Machine Learning for Signal Processing
Expectation Maximization
Mixture Models

Bhiksha Raj
Class 10. 3 Oct 2013

Administrivia

- HW2 is up
  - A final problem will be added
  - You have four weeks
  - It’s a loooooong homework
  - About 12-24 hours of work

- Does everyone have teams/project proposals

- Begin working on your projects immediately.

A Strange Observation

The pitch of female Indian playback singers is on an ever-increasing trajectory

- Mean pitch values: 278Hz, 410Hz, 580Hz

I’m not the only one to find the high-pitched stuff annoying

- 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

The pitch of female Indian playback singers is on an ever-increasing trajectory

- Mean pitch values: 278Hz, 410Hz, 580Hz

Lets 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.

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
- Separation: Extensive use of Expectation Maximization

Learning Distributions for Data

- Problem: Given a collection of examples from some data, estimate its distribution
- Solution: Assign a model to the distribution
  - Learn parameters of model from data
- Models can be arbitrarily complex
  - Mixture densities, Hierarchical models.
- Learning must be done using Expectation Maximization
- Following slides: An intuitive explanation using a simple example of multinomials
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(number) = count(number)/sum(rolls)
- This is a maximum likelihood estimate
  - Estimate that makes the observed sequence of numbers most probable

3 Oct 2011 1:1756/10797 12

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 \( \{\text{head, tails}\} \)

3 Oct 2011 1:1756/10797 14

Maximum Likelihood Estimation

- Basic principle: Assign a form to the distribution
  - E.g. a multinomial
  - Or a Gaussian
- Find the distribution that best fits the histogram of the data

3 Oct 2011 1:1756/10797 16

Defining “Best Fit”

- The data are generated by draws from the distribution
  - i.e. the generating process draws from the distribution
- Assumption: The world is a boring place
  - The data you have observed are very typical of the process
- Consequent 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

3 Oct 2011 1:1756/10797 18

Maximum Likelihood Estimation: Multinomial

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

3 Oct 2011 1:1756/10797 17

Segue: Gaussians

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

3 Oct 2011 1:1756/10797 18
**Maximum Likelihood: Gaussian**

- Given a collection of observations \( \{X_1, X_2, \ldots\} \), estimate mean \( \mu \) and covariance \( \Theta \)

\[
P(X_1, X_2, \ldots) = \prod \frac{1}{\sqrt{2\pi} \sqrt{\det(\Theta)}} \exp\left(-\frac{1}{2} (X_i - \mu)^T \Theta^{-1} (X_i - \mu)\right)
\]

\[
\log(P(X_1, X_2, \ldots)) = C - 0.5 \sum \text{log}(\Theta) - 0.5 (X_i - \mu)^T \Theta^{-1} (X_i - \mu)
\]

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

\[
\mu = \frac{1}{N} \sum X_i, \quad \Theta = \frac{1}{N} \sum (X_i - \mu)(X_i - \mu)^T
\]

**Maximum Likelihood: Laplacian**

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

\[
\log(P(x_1, x_2, \ldots)) = C - N \log(b) - \frac{1}{b} \sum |x_i - \mu|
\]

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

\[
\mu = \frac{1}{N} \sum x_i, \quad b = \frac{1}{N} \sum |x_i - \mu|
\]

**Maximum Likelihood: Dirichlet**

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

\[
\log(P(X_1, X_2, \ldots)) = \sum (\alpha_i - 1) \log(X_i) + N \sum \log(\Gamma(\alpha_i)) - N \log(\Gamma(\sum \alpha_i))
\]

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

**Laplacian**

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

**Dirichlet**

- From Wikipedia

\[
P(X) = D(X; \alpha) = \frac{\prod \Gamma(\alpha_i)}{\Gamma(\sum \alpha_i)} \prod \alpha_i^{x_i}
\]

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

**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
- Segregation: Separate the blue observations from the red
- From each set compute probabilities for each of the 6 possible outcomes

\[
P(\text{number}) = \frac{\text{no of times number was rolled}}{\text{total number of observed rolls}}
\]

A Thought Experiment

- 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
- From each set compute probabilities for each of the 6 possible outcomes

\[
P(\text{number}) = \frac{\text{no of times number was rolled}}{\text{total number of observed rolls}}
\]

A Thought Experiment

- How do you now determine the probability distributions for the two sets of dice ...
- If you do not even know what fraction of time the blue numbers are called, and what fraction are red?

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

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

Mixture weights Component multinomials
Mixture Distributions

\[ P(X) = \sum_z P(Z)P(X \mid Z) \]

Mixture of Gaussians and Laplacians

\[ P(X) = \sum_z P(Z)N(X; \mu_z, \theta_z) + \sum_z P(Z)\Delta(X; \mu_z, b_z) \]

- Mixture 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

Maximum Likelihood Estimation

- For our problem:
  \[ P(X) = \sum_z P(Z)P(X \mid Z) \]
  \[ \quad Z = \text{color of dice} \]
  \[ P(n_i, n_j, n_k, n_l, n_m) = \text{Constr} \left( \sum P(X) \right) = \text{Constr} \left( \sum P(Z)P(X \mid Z) \right) \]
- Maximum likelihood solution: Maximize
  \[ \log(P(n_i, n_j, n_k, n_l, n_m)) = \log(\text{Constr}) + \sum n_i \log(\sum P(Z)P(X \mid Z)) \]
- No closed form solution (summation inside log!)
  - In general ML estimates for mixtures do not have a closed form
  - USE EM!

Expectation Maximization

- It is possible to estimate all parameters in this setup using the Expectation Maximization (or EM) algorithm
- First described in a landmark paper by Dempster, Laird and Rubin
- Much work on the algorithm since then
- The principles behind the algorithm existed for several years prior to the landmark paper, however.

EM: The auxiliary function

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

Expectation Maximization as counting

- Hidden variable: Z
  - Dice: The identity of the dice whose number has been called out
- If we knew Z for every observation, we could estimate all terms
  - By adding the observation to the right bin
- Unfortunately, we do not know Z – it is hidden from us!
- Solution: FRAGMENT THE OBSERVATION
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:
    — \( P(\text{blue}) = P(\text{red}) = 0.5 \)
    — \( P(4 | \text{blue}) = 0.1 \), for \( P(4 | \text{red}) = 0.05 \)
• Caller has just called out 4
• Posterior probability of colors:
  \[
P(\text{red} | X = 4) = CP(X = 4 | Z = \text{red})P(Z = \text{red}) = C \times 0.05 \times 0.5 = C \times 0.025
\]
  \[
P(\text{blue} | X = 4) = CP(X = 4 | Z = \text{blue})P(Z = \text{blue}) = C \times 0.1 \times 0.5 = C \times 0.05
\]
  Normalized: \( P(\text{red} | X = 4) = 0.33 \), \( P(\text{blue} | X = 4) = 0.67 \)

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”
**Expectation Maximization**

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

<table>
<thead>
<tr>
<th>Red</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 (0.8), 4 (0.33)</td>
<td>6 (0.2), 4 (0.67)</td>
</tr>
</tbody>
</table>

**Expectation Maximization**

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

<table>
<thead>
<tr>
<th>Red</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 (0.8), 4 (0.33), 5 (0.33), 1 (0.57), 2 (0.14), 3 (0.33), 4 (0.33), 2 (0.14), 2 (0.14), 1 (0.57), 4 (0.33), 3 (0.33), 4 (0.33), 6 (0.8), 2 (0.14), 1 (0.57), 6 (0.8)</td>
<td>6 (0.2), 4 (0.67), 5 (0.67), 1 (0.43), 4 (0.67), 3 (0.67), 6 (0.2), 2 (0.86), 1 (0.43), 6 (0.2)</td>
</tr>
</tbody>
</table>

**Expectation Maximization**

- Total count for "Red" : 7.31
- Red:
  - Total count for 1: 1.71

<table>
<thead>
<tr>
<th>Red</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

**Expectation Maximization**

- Total count for "Red" : 7.31
- Red:
  - Total count for 1: 1.71
  - Total count for 2: 0.56

<table>
<thead>
<tr>
<th>Red</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>
**Expectation Maximization**

- Total count for "Red": 7.31
- Red:
  - Total count for 1: 1.71
  - Total count for 2: 0.56
  - Total count for 3: 0.66

**Expectation Maximization**

- Total count for "Red": 7.31
- Red:
  - Total count for 1: 1.71
  - Total count for 2: 0.56
  - Total count for 3: 0.66
  - Total count for 4: 1.32
  - Total count for 5: 0.66

**Expectation Maximization**

- Total count for "Red": 7.31
- Red:
  - Total count for 1: 1.71
  - Total count for 2: 0.56
  - Total count for 3: 0.66
  - Total count for 4: 1.32
  - Total count for 5: 0.66
  - Total count for 6: 2.4

**Expectation Maximization**

- Total count for "Red": 7.31
- Red:
  - Total count for 1: 1.71
  - Total count for 2: 0.56
  - Total count for 3: 0.66
  - Total count for 4: 1.32
  - Total count for 5: 0.66
  - Total count for 6: 2.4

**Expectation Maximization**

- Total count for "Red": 7.31
- Red:
  - Total count for 1: 1.66
  - Total count for 2: 0.56
  - Total count for 3: 0.66
  - Total count for 4: 1.32
  - Total count for 5: 0.66
  - Total count for 6: 2.4

**Updated probability of Red dice:**

- \( P(1 | \text{Red}) = 1.71/7.31 = 0.234 \)
- \( P(2 | \text{Red}) = 0.56/7.31 = 0.077 \)
- \( P(3 | \text{Red}) = 0.66/7.31 = 0.090 \)
- \( P(4 | \text{Red}) = 1.32/7.31 = 0.181 \)
- \( P(5 | \text{Red}) = 0.66/7.31 = 0.090 \)
- \( P(6 | \text{Red}) = 2.40/7.31 = 0.328 \)
### Expectation Maximization

- **Total count for “Blue”**: 10.69
- **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:**
- $P(B|\text{Blue}) = 0.29/11.69 = 0.024$
- $P(B|\text{Blue}) = 0.33/11.69 = 0.028$
- $P(B|\text{Blue}) = 0.13/11.69 = 0.011$
- $P(B|\text{Blue}) = 1.32/11.69 = 0.114$
- $P(B|\text{Blue}) = 0.66/11.69 = 0.056$
**Expectation Maximization**

- 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
- \( P(Z|Red) = 7.31/18 = 0.41 \)
- \( P(Z|Blue) = 10.69/18 = 0.59 \)

**The updated values**

- Probability of Red dice:
  - \( P(1|Red) = 1.71/7.31 = 0.234 \)
  - \( P(2|Red) = 0.56/7.31 = 0.077 \)
  - \( P(3|Red) = 0.68/7.31 = 0.090 \)
  - \( P(4|Red) = 5.32/7.31 = 0.181 \)
  - \( P(5|Red) = 0.66/7.31 = 0.090 \)
  - \( P(6|Red) = 2.40/7.31 = 0.328 \)

- Probability of Blue dice:
  - \( P(1|Blue) = 1.29/11.69 = 0.122 \)
  - \( P(2|Blue) = 0.56/11.69 = 0.322 \)
  - \( P(3|Blue) = 0.66/11.69 = 0.125 \)
  - \( P(4|Blue) = 1.32/11.69 = 0.250 \)
  - \( P(5|Blue) = 0.66/11.69 = 0.125 \)
  - \( P(6|Blue) = 2.40/11.69 = 0.056 \)

**The updated values can be used to repeat the process. Estimation is an iterative process.**

---

**The Dice Shooter Example**

- \( 6 4 1 5 3 2 2 2 ... \)
- \( 6 3 1 5 4 1 2 4 ... \)
- \( 4 4 1 6 3 2 1 2 ... \)

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

---

**In Squiggle**

- Given a sequence of observations \( O_1, O_2, ... \)
  - \( N_p \) is the number of observations of number \( X \)
- Initialize \( P(Z), P(X|Z) \) for dice \( Z \) and numbers \( X \)
- Iterate:
  - For each number \( X \):
    \[ P(Z|X) = \frac{\sum Z P(Z)P(X|Z)}{\sum Z P(Z)P(X|Z)} \]
  - Update:
    \[ P(X|Z) = \frac{\sum Z P(Z|X)P(Z)}{\sum Z P(Z|X)P(Z)} \]

---

**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:
    \[ P(6) = \alpha P(6|red) + \beta P(6|blue) = \alpha P_r + \beta P_b \]
    - Assigns \( P_r \) as the probability of 6 for the red die
    - Assigns \( P_b \) as the probability of 6 for the blue die
  - The following too is a valid solution [FX]
    \[ P(6) = 1.0(\alpha P_r + \beta P_b) + 0.0\text{anything} \]
    - 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

---

**A more complex model: Gaussian mixtures**

- A Gaussian mixture can represent data distributions far better than a simple Gaussian
  - The two panels show the histogram of an unknown random variable
  - The first panel shows how it is modeled by a simple Gaussian
  - The second panel models the histogram by a mixture of two Gaussians
  - Caveat: It is hard to know the optimal number of Gaussians in a mixture
A More Complex Model

\[ P(X) = \sum_{k} P(k|X) P(k) = \sum_{k} \frac{P(k|X)}{N} P(k) \exp\left(-\frac{1}{2}(X - \mu_k)^T \Sigma_k^{-1} (X - \mu_k)\right) \]

- 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

\[ P(X) = \sum_{k} P(k|X; \mu_k, \Theta_k) \]

- 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)

Estimating GMM with complete information

- 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

\[ \mu_w = \frac{1}{N} \sum_i x_i \quad \Theta_w = \frac{1}{N} \sum_i (x_i - \mu_w)(x_i - \mu_w)^T \]

\[ P(\text{red}) = \frac{N_r}{N} \]

Gaussian Mixtures: Generating model

\[ P(X) = \sum_{k} P(k|X; \mu_k, \Theta_k) \]

- Problem: In reality we will not know which Gaussian any observation was drawn from.
  - The color information is missing

Fragmenting the observation

- The identity of the Gaussian is not known!
- Solution: Fragment the observation
- Fragment size proportional to a posteriori probability

\[ P(k|X) = \frac{P(X|k)P(k)}{\sum_k P(X|k)P(k)} \]

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
- Compute fragment sizes for each Gaussian, for each observation

\[ P(k|X) = \frac{P(k|X; \mu_k, \Theta_k)}{\sum_k P(k|X; \mu_k, \Theta_k)} \]
### Expectation Maximization

- Each observation contributes only as much of its fragment size to each statistic
- Mean(red) = \([6.1^{0.81} + 1.4^{0.33} + 3.3^{0.75} + 4.2^{0.41} + 4.2^{0.43} + 4.9^{0.66} + 0.5^{0.05}] / (0.81 + 0.33 + 0.75 + 0.41 + 0.64 + 0.43 + 0.66 + 0.05)\) = 17.05 / 4.08 ≈ 4.18
- Var(red) = \([(6.1-4.18)^2*0.81 + (1.4-4.18)^2*0.33 + (3.3-4.18)^2*0.75 + (4.2-4.18)^2*0.41 + (4.2-4.18)^2*0.43 + (4.9-4.18)^2*0.66 + (0.5-4.18)^2*0.05)] / (0.81 + 0.33 + 0.75 + 0.41 + 0.64 + 0.43 + 0.66 + 0.05)

\[
P(\text{red}) = \frac{4.08}{4} = 1.02
\]

### 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

\[
\mu_k = \frac{\sum_k P(k)X}{\sum_k P(k)}
\]

\[
\Theta_k = \frac{\sum_k P(k)(X - \mu_k)^2}{\sum_k P(k)}
\]

4. If not converged, return to 2

### Expectation Maximization

- The same principle can be extended to mixtures of other distributions.
- E.g. Mixture of Laplacians: Laplacian parameters become

\[
\mu_k = \frac{1}{\sum_k P(k)X} \sum_k P(k)X
\]

\[
\theta_k = \frac{1}{\sum_k P(k)} \sum_k P(k)(X - \mu_k)^2
\]

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

### EM estimation of Gaussian Mixtures

- An Example

### Solve this problem:

- **Problem 1:**
  - 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

- **Problem 2:**
  - 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
- $\text{Count}_{\text{blue}}(3) += P(3,1 \mid 4)$
- $\text{Count}_{\text{blue}}(2) += P(2,2 \mid 4)$
- $\text{Count}_{\text{blue}}(1) += P(1,3 \mid 4)$

---

**Fragmentation can be hierarchical**

$$P(X) = \sum_r P(r) \sum_k P(Z \mid k) P(X \mid Z, k)$$

- 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
- EM for signal representation: PCA and factor analysis
- EM for signal separation
- EM for parameter estimation
- EM for homework..