PCA Models

- Problem: model subtopic structure of a document
- Application: IR ranking according to aspect coverage
- Standard mixtures models don’t work:

\[ p(d) = \sum_{j} \lambda_j \ p(d \mid \text{topic } j) \]

- Need mixing proportions to vary stochastically
Aspect Models

(Hofmann 1999; Blei, Ng, and Jordan 2002)

Generative: $\lambda \sim \text{Dirichlet}(\alpha)$

**Inference** — Given aspects and document, what is posterior for $\lambda$

**Learning** — Given some documents, what are the (ML) aspects?
Inference

Likelihood has terms

\[ t_w(\lambda)^{n_w} = \left( \sum_a \lambda_a p(w \mid a) \right)^{n_w} \]

Want Dirichlet approximation

\[ \tilde{t}_w(\lambda) \propto \prod_a \lambda_{a}^{\beta_{wa}} \]

Variational Bayes vs. Expectation-Propagation
Two words, two aspects, \( p(w=2 \mid a=1) = 1, \alpha_1 = \alpha_2 = 1 \)
More Results on Synthetic Data
Main Points

• Simple model – difficult inference and learning problems

• Variational methods can lead to biased learning; EP is better on this and other models

• Perplexity is a poor performance measure

• Visualizations and synthetic data are essential tools!

• Future – need more efficient algorithms

http://www.stat.cmu.edu/~minka/papers/aspect.html
Part II: Richer Sequence Models

- **Goal**: mark up sequences with content tags

- **Problem**: overlapping dependencies on context
  - long-distance dependencies
  - multiple levels of granularity (e.g., words & characters)
  - aggregate properties (e.g., layout, html)
  - past and future observations

- *Generative* models that can represent such dependencies quickly become computationally intractable

- Problems arise in many domains; e.g., biological sequence analysis
Motivating Problem:
Segment & Annotate Data with Content Tags
Conditional Models

- Model $p(\text{label sequence } y \mid \text{observation sequence } x)$ rather than joint probability $p(y, x)$

- Allow arbitrary dependencies on the observation sequence

- Still efficient (Viterbi, forward-backward) if dependencies within the state sequence $y$ are local

- Do not need to use states to model dependency on past and future observations $\Rightarrow$ smaller state space, easier to design
Conditional Random Fields

\[
p(y | x) = \frac{1}{Z(x)} \exp \left( \sum_s \lambda_s \Phi_s(x, y) \right)
\]

- **Global** normalization. Undirected graphical models; more general and expressive modeling technique
- Easy to incorporate domain knowledge without increasing state space or designing special-purpose transition structure
- Parameters need not fully specify states or observations; require less training data
Main Points

• Conditional models, global normalization

• Rich and “expressive” family of models

• “It’s the features stupid!”

• Biases and misuse of models can be subtle

• Perceptron-esque, stochastic gradient ascent algorithms promising
Part III: Kernels

(Aizerman, Braverman & Rozonoér, 1964) (Boser, Guyon & Vapnik, 1992)

• A powerful & elegant tool—dramatic improvements in accuracy (and sometimes computational efficiency)

• Popular indoor sport—“Kernel $X$”, where $X$ is PCA, ICA, Clustering, . . . .

• Turns linear classifiers into non-linear classifiers

• Implicit representation of high dimensional feature spaces

• *Primarily a tool for Euclidean space*
Linear Classifiers

\[
\hat{f}(x) = \sum_{i=1}^{N} \alpha_i y_i \langle x, x_i \rangle + \hat{\beta}_0
\]
Using a Kernel

\[ \hat{f}(x) = \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + \hat{\beta}_0 \]
Popular Kernels

Gaussian: \[ K_\sigma(x, x') = \exp \left( -\frac{\|x - x'\|^2}{2\sigma^2} \right) \]

Polynomial: \[ K_d(x, x') = (1 + \langle x, x' \rangle)^d \]

Sigmoid: \[ K_\alpha(x, x') = \tanh (\langle x, x' \rangle + \alpha) \]

However, these only apply in Euclidean spaces. Usually means data has already been “compromised”
Structured Data

What if data lies on a graph or other data structure?
In analogy with the continuous setting, we call

$$\frac{d}{d\beta} K_\beta = -\Delta K_\beta$$

the *heat equation* on a graph. Solution

$$K_\beta = e^{-\beta \Delta}$$

is called the *heat kernel* or *diffusion kernel*.
Combining with Generative Models

Statistical family \( \{ p(x \mid \theta) \}_{\theta \in \Theta} \)

Fisher information

\[
g_{ij}(\theta) = \int_{\mathcal{X}} \frac{\partial}{\partial \theta_i} \log p(x \mid \theta) \frac{\partial}{\partial \theta_j} \log p(x \mid \theta) p(x \mid \theta) \, dx
\]

Gives \( \Theta \) the structure of a Riemannian manifold
Idea: Diffusion on the Information Manifold

• Generative models are fairly easy to derive

• “Similarity” between \( x \) and \( x' \) may be difficult to quantify—
even for domain experts

• A good generative model will use domain knowledge; e.g.,
state structure of an HMM

• Fisher information on the family “codifies” this into a
distance

• Associate each data point with a model: \( x \mapsto \theta(x) \)
Idea: The Heat Diffusion Kernel

- \( (\Delta - \frac{\partial}{\partial t}) K = 0 \), initial condition \( \delta_x(y) \):
- \( K_t(x, y) = \int_M K_{t-s}(x, z) K_s(z, y) \, dz \)
- \( e^{t\Delta} f(x) = \int_M K_t(x, y) f(y) \, dy \)

For a Kernel-based classifier

\[
\hat{y}(x) = \sum_i \alpha_i y_i K_t(x_i, x)
\]

decision function is heat flow with initial condition

\[
f(x) = \begin{cases} \alpha_i y_i & x \in \text{data} \\ 0 & \text{otherwise} \end{cases}
\]
Special Case: Multinomial

\[ \theta \in d\text{-}simplex \Rightarrow \left( \sqrt{\theta_1}, \ldots, \sqrt{\theta_{d+1}} \right) \in d\text{-}sphere \]

Multinomial geometry is geometry of the sphere

\[ d(\theta, \theta') = 2 \arccos \left( \sum_{i=1}^{d+1} \sqrt{\theta_i \theta'_i} \right) \]

\[ K_t(\theta, \theta') \approx \frac{1}{(4\pi t)^{d/2}} \exp \left( -\frac{1}{t} \arccos^2 \left( \sum_{i=1}^{d+1} \sqrt{\theta_i \theta'_i} \right) \right) \]
SVM Decision Boundaries: Trinomial

Gaussian kernel

Information diffusion kernel
WebKB Text Classification Experiments

![Graph showing test set error rate vs. number of training examples for different kernels.](image-url)

- Linear
- RBF
- IDK

Number of training examples: 50, 100, 150, 200, 250
Test set error rate: 0, 0.02, 0.04, 0.06, 0.08, 0.1, 0.12, 0.14, 0.16
Summary

• Diffusion kernels are broad generalization of standard Gaussian kernels

• Applying to Fisher information combines generative models and discriminative learning methods.

• “Physical interpretation” of learning algorithm

• Many interesting directions to explore. Example: HMMs and other sequence models
Part IV: Are We Making Progress?

- Yes!

- More data hardly solves problems—only makes them more interesting and important

- A lot to learn even about simple models, algorithms, kernels

- Confluence of language learning with general machine learning is welcome, fruitful.
“In the past fifteen years, the growth in algorithmic modeling applications and methodology has been rapid. It has occurred largely outside statistics in a new community—often called machine learning—that is mostly young computer scientists. The advances, particularly over the last five years, have been startling.”

—Leo Breiman (Berkeley Statistics)

Statistical Science, August 2001
Going to Where the Wild Data Are