Understanding (and Predicting) Data

- Many different data streams around us
- We process, understand and respond
- What is the response based on?
  - The data we observed
  - Underlying characteristics that we inferred

Examples

- Stock Market
  ![Stock Market Graph]

  Market sentiment as a latent variable?

Examples

- Sports
  ![MONEYBALL poster]

  What skills in players should be valued?

Sidenote: For anyone interested, Baseball as a Markov Chain
Examples
- Many audio applications use latent variables
  - Signal Separation
  - Voice Modification
  - Music Analysis
  - Music and Speech Generation

A Strange Observation

Comments on the high-pitched singing
- Sarah McDonald (Holy Cow): “.. shrieking…”
- Khazana.com: “.. female Indian movie playback singers who can produce ultra high frequencies which only dogs can hear clearly.”
- www.roadjunky.com: “.. High pitched female singers doing their best to sound like they were seven years old…”

A Disturbing Observation

Let's Fix the Song
- The pitch is unpleasant
- The melody isn't bad
- Modify the pitch, but retain melody

Problem:
- Cannot just shift the pitch: will destroy the music
  - The music is fine, leave it alone
- Modify the singing pitch without affecting the music
“Personalizing” the Song

- Separate the vocals from the background music
  - Modify the separated vocals, keep music unchanged
- Separation need not be perfect
  - Must only be sufficient to enable pitch modification of vocals
  - Pitch modification is tolerant of low-level artifacts
    - For octave level pitch modification artifacts can be undetectable.

Some examples

- Example 1: Vocals shifted down by 4 semitones
- Example 2: Gender of singer partially modified

Techniques Employed

- Signal separation
  - Employed a simple latent-variable based separation method
- Voice modification
  - Equally simple techniques
  - Will consider the underlying methods over next few lectures
- Extensive use of Expectation Maximization

Learning Distributions for Data

- Problem: Given a collection of examples from some data, estimate its distribution
  - Basic ideas of Maximum Likelihood and MAP estimation can be found in Aarti/Paris’ slides
    - Pointed to in a previous class
- Solution: Assign a model to the distribution
  - Learn parameters of model from data
  - Models can be arbitrarily complex
    - Mixture densities, Hierarchical models
- Learning can be done using Expectation Maximization
A Thought Experiment

- A person shoots a loaded dice repeatedly
- You observe the series of outcomes
- You can form a good idea of how the dice is loaded
  - Figure out what the probabilities of the various numbers are for dice
    - \( P(\text{number}) = \text{count(\text{number})}/\text{sum(rolls)} \)
    - This is a maximum likelihood estimate
      - Estimate that makes the observed sequence of numbers most probable

Generative Model

- The data are generated by draws from the distribution
  - I.e. the generating process draws from the distribution
- Assumption: The distribution has a high probability of generating the observed data
  - Not necessarily true
- Select the distribution that has the highest probability of generating the data
  - Should assign lower probability to less frequent observations and vice versa

The Multinomial Distribution

- A probability distribution over a discrete collection of items is a Multinomial
  - \( P(X : X \text{ belongs to a discrete set}) = P(X) \)
- E.g. the roll of dice
  - \( X : X \in \{1, 2, 3, 4, 5, 6\} \)
- Or the toss of a coin
  - \( X : X \in \{\text{head}, \text{tails}\} \)

Maximum Likelihood Estimation: Multinomial

- Probability of generating \((n_1, n_2, n_3, n_4, n_5, n_6)\)
  - \( P(n_1, n_2, n_3, n_4, n_5, n_6) = \text{Const} \prod_i p_i^{n_i} \)
- Find \( p_1, p_2, p_3, p_4, p_5, p_6 \) so that the above is maximized
- Alternately maximize
  - \( \log(P(n_1, n_2, n_3, n_4, n_5, n_6)) = \log(\text{Const}) + \sum_i n_i \log(p_i) \)
  - \( \log() \) is a monotonic function
    - \( \text{argmax}_x f(x) = \text{argmax}_x \log(f(x)) \)
- Solving for the probabilities gives us
  - Requires constrained optimization to ensure probabilities sum to 1

Segue: Gaussians

- Parameters of a Gaussian:
  - Mean \( \mu \), Covariance \( \Theta \)

Maximum Likelihood: Gaussian

- Given a collection of observations \((X_1, X_2, \ldots)\), estimate mean \( \mu \) and covariance \( \Theta \)
  - \( P(X_1, X_2, \ldots) = \prod \frac{1}{\sqrt{2\pi}^{d/2} |\Theta|^{1/2}} \exp\left( -0.5 (X - \mu)^T \Theta^{-1} (X - \mu) \right) \)
  - \( \log(P(X_1, X_2, \ldots)) = \log(\text{Const}) + \sum_i \log(|\Theta|) + (X_i - \mu)^T \Theta^{-1} (X_i - \mu) \)
- Maximizing w.r.t. \( \mu \) and \( \Theta \) gives us
  - \( \mu = \frac{1}{N} \sum_i X_i \)
  - \( \Theta = \frac{1}{N} \sum_i (X_i - \mu)(X_i - \mu)^T \)
  - \( \text{EVENTUALLY ITS JUST COUNTING!} \)

\[ P(X) = N(X; \mu, \Theta) = \frac{1}{\sqrt{2\pi}^{d/2} |\Theta|^{1/2}} \exp\left( -0.5 (X - \mu)^T \Theta^{-1} (X - \mu) \right) \]
Laplacian

Parameters: Mean \( \mu \), scale \( b \) (\( b > 0 \))

\[
P(x) = L(x; \mu, b) = \frac{1}{2b} \exp\left(\frac{x - \mu}{b}\right)
\]

Maximum Likelihood: Laplacian

- Given a collection of observations \( (x_1, x_2, \ldots) \), estimate mean \( \mu \) and scale \( b \)

\[
\log(P(x_1, x_2, \ldots)) = C - N \log b - \frac{1}{N} \sum_{i=1}^{N} \left| \frac{x_i - \mu}{b} \right|
\]

- Maximizing w.r.t \( \mu \) and \( b \) gives us

\[
\mu = \frac{1}{N} \sum_{i=1}^{N} x_i, \quad b = \frac{1}{N} \sum_{i=1}^{N} \left| x_i - \mu \right|
\]

Dirichlet

- Parameters are \( \alpha \)
- Determine mode and curvature
- Defined only of probability vectors
- \( X = \{x_1, x_2, \ldots\} \), \( \sum_i x_i = 1, \quad x_i \geq 0 \) for all \( i \)

Maximum Likelihood: Dirichlet

- Given a collection of observations \( (X_1, X_2, \ldots) \), estimate \( \alpha \)

\[
\log(P(X_1, X_2, \ldots)) = \sum_i \log(\Gamma(\alpha_i)) - N \sum_i \log(\Gamma(\alpha_i)) - N \log\left(\sum_i \alpha_i\right)
\]

- No closed form solution for \( \alpha \)s.
- Needs gradient ascent
- Several distributions have this property: the ML estimate of their parameters have no closed form solution

Continuing the Thought Experiment

- Two persons shoot loaded dice repeatedly
  - The dice are differently loaded for the two of them
  - We observe the series of outcomes for both persons
- How to determine the probability distributions of the two dice?

Estimating Probabilities

- Observation: The sequence of numbers from the two dice
  - As indicated by the colors, we know who rolled what number
Estimating Probabilities

- Observation: The sequence of numbers from the two dice
  - As indicated by the colors, we know who rolled what number

- Segregation: Separate the blue observations from the red

<table>
<thead>
<tr>
<th>Collection of &quot;blue&quot; numbers</th>
<th>Collection of &quot;red&quot; numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 5 3 2 2 4</td>
<td>6 4 5 3 2 2 2</td>
</tr>
<tr>
<td>1 4 6 2 1 3 6 1</td>
<td>4 1 3 5 2 4 4 2 6</td>
</tr>
</tbody>
</table>

A Thought Experiment

- Now imagine that you cannot observe the dice yourself
- Instead there is a "caller" who randomly calls out the outcomes
  - 40% of the time he calls out the number from the left shooter, and 60% of the time, the one from the right (and you know this)

- At any time, you do not know which of the two he is calling out
- How do you determine the probability distributions for the two dice?

A Mixture Multinomial

- The caller will call out a number X in any given callout IF
  - He selects "RED", and the Red die rolls the number X
  - OR
  - He selects "BLUE" and the Blue die rolls the number X

- \( P(X) = P(\text{Red})P(X|\text{Red}) + P(\text{Blue})P(X|\text{Blue}) \)
  - E.g. \( P(6) = P(\text{Red})P(6|\text{Red}) + P(\text{Blue})P(6|\text{Blue}) \)

- A distribution that combines (or mixes) multiple multinomials is a mixture multinomial
  \[ P(X) = \sum P(Z)P(X|Z) \]

Mixture Distributions

- Mixtures distributions mix several component distributions
- Component distributions may be of varied type
- Mixing weights must sum to 1.0
- Component distributions integrate to 1.0
- Mixture distribution integrates to 1.0

P(number) = number of times number was rolled / total number of observed rolls
Maximum Likelihood Estimation

For our problem:

- \( Z = \) color of dice

\[
P(X) = \sum_Z P(Z)P(X | Z)
\]

Maximum likelihood solution: Maximize

\[
\log P(n_1, n_2, n_3, n_4, n_5, n_6) = \log \text{Const} + \sum_i n_i \log \left( \sum_Z P(Z | X) \right)
\]

No closed form solution (summation inside log)!

- In general ML estimates for mixtures do not have a closed form
- USE EM!

Expectation Maximization

- Iterative solution
- Get some initial estimates for all parameters
  - Dice shooter example: This includes probability distributions for dice AND the probability with which the caller selects the dice
- Two steps that are iterated:
  - **Expectation Step:** Estimate statistically, the values of unseen variables
  - **Maximization Step:** Using the estimated values of the unseen variables as truth, estimates of the model parameters

EM: The auxiliary function

- EM iteratively optimizes the following auxiliary function
  \[
  Q(0, 0') = \sum_Z P(Z | X, 0') \log P(Z, X | 0) \\
  \]
  - \( Z \) are the unseen variables
  - Assuming \( Z \) is discrete (may not be)
  - \( 0 \) are the parameter estimates from the previous iteration
  - \( 0' \) are the estimates to be obtained in the current iteration

Fragmenting the Observation

- EM is an iterative algorithm
  - At each time there is a current estimate of parameters
- The “size” of the fragments is proportional to the **a posteriori probability** of the component distributions
  - The **a posteriori** probabilities of the various values of \( Z \) are computed using Bayes’ rule:
    \[
    P(Z | X) = \frac{P(X | Z)P(Z)}{P(X)} = CP(X | Z)P(Z)
    \]
- Every dice gets a fragment of size \( P(\text{dice} | \text{number}) \)
Expectation Maximization

- Hypothetical Dice Shooter Example:
  - We obtain an initial estimate for the probability distribution of the two sets of dice (somehow):
  - We obtain an initial estimate for the probability with which the caller calls out the two shooters (somehow)

Expectation Maximization

- Hypothetical Dice Shooter Example:
  - Initial estimate:
    - $P(\text{blue}) = P(\text{red}) = 0.5$
    - $P(4 \mid \text{blue}) = 0.1$, for $P(4 \mid \text{red}) = 0.05$
  - Caller has just called out 4
  - Posterior probability of colors:
    - $P(\text{red} \mid X = 4) = CP(\text{red} \mid X = 4) P(Z) = C \times 0.05 \times 0.5 = 0.025$
    - $P(\text{blue} \mid X = 4) = CP(\text{blue} \mid X = 4) P(Z) = C \times 0.1 \times 0.5 = 0.05$
    - Normalizing: $P(\text{red} \mid X = 4) = 0.33$, $P(\text{blue} \mid X = 4) = 0.67$

Expectation Maximization

- Every observed roll of the dice contributes to both “Red” and “Blue”

Expectation Maximization

- Every observed roll of the dice contributes to both “Red” and “Blue”
Every observed roll of the dice contributes to both “Red” and “Blue”

- Red:
  - Total count for 1: 1.71
  - Total count for 2: 0.56

- Blue:
  - Total count for 1: 1.71
  - Total count for 2: 0.56
  - Total count for 3: 0.66
Blue:
- Total count for 1: 1.29
- Total count for 2: 3.44
- Total count for 3: 1.34
- Total count for 4: 2.68
- Total count for 5: 1.34
- Total count for 6: 0.6

Updated probability of Blue dice:
- $P(1 | \text{Blue}) = 1.29/11.69 = 0.122$
- $P(2 | \text{Blue}) = 0.56/11.69 = 0.057$
- $P(3 | \text{Blue}) = 0.66/11.69 = 0.056$
- $P(4 | \text{Blue}) = 1.32/11.69 = 0.114$
- $P(5 | \text{Blue}) = 0.66/11.69 = 0.056$
- $P(6 | \text{Blue}) = 2.40/11.69 = 0.206$

Blue:
- Total instances = 18
- $P(Z=\text{Red}) = 7.31/18 = 0.41$
- $P(Z=\text{Blue}) = 10.69/18 = 0.59$

Red or Blue
- Total count for "Red": 7.31
- Total count for "Blue": 10.69
- Total instances = 18
- Note 7.31+10.69 = 18
- We also revise our estimate for the probability that the caller calls out Red or Blue
- i.e the fraction of times that he calls Red and the fraction of times he calls Blue

$P(\text{Red}|X)$
- $\text{Red}$: 0.57
- $\text{Red}$: 0.33
- $\text{Red}$: 0.14
- $\text{Red}$: 0.86
- $\text{Red}$: 0.67

$P(\text{Blue}|X)$
- $\text{Blue}$: 0.57
- $\text{Blue}$: 0.33
- $\text{Blue}$: 0.14
- $\text{Blue}$: 0.86
- $\text{Blue}$: 0.67

Oct 2, 2012
1756/14797
**The Dice Shooter Example**

- Initialize $P(Z)$, $P(X | Z)$
- Estimate $P(Z | X)$ for each $Z$, for each called out number
  - Associate $X$ with each value of $Z$, with weight $P(Z | X)$
  - Re-estimate $P(X | Z)$ for every value of $X$ and $Z$
- Re-estimate $P(Z)$
  - If not converged, return to 2

**Solutions may not be unique**

- The EM algorithm will give us one of many solutions, all equally valid!
  - The probability of 6 being called out:
    \[
    P(6) = \alpha P(6 | \text{red}) + \beta P(6 | \text{blue}) = \alpha P_1 + \beta P_6
    \]
    - Assigns $P_1$ as the probability of 6 for the red die
    - Assigns $P_6$ as the probability of 6 for the blue die
  - The following too is a valid solution
    \[
    P(6) = 1.0 (\alpha P_1 + \beta P_6) + 0.0 \text{anything}
    \]
    - Assigns 1.0 as the a priori probability of the red die
    - Assigns 0.0 as the probability of the blue die
  - The solution is NOT unique

**In Squiggles**

- Given a sequence of observations $O_1, O_2, ..$
  - $N_k$ is the number of observations of number $X$
- Initialize $P(Z)$, $P(X | Z)$ for dice $Z$ and numbers $X$
- Iterate:
  - For each number $X$:
    \[
    P(Z | X) = \frac{P(X | Z) P(Z)}{\sum_{Z'} P(Z') P(X | Z')}
    \]
- Update:
  \[
  P(X | Z) = \frac{\prod_{k=1}^{N_k} P(Z | X)}{\sum_{Z'} \prod_{k=1}^{N_k} P(Z | X)}
  \]

**The updated values**

- **Probability of Red dice:**
  - $P(1 | \text{Red}) = 0.17/0.234$
  - $P(2 | \text{Red}) = 0.56/0.077$
  - $P(3 | \text{Red}) = 0.66/0.090$
  - $P(4 | \text{Red}) = 1.32/0.181$
  - $P(5 | \text{Red}) = 0.66/0.090$
  - $P(6 | \text{Red}) = 2.40/0.328$

- **Probability of Blue dice:**
  - $P(1 | \text{Blue}) = 1.29/0.122$
  - $P(2 | \text{Blue}) = 0.56/0.322$
  - $P(3 | \text{Blue}) = 0.66/0.125$
  - $P(4 | \text{Blue}) = 1.32/0.250$
  - $P(5 | \text{Blue}) = 0.66/0.125$
  - $P(6 | \text{Blue}) = 2.40/0.056$

- $P(Z=\text{Red}) = 7.31/18 = 0.41$
- $P(Z=\text{Blue}) = 10.69/18 = 0.59$

**A More Complex Model**

- Gaussian mixtures are often good models for the distribution of multivariate data
- Problem: Estimating the parameters, given a collection of data

**Gaussian Mixtures: Generating model**

\[
P(X) = \sum_k P(k) N(X; \mu_k, \Theta_k)
\]

- The caller now has two Gaussians
  - At each draw he randomly selects a Gaussian, by the mixture weight distribution
  - He then draws an observation from that Gaussian
  - Much like the dice problem (only the outcomes are now real numbers and can be anything)
Observation: A collection of numbers drawn from a mixture of 2 Gaussians
- As indicated by the colors, we know which Gaussian generated what number

Segregation: Separate the blue observations from the red

From each set compute parameters for that Gaussian

$$\mu_{red} = \frac{1}{N_{red}} \sum_{X \in \text{red}} X$$
$$\sigma_{red}^2 = \frac{1}{N_{red}} \sum_{X \in \text{red}} (X - \mu_{red})^2$$

Compute fragment sizes for each Gaussian, for each observation

$$P(k \mid X) = \frac{P(k)N(X; \mu_k, \Theta_k)}{\sum_{k'} P(k')N(X; \mu_{k'}, \Theta_{k'})}$$

Expectation Maximization

- Initialize $$P(k)$$, $$\mu_k$$ and $$\Theta_k$$ for both Gaussians
  - Important how we do this
  - Typical solution: Initialize means randomly, $$\Theta_k$$ as the global covariance of the data and $$P(k)$$ uniformly

- Update mixture weights, means and variances for all Gaussians

$$P(k) = \frac{\sum_i P(k | x_i) \mu_k}{N}$$
$$\mu_k = \frac{\sum_i P(k | x_i) x_i}{N}$$
$$\Theta_k = \frac{\sum_i P(k | x_i) (x_i - \mu_k)^2}{\sum_i P(k | x_i)}$$

- If not converged, return to 2

EM for Gaussian Mixtures

1. Initialize $$P(k)$$, $$\mu_k$$ and $$\Theta_k$$ for all Gaussians
2. For each observation $$X$$ compute a posteriori probabilities for all Gaussian

$$P(k \mid X) = \frac{P(k)N(X; \mu_k, \Theta_k)}{\sum_{k'} P(k')N(X; \mu_{k'}, \Theta_{k'})}$$

3. Update mixture weights, means and variances for all Gaussians

$$P(k) = \frac{\sum_i P(k | x_i) \mu_k}{N}$$
$$\mu_k = \frac{\sum_i P(k | x_i) x_i}{N}$$
$$\Theta_k = \frac{\sum_i P(k | x_i) (x_i - \mu_k)^2}{\sum_i P(k | x_i)}$$

- If not converged, return to 2

EM estimation of Gaussian Mixtures

- An Example

Expectation Maximization

- Each observation contributes only as much as its fragment size to each statistic
- Mean(red) = (6.1*0.81 + 1.4*0.33 + 5.3*0.75 + 1.9*0.41 + 4.2*0.64 + 2.2*0.43 + 4.9*0.66 + 6.1*0.05) / (0.81 + 0.33 + 0.75 + 0.41 + 0.64 + 0.43 + 0.66 + 0.05) = 17.05 / 4.08 = 4.18
- Var(red) = (6.1*0.81^2 + 1.4*0.33^2 + 5.3*0.75^2 + 1.9*0.41^2 + 4.2*0.64^2 + 2.2*0.43^2 + 4.9*0.66^2 + 6.1*0.05^2) / (0.81 + 0.33 + 0.75 + 0.41 + 0.64 + 0.43 + 0.66 + 0.05)

The identity of the Gaussian is not known!

Solution: Fragment the observation

Fragment size proportional to a posteriori probability

$$P(k \mid X) = \frac{P(X \mid k)P(k)}{\sum_k P(k')P(X \mid k')}$$
Expectation Maximization

- The same principle can be extended to mixtures of other distributions.

- E.g. Mixture of Laplacians: Laplacian parameters become

$$
\mu_k = \sum_x P(k \mid x) \sum_x P(k \mid x) x \\
\beta_k = \sum_x P(k \mid x) \sum_x P(k \mid x) |x - \mu_k|
$$

In a mixture of Gaussians and Laplacians, Gaussians use the Gaussian update rules, Laplacians use the Laplacian rule.

Solve this problem:

- Caller rolls a dice and flips a coin
- He calls out the number rolled if the coin shows head
- Otherwise he calls the number+1
- Determine p(heads) and p(number) for the dice from a collection of outputs

- Caller rolls two dice
- He calls out the sum
- Determine P(dice) from a collection of outputs

The dice and the coin

- Unknown: Whether it was head or tails

The two dice

- Unknown: How to partition the number

  - Count\textsubscript{blue}(3) += P(3,1 | 4)
  - Count\textsubscript{blue}(2) += P(2,2 | 4)
  - Count\textsubscript{blue}(1) += P(1,3 | 4)

Fragmentation can be hierarchical

- E.g. mixture of mixtures
- Fragments are further fragmented.. Work this out
More later

- Will see a couple of other instances of the use of EM
- Work out HMM training
  - Assume state output distributions are multinomials
  - Assume they are Gaussian
  - Assume Gaussian mixtures