# W4240/W6240

# Data Mining/ Statistical Machine Learning

Frank Wood

January 18, 2011

# Introduction

- Data mining is the search for patterns in large collections of data
  - Designing models
  - Fitting models to data
  - Using models to perform inference/prediction
- Pattern recognition is concerned with *automatically* finding patterns in data / learning models
- Machine learning is pattern recognition with concern for computational tractability and full automation
- Data mining = Machine Learning = Applied Statistics
  - Scale
  - *Computation*

# High Level Course Goals

- Problem formulation
  - Starting from data and/or a question, you will learn how to create and design a model that will answer the question(s) of interest.
  - You will learn how to formally, mathematically codify a model.
  - You will learn how to fit models.
  - You will learn to think about computational/inferential trade-offs.
- Tool *Creation*
  - You will learn how to design and implement general purpose learning algorithms
  - You will implement inference algorithms for several models such as latent Dirichlet allocation, Bayesian logistic regression, Gaussian mixture models and more.
- Practice
  - Through your homework and final project you will be evaluated on how well you can put into practice the theory that will be taught in class.

# Style of Instruction

- Great textbook (Bishop, Pattern Recognition and Machine Learning, *required!*)!
  - Will follow second half of text closely.
  - Class time will be spent on *theory* and "*proof*," explaining the tricky parts of the text by doing math on the board.
  - Homework (extensive and hard) will be spent on practice.
  - Team-based final project
  - You will learn to think about computational/inferential trade-offs.
- Tool *Creation*
  - You will learn how to design and implement general purpose learning algorithms
  - You will implement inference algorithms for several models such as latent Dirichlet allocation, Bayesian logistic regression, Gaussian mixture models and more.
- This is *not* a *"What tool do we use and how do we use it?"* course!

# Grading - What you should expect.

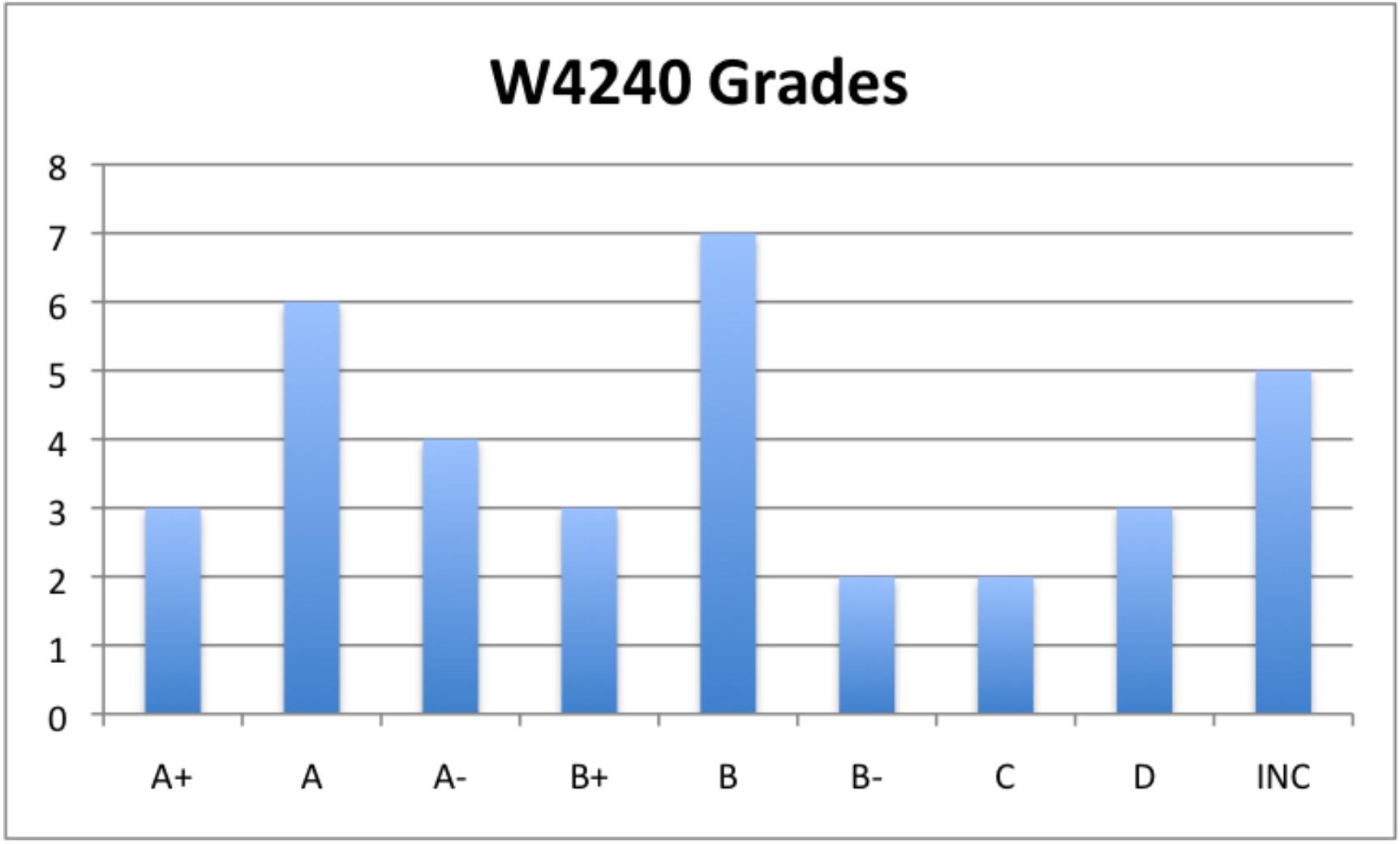


Figure: Fall 2010 Grade Distribution

# Links and Syllabus

- Course home page :
  http://www.stat.columbia.edu/∼fwood/w4240/
- *Bookmark* this page (not courseworks)
- Guest lectures may be sprinkled throughout the course.

# Prerequisites

- Linear algebra
- Multivariate calculus (matrix and vector calculus)
- Probability and statistics at a masters level
- Programming experience in some language like pascal, matlab, c++, java, c, fortran, scheme, etc.
- Some algorithmic complexity theory (basics) useful
- Information theory (entropy, KL-divergence)

# Review

Good idea to familiarize yourself with PRML [3] Chapter 1 and 2 and Appendices B,C,D, and E. In particular:

- Information theory
- Multivariate Gaussian distribution
- Discrete, multinomial, and Dirichlet distributions
- Lagrange multipliers
- Matlab

We will offer extra Matlab programming sections, a review of both the multivariate Gaussian and information theory.

# Homework (from anonymous student feedback)

> *The assignments were challenging and probably the best part of the course. It was with bated breath, and nervous anticipation - almost like a blind date with someone you knew would be pretty - that we waited for the homework ok, maybe that was 10% exaggerated. ....*
>
> *The assignments in this course were more challenging than any other course I have taken in the past.*
>
> *Really enjoyed the assignments in this class. Excellent job putting them together.*
>
> *Very difficult programming assignments, and I learned a lot doing them,*

The homework in this course is *programming*. You will *implement* various data mining / machine learning algorithms. If you have not programmed before the course is doable but difficult.

# Project

The final project is a significant piece of team-based work that demonstrates your data mining / statistical machine learning knowledge on a problem domain of interest to you (and hopefully also of interest to a larger academic, governmental, or industry community). The deliverables include a 2 page proposal; a short, publication-quality paper; and a 10-20 minute presentation. You will fail to complete a satisfactory final project if you wait until the last minute to begin.

# Past Example Projects

- Government Bias in sub-Saharan African Press
  - Bayesian logistic regression to predict "slant" of press articles from extracted text features
- A Pattern Recognition System for American Sign Language
  - Component analysis approach to feature extraction and classification of ASL from *videos* of expert signers
- Supervised Topic Modeling in Clinical Text
  - Automatically assign ICD-9-CM codes to patient discharge records.
- FriendFinder: Predicting missing links in a Gargantuan social network
- Ordering Shakespeares Plays using a Sequential Generative Model

# Syllabus

- Review
- Graphical models
  - Belief propagation
- Expectation Maximization
- Variational Inference
- Sampling
- Misc.

Along the way you will implement estimation and inference procedures for the following models (minimally)

- Gaussian mixture model
- Bayesian linear regression
- Bayesian logistic regression
- Latent Dirichlet allocation
- ...

# Today: Really Big Picture

- The glue that binds the course together : graphical models.
- A guiding philosophy : Bayesian inference.

# The Glue: Graphical Models

- Many probabilistic models can be expressed in the "language" of graphical models.

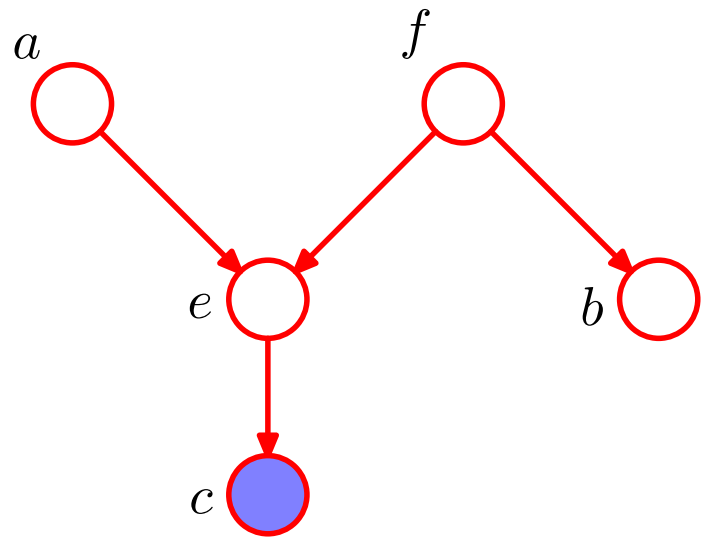


Figure: Directed Graphical Model : Chapter 8, Figure 22a, PRML [3]

$$P(a, b, c, e, f) = P(a)P(f)P(e|a, f)P(b|f)P(c|e)$$

# Graphical Models Cont.

- Correspond (sometimes) to a "plausible" generative mechanism.
- Reveal latent variable choices and help clarify what inferences can be performed
- Provide datastructure on which inference and estimation algorithms can run
- Specify conditional indendencies and highlight computational savings

## Course Goal

Learn how to "think" in terms of graphical models. Be able to write down a generative graphical model for data of interest to you and to know how to do inference in generic graphical models.

*Give ordering Shakespeare example*

# Example directed graphical model / Bayes net : ALARM, expert diagnostic system

Goal: Inference in given/known/hand-specified Bayesian network

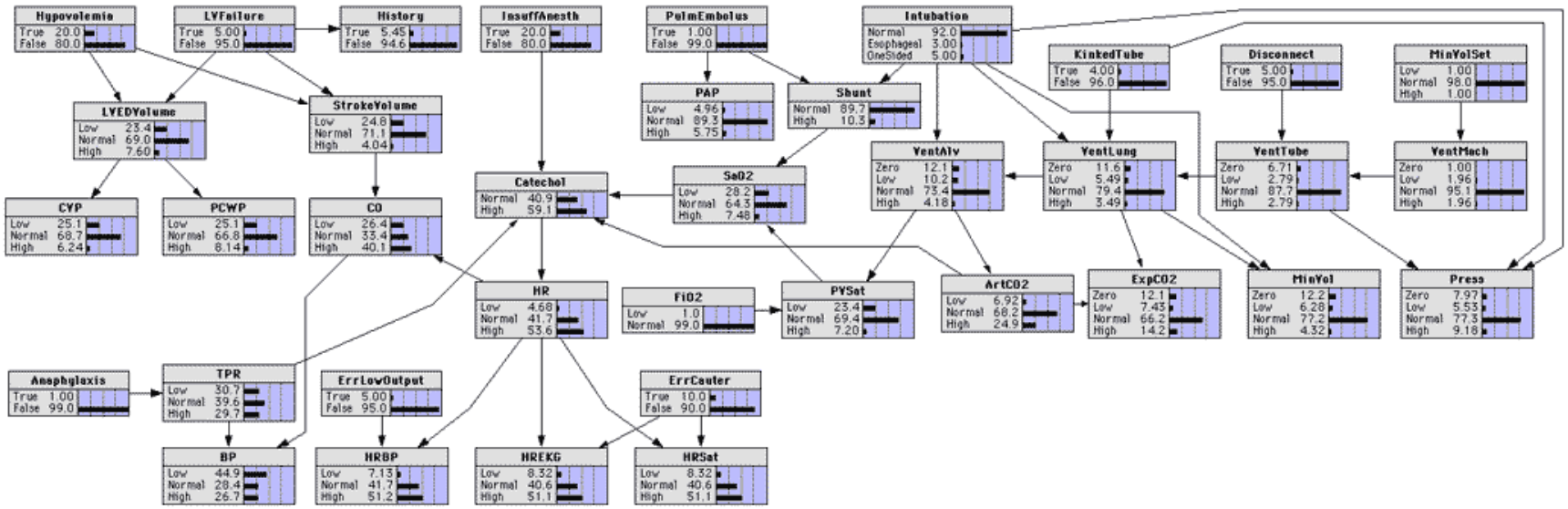


Figure: ALARM stands for 'A Logical Alarm Reduction Mechanism'. This is a medical diagnostic system for patient monitoring. It is a nontrivial belief network with 8 diagnoses, 16 findings and 13 intermediate variables. Described in [2]

# Inference in discrete directed acyclic graphical models

Inference procedures known as the sum-product algorithm and belief propagation are general inference techniques that can easily be adapted to discrete and linear-Gaussian graphical models.

Belief propagation

- Computes marginal distributions of any subset of variables in the graphical model conditioned on any other subset of variables (values observed / fixed)
- Generalizes many, many inference procedures such as Kalman filter, forward-backward, etc.
- Can be used for parameter estimation in the case where all latent, unknown variables are "parameters" and all observations are fixed, known variables.

# Bayesian Analysis Recipe

Bayesian data analysis can be described as a three step process

1. Set up a full (generative) probability model
2. Condition on the observed data to produce a posterior distribution, the conditional distribution of the unobserved quantities of interest (parameters or functions of the parameters, etc.)
3. Evaluate the goodness of the model
4. Perform inference taking into account the uncertainty about the model parameters encoded in the posterior distribution

# Philosophy

Gelman, "Bayesian Data Analysis"

> *A primary motivation for believing Bayesian thinking important is that it facilitates a common-sense interpretation of statistical conclusions. For instance, a Bayesian (probability) interval for an unknown quantity of interest can be directly regarded as having a high probability of containing the unknown quantity, in contrast to a frequentist (confidence) interval, which may strictly be interpreted only in relation to a sequence of similar inferences that might be made in repeated practice.*

# Theoretical Setup

Consider a model with parameters $\Theta$ and observations that are independently and identically distributed from some distribution $X_i \sim F(\cdot, \Theta)$ parameterized by $\Theta$.
Consider a prior distribution on the model parameters $P(\Theta; \Psi)$

- What does

$$P(\Theta|X_1, \ldots, X_N; \Psi) \propto P(X_1, \ldots, X_N|\Theta; \Psi)P(\Theta; \Psi)$$

  mean?
- What does $P(\Theta; \Psi)$ mean? What does it represent?

In this course we will consider complicated likelihoods and priors (many parameters, often related in non-trivial ways) and the algorithms required to perform inference in such models.

# Very Simple Example

Consider the following example: suppose that you are thinking about purchasing a factory that makes pencils. Your accountants have determined that you can make a profit (i.e. you should transact the purchase) if the percentage of defective pencils manufactured by the factory is less than 30%.

In your prior experience, you learned that, on average, pencil factories produce defective pencils at a rate of 50%.

To make your judgement about the efficiency of this factory you test pencils one at a time in sequence as they emerge from the factory to see if they are defective.

# Notation

Let $X_1, \ldots, X_N, X_i \in \{0, 1\}$ be a set of defective/not defective observations.

Let $\Theta$ be the probability of pencil defect.

Let $P(X_i|\Theta) = \Theta^{X_i}(1 - \Theta)^{1-X_i}$ (a Bernoulli random variable)

# Typical elements of Bayesian inference

Two typical Bayesian inference objectives are

1. The *posterior distribution* of the model parameters

$$P(\Theta|X_1, \ldots, X_n) \propto P(X_1, \ldots, X_n|\Theta)P(\Theta)$$

   This distribution is used to make statements about the distribution of the unknown or latent quantities in the model.

2. The *posterior predictive distribution*

$$P(X_n|X_1, \ldots, X_{n-1}) = \int P(X_n|\Theta)P(\Theta|X_1, \ldots, X_{n-1})d\Theta$$

   This distribution is used to make predictions about the population given the model and a set of observations.

# The Prior

Both the posterior and the posterior predictive distributions require the choice of a prior over model parameters $P(\Theta)$ which itself will usually have some parameters. If we call those parameters $\Psi$ then you might see the prior written as $P(\Theta; \Psi)$.

The prior encodes your prior belief about the values of the parameters in your model. The prior has several interpretations and many modeling uses

- Encoding previously observed, related observations (pseudocounts)
- Biasing the estimate of model parameters towards more realistic or probable values
- Regularizing or contributing towards the numerical stability of an estimator
- Imposing constraints on the values a parameter can take

# Choice of Prior - Continuing the Example

In our example the model parameter $\Theta$ can take a value in $\Theta \in [0, 1]$. Therefore the prior distribution's support should be $[0, 1]$

One possibility is $P(\Theta) = 1$. This means that we have no prior information about the value $\Theta$ takes in the real world. Our prior belief is uniform over all possible values.

Given our assumptions (that 50% of manufactured pencils are defective in a typical factory) this seems like a poor choice.

A better choice might be a non-uniform parameterization of the Beta distribution.

# Beta Distribution

The Beta distribution $\Theta \sim \text{Beta}(\alpha, \beta)$ ($\alpha > 0, \beta > 0, \Theta \in [0, 1]$) is a distribution over a single number between 0 and 1. This number can be interpreted as a probability. In this case, one can think of $\alpha$ as a pseudo-count related to the number of successes (here a success will be the failure of a pencil) and $\beta$ as a pseudo-count related to the number of failures in a population. In that sense, the distribution of $\Theta$ encoded by the Beta distribution can produce many different biases.

The formula for the Beta distribution is

$$P(\Theta|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \Theta^{\alpha-1}(1 - \Theta)^{\beta-1}$$

Run introduction_to_bayes/main.m

# $\Gamma$ function

In the formula for the Beta distribution

$$P(\Theta|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\Theta^{\alpha-1}(1-\Theta)^{\beta-1}$$

The gamma function (written $\Gamma(x)$) appears.

It can be defined recursively as $\Gamma(x) = (x-1)\Gamma(x-1) = (x-1)!$ with $\Gamma(1) = 1$.

This is just a generalized factorial (to real and complex numbers in addition to integers). It's value can be computed. It's derivative can be taken, etc.

Note that, by inspection (and definition of distribution)

$$\int \Theta^{\alpha-1}(1-\Theta)^{\beta-1}d\Theta = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

# Beta Distribution

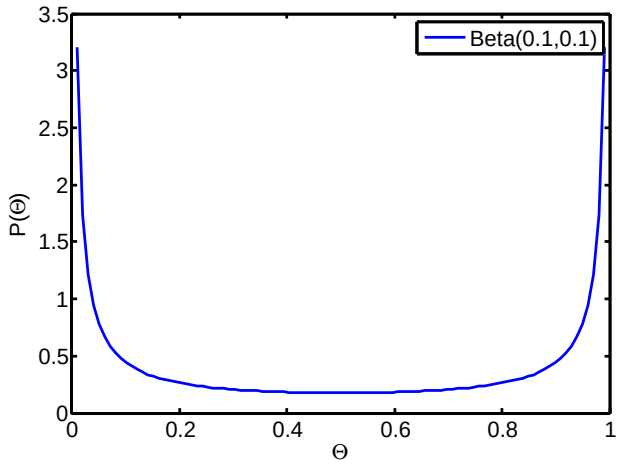


Figure: Beta(.1,.1)

# Beta Distribution

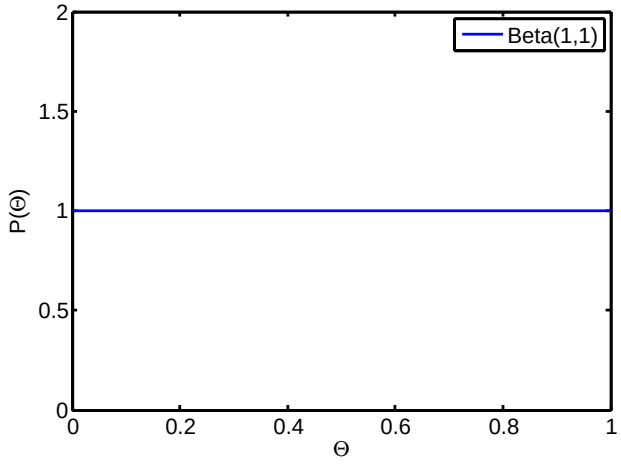


Figure: Beta(1,1)

# Beta Distribution

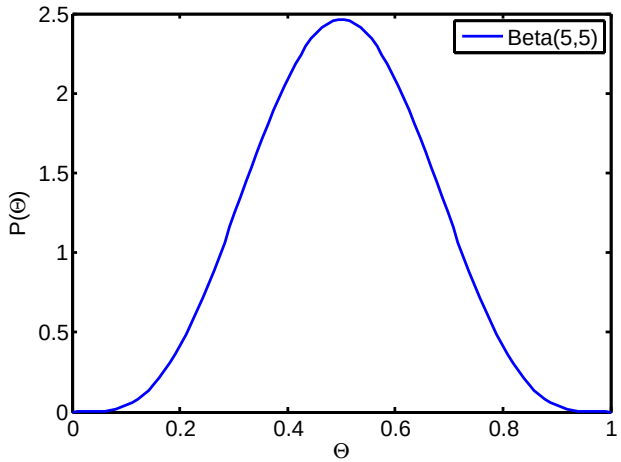


Figure: Beta(5,5)

# Beta Distribution

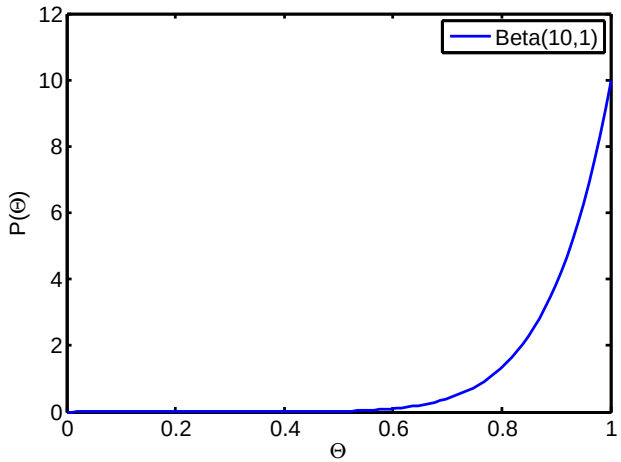


Figure: Beta(10,1)

# Generative Model

With the introduction of this prior we now have a full generative model of our data (given $\alpha$ and $\beta$, the model's hyperparameters). Consider the following procedure for generating pencil failure data:

- Sample a failure rate parameter $\Theta$ for the "factory" from a Beta$(\alpha, \beta)$ distribution. This yields the failure rate for the factory.
- Given the failure rate $\Theta$, sample $N$ defect/no-defect observations from a Bernoulli distribution with parameter $\Theta$.

Bayesian inference involves "turning around" this generative model, i.e. uncovering a distribution over the parameter $\Theta$ given both the observations and the prior.

This class will be about the general purpose computations necessary to do this.

# Inferring the Posterior Distribution

Remember that the *posterior distribution* of the model parameters is given by Bayes rule, here

$$P(\Theta|X_1, \ldots, X_n) \propto P(X_1, \ldots, X_n|\Theta)P(\Theta)$$

Let's consider what the posterior looks like after observing a single observation (in our example).

Our likelihood is given by

$$P(X_1|\Theta) = \Theta^{X_1}(1-\Theta)^{1-X_1}$$

Our prior, the Beta distribution, is given by

$$P(\Theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\Theta^{\alpha-1}(1-\Theta)^{\beta-1}$$

# Analytic Posterior Update - No Computation

Since we know that

$$P(\Theta|X_1) \propto P(X_1|\Theta)P(\Theta)$$

we can write

$$P(\Theta|X_1) \propto \Theta^{X_1}(1-\Theta)^{1-X_1}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\Theta^{\alpha-1}(1-\Theta)^{\beta-1}$$

but since we are interested in a function (distribution) of $\Theta$ and we are working with a proportionality, we can throw away terms that do not involve $\Theta$ yielding

$$P(\Theta|X_1) \propto \Theta^{\alpha+X_1-1}(1-\Theta)^{1-X_1+\beta-1}$$

Because of *conjugacy* we have an analytic form for the posterior distribution of the model parameters of the data. The class is about similar computation in more difficult models.
Note that this is an incremental procedure.

# Bayesian Computation, Implicit Integration

From the previous slide we have

$$P(\Theta|X_1) \propto \Theta^{\alpha+X_1-1}(1-\Theta)^{1-X_1+\beta-1}$$

To make this proportionality an equality (i.e. to construct a properly normalized distribution) we have to integrate this expression w.r.t. $\Theta$, i.e.

$$P(\Theta|X_1) = \frac{\Theta^{\alpha+X_1-1}(1-\Theta)^{1-X_1+\beta-1}}{\int \Theta^{\alpha+X_1-1}(1-\Theta)^{1-X_1+\beta-1} d\Theta}$$

But in this and other special cases like it (when the likelihood and the prior form a conjugate pair) this integral can be solved by recognizing the form of the distribution, i.e. note that this expression looks exactly like a Beta distribution but with updated parameters, $\alpha_1 = \alpha + X_1, \beta_1 = \beta + 1 - X_1$

# Posterior and Repeated Observations

This yields the following pleasant result

$$\Theta|X_1, \alpha, \beta \sim \text{Beta}(\alpha + X_1, \beta + 1 - X_1)$$

This means that the posterior distribution of $\Theta$ given an observation is in the same parametric family as the prior. This is characteristic of conjugate likelihood/prior pairs.

Note the following decomposition

$$P(\Theta|X_1, X_2, \alpha, \beta) \propto P(X_2|\Theta, X_1)P(\Theta|X_1, \alpha, \beta)$$

This means that the preceding posterior update procedure can be repeated. This is because $P(\Theta|X_1, \alpha, \beta)$ is in the same family (Beta) as the original prior. The posterior distribution of $\Theta$ given two observations will still be Beta distributed, now just with further updated parameters.

# Incremental Posterior Inference

Starting with

$$\Theta|X_1, \alpha, \beta \sim \text{Beta}(\alpha + X_1, \beta + 1 - X_1)$$

and adding $X_2$ we can almost immediately identify

$$\Theta|X_1, X_2, \alpha, \beta \sim \text{Beta}(\alpha + X_1 + X_2, \beta + 1 - X_1 + 1 - X_2)$$

which simplifies to

$$\Theta|X_1, X_2, \alpha, \beta \sim \text{Beta}(\alpha + X_1 + X_2, \beta + 2 - X_1 - X_2)$$

and generalizes to

$$\Theta|X_1, \ldots, X_N, \alpha, \beta \sim \text{Beta}(\alpha + \sum X_i, \beta + N - \sum X_i)$$

# Interpretation, Notes, and Caveats

- The posterior update computation performed here is unusually simple in that it is analytically tractable. The integration necessary to normalize the posterior distribution is more often not analytically tractable than it is analytically tractable. When it is not analytically tractable other methods must be utilized to get an estimate of the posterior distribution – numerical integration and Markov chain Monte Carlo (MCMC) amongst them.
- The posterior distribution can be interpreted as the distribution of the model parameters given both the structural assumptions made in the model selection step and the selected prior parameterization. Asking questions like, "What is the probability that the factory has a defect rate of less than 10%?" can be answered through operations on the posterior distribution.

# More Interpretation, Notes, and Caveats

The posterior can be seen in multiple ways

$$\begin{aligned} P(\Theta|X_{1:N}) &\propto P(X_1,\ldots,X_N|\Theta)P(\Theta) \\ &\propto P(X_N|X_{1:N-1},\Theta)P(X_{N-1}|X_{1:N-2},\Theta)\cdots P(X_1|\Theta)P(\Theta) \\ &\propto P(X_N|\Theta)P(X_{N-1}|\Theta)\cdots P(X_1|\Theta)P(\Theta) \end{aligned}$$

(when $X$'s are iid given $\Theta$ or exchangeable) and

$$\begin{aligned} P(\Theta|X_1,\ldots,X_N) &\propto P(X_N,\Theta|X_1\ldots,X_{N-1}) \\ &\propto P(X_N|\Theta)P(\Theta|X_1\ldots,X_{N-1}) \end{aligned}$$

The first decomposition highlights the fact that the posterior distribution is influenced by each observation.

The second recursive decomposition highlights the fact that the posterior distribution can be interpreted as the full characterization of the uncertainty about the hidden parameters after having accounted for all observations to some point.

# Posterior Predictive Inference

Now that we know how to update our prior beliefs about the state of latent variables in our model we can consider posterior predictive inference.

Posterior predictive inference performs a weighted average prediction of future values over all possible settings of the model parameters. The prediction is weighted by the posterior probability of the model parameter setting, i.e.

$$P(X_{N+1}|X_{1:N}) = \int P(X_{N+1}|\Theta)P(\Theta|X_{1:N})d\Theta$$

Note that this is just the likelihood convolved against the posterior distribution having accounted for $N$ observations.

# More Implicit Integration

If we return to our example we have the updated posterior distribution

$$\Theta|X_1,\ldots,X_N,\alpha,\beta \sim \text{Beta}(\alpha+\sum_{i=1}^{N} X_i, \beta + N - \sum_{i=1}^{N} X_i)$$

and the likelihood of the $(N+1)^{th}$ observation

$$P(X_{N+1}|\Theta) = \Theta^{X_{N+1}}(1-\Theta)^{1-X_{N+1}}$$

Note that the following integral is similar in many ways to the posterior update

$$P(X_{N+1}|X_{1:N}) = \int P(X_{N+1}|\Theta)P(\Theta|X_{1:N})d\Theta$$

which means that in this case (and in all conjugate pairs) this is easy to do.

# More Implicit Integration

$$
\begin{aligned}
P(X_{N+1}|X_{1:N}) &= \int \Theta^{X_{N+1}}(1-\Theta)^{1-X_{N+1}} \\
&\times \frac{\Gamma(\alpha+\beta+N)}{\Gamma(\alpha+\sum_{i=1}^{N} X_i)\Gamma(\beta+N-\sum_{i=1}^{N} X_i)} \\
&\times \Theta^{\alpha+\sum_{i=1}^{N} X_i-1}(1-\Theta)^{\beta+N-\sum_{i=1}^{N} X_i)-1} d\Theta
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{\Gamma(\alpha+\beta+N)}{\Gamma(\alpha+\sum_{i=1}^{N} X_i)\Gamma(\beta+N-\sum_{i=1}^{N} X_i)} \\
&\times \frac{\Gamma(\alpha+\sum_{i=1}^{N} X_i+X_{N+1})\Gamma(\beta+N+1-\sum_{i=1}^{N} X_i-X_{N+1})}{\Gamma(\alpha+\beta+N+1)}
\end{aligned}
$$

# Interpretation

$$
\begin{aligned}
& P(X_{N+1}|X_{1:N}) \\
&= \frac{\Gamma(\alpha+\beta+N)}{\Gamma(\alpha+\sum_{i=1}^{N} X_i)\Gamma(\beta+N-\sum_{i=1}^{N} X_i)} \\
&\times \frac{\Gamma(\alpha+\sum_{i=1}^{N} X_i + X_{N+1})\Gamma(\beta+N+1-\sum_{i=1}^{N} X_i - X_{N+1})}{\Gamma(\alpha+\beta+N+1)}
\end{aligned}
$$

Is a ratio of Beta normalizing constants.

This a distribution over $[0, 1]$ which averages over all possible models in the family under consideration (again, weighted by their posterior probability).

# Caveats again

In posterior predictive inference many of the same caveats apply.

- Inference can be computationally demanding if conjugacy isn't exploited.
- Inference results are only as good as the model and the chosen prior.

But Bayesian inference has some pretty big advantages

- Assumptions are explicit and easy to characterize.
- It is easy to plug and play Bayesian models.

# Beta Distribution

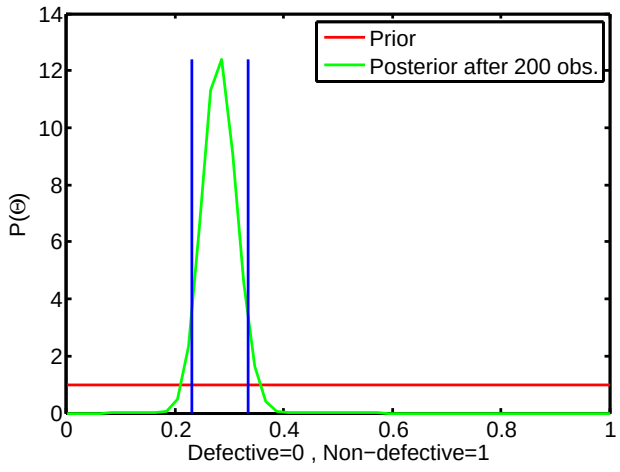


Figure: Posterior after 1000 observations.

# Beta Distribution

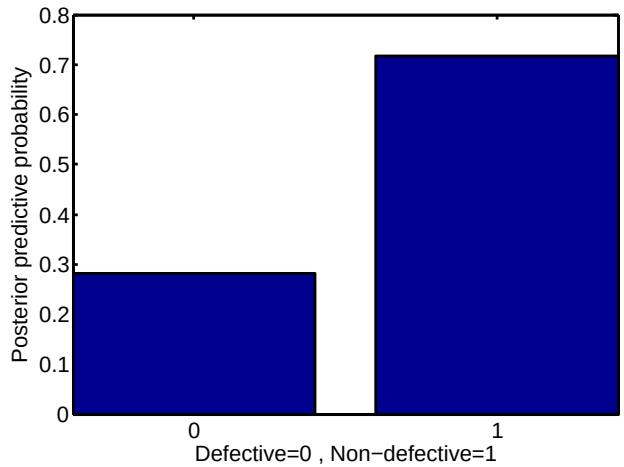


Figure: Posterior predictive after 1000 observations.

# More complicated models

It is possible to extend this generative framework to model more complicated phenomena. The basics stay the same, computation just gets harder.
Canonical examples include

- Classification of handwritten digits
- Trajectory inference
- Clustering

# Digit classification cast as probabilistic modeling challenge

Goal

- Build a machine that can identify handwritten digits automatically

Approaches

- Hand craft a set of rules that separate each digit from the next
- Set of rules invariably grows large and unwieldy and requires many "exceptions"
- "Learn" a set of models for each digit automatically from labeled training data, i.e. *mine* a large collection of handwritten digits and produce a model of each
- Use model to do classification

Formalism

- Each digit is 28x28 pixel image
- Vectorized into a 784 entry vector $\mathbf{x}$

# Handwritten Digit Recognition Training Data

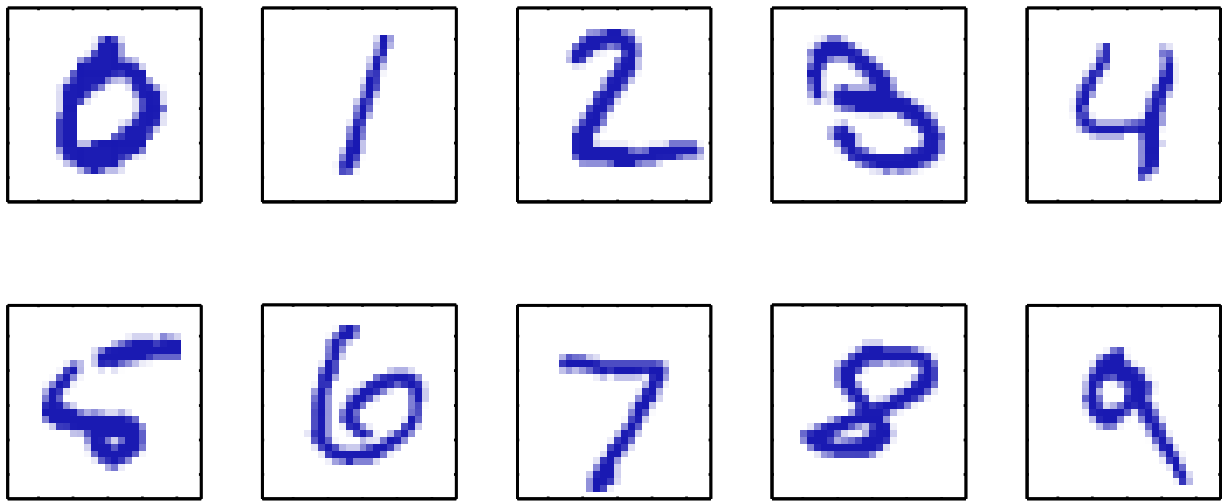

Figure: Hand written digits from the USPS

# Machine learning approach to digit recognition

Recipe

- Obtain a of $N$ digits $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ called the *training set*.
- Label (by hand) the training set to produce a label or "target" $\mathbf{t}$ for each digit image $\mathbf{x}$
- Learn a function $\mathbf{y}(\mathbf{x})$ which takes an image $\mathbf{x}$ as input and returns an output in the same "format" as the target vector.

Terminology

- The process of determining the precise shape of the function $\mathbf{y}$ is known as the "training" or "learning" phase.
- After training, the model (function $\mathbf{y}$) can be used to figure out what digit unseen images might be of. The set comprised of such data is called the "test set"

# Tools for the handwriting recognition job

Supervised Regression/Classification Models

- Logistic regression
- Neural networks
- Support vector machines
- Naive Bayes classifiers

Unsupervised Clustering

- Gaussian mixture model

Model Parameter Estimation

- Maximum likelihood / Expectation Maximization
- Variational inference
- Sampling
- Sequential Monte Carlo
  - ... for all, batch or online

# Example Application: Trajectory Inference From Noisy Data

Goal

- Build a machine that can uncover and compute the true trajectory of an indirectly and noisily observed moving target

Approaches

- Hand craft a set of rules that govern the possible movements of said target
- Set of rules invariably grows large and unwieldy and requires many "exceptions"
- "Learn" a model of the kind of movements such a target can make and perform inference in said model

Formalism

- Example observed trajectories $\{\mathbf{x}_n\}_{n=1}^N$
- Unobserved latent trajectories $\{\mathbf{z}_n\}_{n=1}^N$

# Latent trajectory Inference

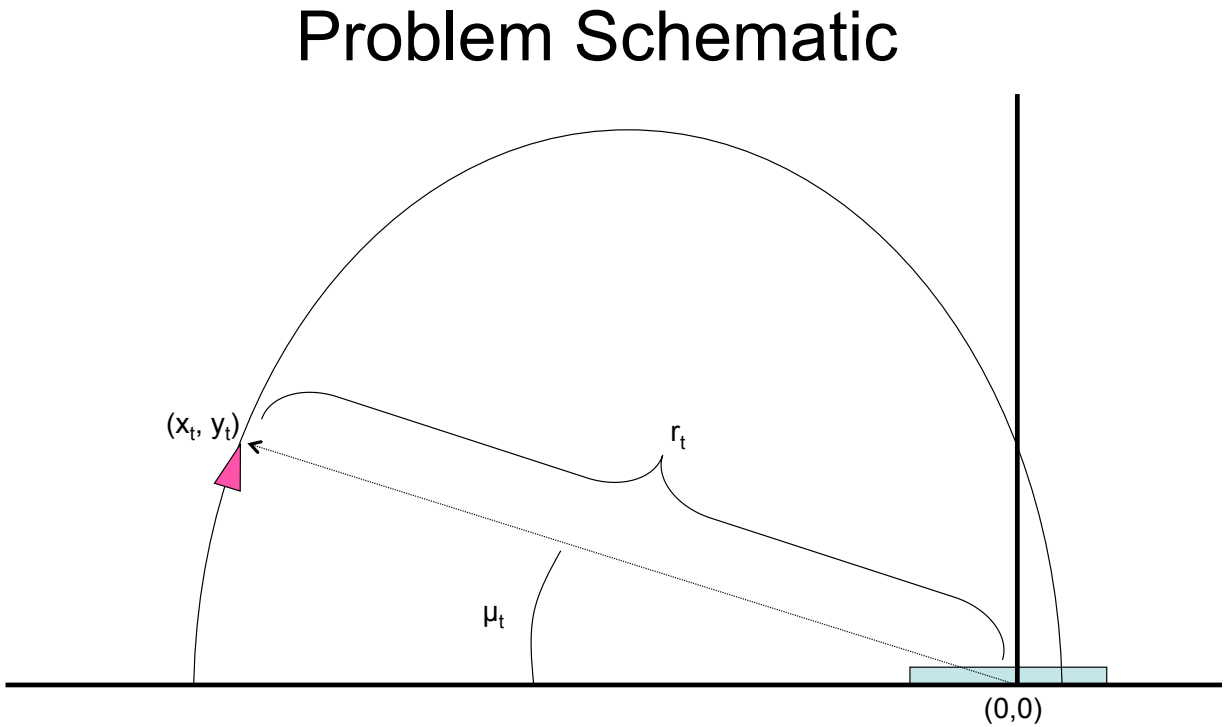


Figure: Schematic of trajectory inference problem

# Tools for Latent Trajectory Inference

Known/hand-crafted model, inference only

- Belief propagation
- Kalman filter
- Particle filter
- Switching variants thereof
- Hidden Markov Models

Learning too / Model Parameter Estimation

- Maximum likelihood / Expectation Maximization
- Variational inference
- Sampling
- Sequential Monte Carlo
  - ... for all, batch or online

Trajectory need not be "physical," could be an economic indicator, completely abstract, etc.

# Cool Trajectory Inference Application : Neural Decoding

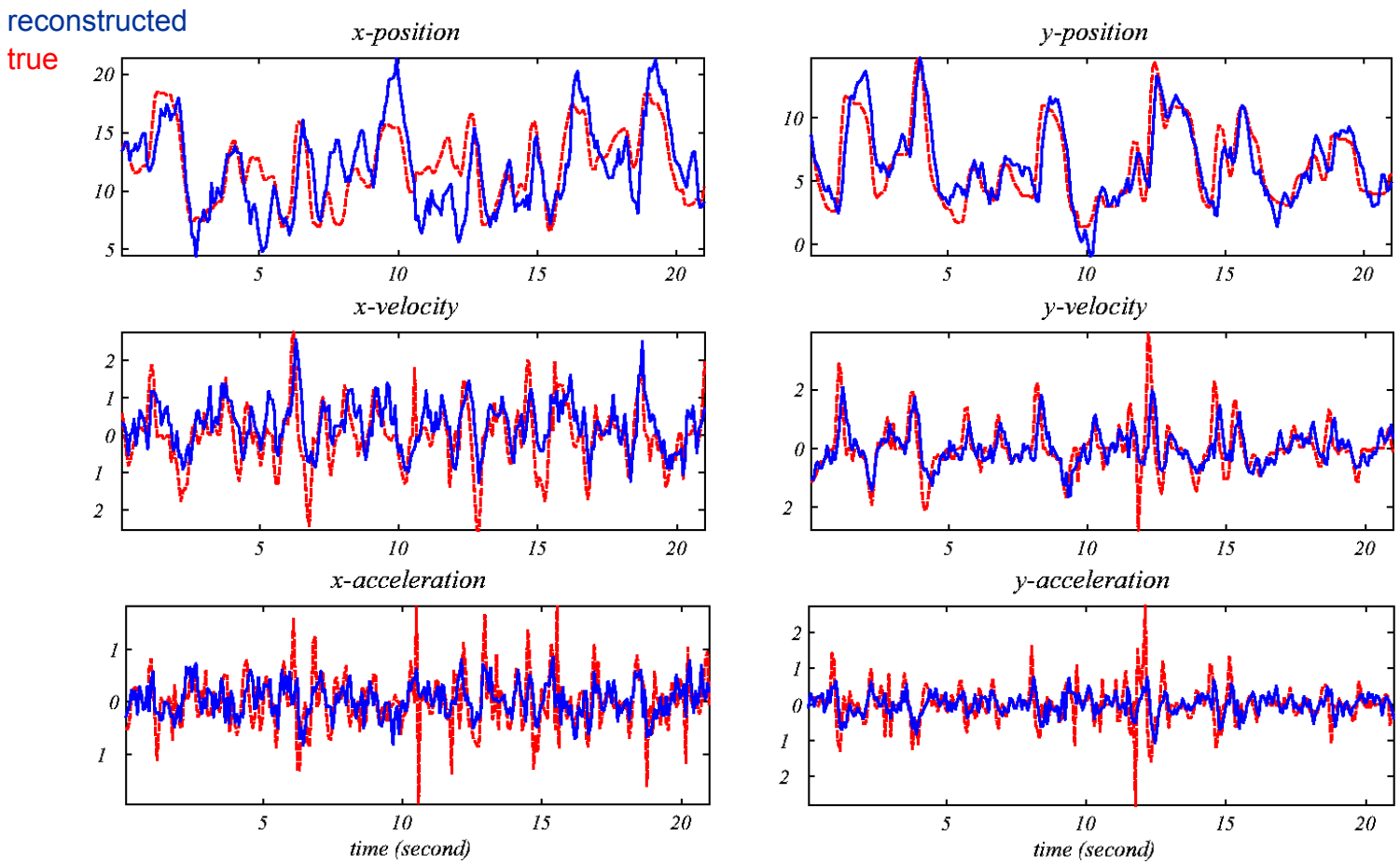


Figure: Actual and predicted hand positions (predicted from neural firing rates alone using a Kalman filter) [5]

# Another Application: Unsupervised Clustering

Forensic analysis of printed documents, infer printer used to print document from visual features.

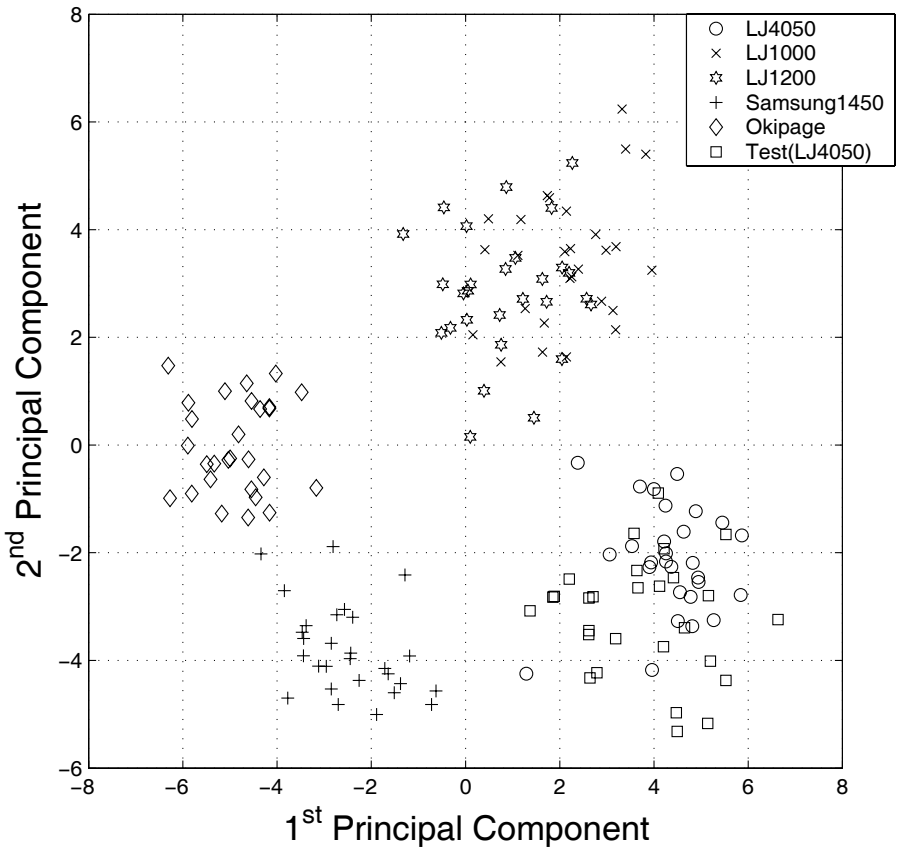


Figure: PCA projection of printer features [1]

# Another Unsupervised Clustering Application

Automatic discovery of number of neurons and assignment of waveforms to neurons. Essential to electrophysiological study of the brain.

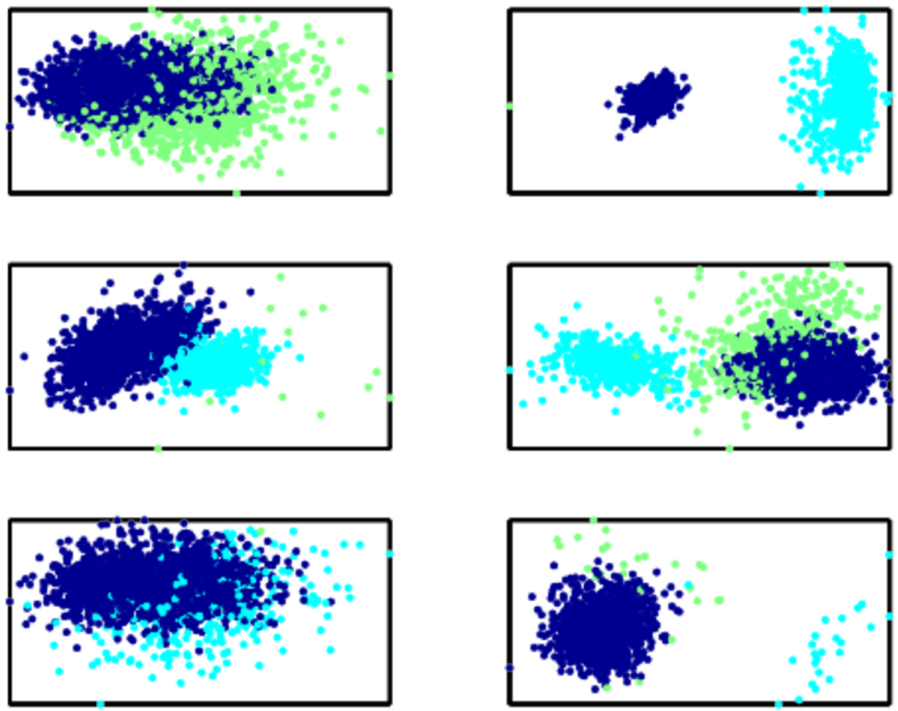

Figure: Automatically sorted action potential PCA projections [4]

# A Big Unsupervised Clustering Application

Multinomial mixture model automatic document clustering for information retrieval.

$$\begin{aligned} z_n|\boldsymbol{\pi} &\sim \text{Discrete}(\boldsymbol{\pi}) \\ \mathbf{x}_n|z_n = k, \boldsymbol{\Theta} &\sim \text{Multinomial}(\boldsymbol{\theta}_{z_n}) \end{aligned}$$

where $\mathbf{x}_n$ is a bag of words or feature representation of a document, $z_n$ is a per document class indicator variable, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_k\}_{k=1}^K$ is a collection of probability vectors over types (or features) (per cluster $k$), and $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K], \sum_k \pi_k = 1$ is the class prior.

Such a model can be used to cluster similar documents together for information retrieval (Google, Bing, etc.) purposes.

# Tools for Unsupervised Clustering

Known/hand-crafted model, inference only

- K-means
- Gaussian mixture models
- Multinomial mixture models

Learning too / Model Parameter Estimation

- Maximum likelihood / Expectation Maximization
- Variational inference
- Sampling
- Sequential Monte Carlo
  - ... for all, batch or online

# Tools for All

- Maximum likelihood / Expectation Maximization
- Variational inference
- Sampling
- Sequential Monte Carlo
  - ... for all, batch or online

# Bibliograpy I


[1] G.N. Ali, P.J. Chiang, A.K. Mikkilineni, G.T.C. Chiu, E.J. Delp, and J.P. Allebach. Application of principal components analysis and gaussian mixture models to printer identification. In *Proceedings of the IS&Ts NIP20: International Conference on Digital Printing Technologies*, volume 20, pages 301–305. Citeseer, 2004.

[2] I. Beinlich, H.J. Suermondt, R. Chavez, G. Cooper, et al. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. 256, 1989.

[3] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, NY, 2006.

[4] F. Wood and M. J. Black. A nonparametric Bayesian alternative to spike sorting. *Journal of Neuroscience Methods*, page to appear, 2008.

# Bibliograpy II

[5] W. Wu, M. J. Black, Y. Gao, E. Bienenstock, M. Serruya, and J. P. Donoghue. Inferring hand motion from multi-cell recordings in motor cortex using a Kalman filter. In *SAB'02-Workshop on Motor Control in Humans and Robots: On the Interplay of Real Brains and Artificial Devices*, pages 66–73, August 2002.