Lang and Stats
Maximum Likelihood
Bhiksha Raj
A Thought Experiment

• A person flips a loaded coin repeatedly
• You observe the series of outcomes
• You must estimate the probability of heads for the coin

H T H H T H T T
Choose the distribution

- Top: Actual histogram of outcomes
- Bottom: Two candidate probability distributions
- Which one is the correct one?
- Why?
A Thought Experiment

- A person shoots a loaded dice repeatedly
- You observe the series of outcomes
- You must estimate the probability distribution for the dice

6 3 1 5 4 1 2 4 ...
Choose the distribution

- Top: Actual histogram of outcomes
- Bottom: Two candidate probability distributions
- Which one is the correct one?
- Why?
Choose the Gaussian

• A number of observations are drawn from a Gaussian PDF
  – The PDF must be estimated from the observations
• The blue patch shows the histogram of outcomes
• The three curves show three candidate Gaussian PDFs
• Which of the three is the correct one
  – Why?
Maximum Likelihood Principle
(AKA the most boring principle)

• 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
    • The world behaving exactly as it must
    – There are no surprises!

• Consequent assumption: The distribution has a high probability of generating the observed data

• Select the distribution that has the highest probability of generating the data
Maximum Likelihood Principle
(AKA the most boring principle)

\[ \hat{\theta}_{ML} = \arg \max_{\theta} L(X; \theta) \]

- Choose the \( \theta \) which maximizes the likelihood of the observed data
  - Maximize the ordinariness of the data
Most boring recipe

- Identify (or propose) the parametric probability distribution for the data class $P_X(X; \theta)$
  - This is a function of both data $x$ and parameters $\theta$

- Write down a likelihood function $L(X; \theta)$ for the observed data
  - Likelihood is just the probability distribution, with the observations plugged into the value of the variable $X$

- Estimate $\theta$ as
  $$\hat{\theta}_{ML} = \arg\max_{\theta} L(X; \theta)$$

- Actual maximization may require expensive optimization
  - Or may not even be possible
    - In some cases even writing down $L(X; \theta)$ may not be possible
Try this

• MLE of Bernoulli distribution (coin toss) given a series of outcomes:

  H  T  H  T  T  H  ....  \( N_1 \) Heads, \( N_2 \) Tails

• MLE of a multinomial (e.g. six-sided die) given a series of outcomes

• MLE of a Gaussian given a set of draws \( X_1, X_2, ..., X_N \)
MLE Existence and Computability

• A maximum likelihood estimate may not always exist
  – E.g. the MLE of a logistic regression, when classes are perfectly separable

• Even if it exists, it may not be computable
  – E.g. \( P(X; \theta) = G(X, U(\theta)) \) where \( U(\theta) \) is an uncomputable function
MLE: Existence

• Necessary condition for MLE to exist
  – There must be a bounded $\theta$ that satisfies
    \[ \frac{\partial L(X;\theta)}{\partial \theta} = 0 \]
    
    • And $\frac{\partial^2 L(X;\theta)}{\partial \theta^2} < 0$ must hold at that $\theta$

• Sufficient conditions for MLE to exist
  – The parameter space is compact
  – The Likelihood function is continuous on parameter space
MLE: uniqueness

• Sufficient condition for MLE to be unique
  – Likelihood function is *strictly concave*
  – Parameter space is convex

• More frequently MLE estimates may exist and be computable, but not identifiable
  – E.g. Mixture model, permuting the mixture components results in the same distribution
Properties of MLE

• The MLE is generally **unbiased**
  – Binomial, multinomial, mean of Gaussian are unbiased
  – Even when locally biased, it is **asymptotically unbiased**
    • E.g. variance of a Gaussian

• The MLE is **consistent**

\[
\lim_{N \to \infty} P(\hat{\theta}_{ML} - \theta = 0) \to 1
\]

• **Asymptotically efficient**

\[
\lim_{N \to \infty} \text{var}(\hat{\theta}_{ML}) = I_\theta^{-1}
\]

  – Given enough data, the MLE has the lowest possible variance
    • Achieves the Cramer-Rao bound
MLE caveat

• All of the previous properties only hold if the functional form of the likelihood accurately characterizes the *true* distribution of $X$

• The log likelihood often has local maxima
  – In general, the estimates obtained are *local* maxima

• For finite sample data, MLE can be badly affected by *surprising* samples
  – E.g. not observing any samples of a value with non-zero probability
  – All observations of language are surprising!
MLE problems

• Rare events are poorly represented

• Rare events that occur in training data are viewed as not being rare

• Rare events that occur in test data may not occur in training data
  – They will be assigned 0 probability
  – This makes our actual test data impossible according to our model!
MLE problems

• Probability ordering of low-probability events is not retained

• If $\text{Count}(V_1) > \text{Count}(V_2)$, this does not imply that $P(V_1) > P(V_2)$
  
  – On the other hand, for ML estimates, if $\text{Count}(V_1) > \text{Count}(V_2)$, the ML estimates will always have $\hat{P}(V_1) > \hat{P}(V_2)$
MLE problems

• Individually unbiased estimates can be cumulatively biased!
  
  ─ I.e. make incorrect predictions about the future

• Lets see why..
The dover mail

The **Dover** mail was in its usual genial position that the guard suspected the passengers, the passengers suspected one another and the guard, they all suspected everybody else, and the coachman was sure of nothing but the horses; as to which cattle he could with a clear conscience have taken his oath on the two Testaments that they were not fit for the journey.
The Dover mail was in its usual genial position that the guard suspected the passengers, the passengers suspected one another and the guard, they all suspected everybody else, and the coachman was sure of nothing but the horses; as to which cattle he could with a clear conscience have taken his oath on the two Testaments that they were not fit for the journey.

- A
- ALL
- ANOTHER
- AS
- BUT
- CATTLE
- CLEAR
- COACHMAN
- CONSCIENCE
- COULD
- DOVER
- ELSE
- EVERYBODY
- FIT
- FOR
- GENIAL
- HAVE
- HE
- HIS
- HORSES
- IN
- ITS
- JOURNEY
- MAIL
- NOT
- NOTHING
- OATH
- OF
- ON
- ONE
- POSITION
- SURE
- TAKEN
- TESTAMENTS
- TO
- TWO
- USUAL
- WERE
- WHICH
- WITH
- AND
- GUARD
- PASSENGERS
- THAT
- THEY
- WAS
- SUSPECTED
- THE
The dover mail: token counts

- Number of word types with token count 1: 40
  - A 1
  - ALL 1
  - ANOTHER 1
  - AS 1
  - BUT 1
  - CATTLE 1
  - CLEAR 1
  - COACHMAN 1
  - CONSCIENCE 1
  - COULD 1
  - DOVER 1
  - ELSE 1
  - EVERYBODY 1
  - FIT 1
  - FOR 1
  - GENIAL 1
  - HAVE 1
  - HE 1
  - HIS 1
  - HORSES 1
  - IN 1

- Number of word types with token count 2: 6
  - AND 2
  - GUARD 2
  - PASSENGERS 2
  - THAT 2
  - THEY 2
  - WAS 2
  - SUSPECTED 3
  - THE 9

- Number of word types with token count 3: 1
- Number of word types with token count 4-8: 0
- Number of word types with token count 9: 1
- Number of word types with token count > 9: 0
Called a called a *frequency-of-frequency* chart
– Or a *count-of-count* chart
Regression to the mean

- Illustration: for a zero mean, unit variance RV, with zero-mean, 0.2 stdev noise
Solution to bias

• The population frequencies of species and the estimation of population parameters, I. J. Good, Biometrika 1953

• Assign probabilities based on category predictions for future data
  – Handwave: Compute this using a leave-one-out justification
A plot of the count of counts of words in a training corpus typically looks like this:

In keeping with Zipf’s law, the number of words that occur \( n \) times in the training corpus is typically more than the number of words that occur \( n+1 \) times.
Total probability mass

- **Black line**: Count of counts
  - Black line value at $K = \text{No. of word types that occur } K \text{ times}$

- **Red line**: Total probability mass of all events with that count
  - Red line value at 1 = $1. \frac{N_1}{N}$
  - Red line value at 2 = $2. \frac{N_2}{N}$
  - Red line value at $R = R. \frac{N_R}{N}$
Red Line

\[ P(C_R) = \frac{R N_R}{N} \]

- \( R \) = No. of times word type was seen
- \( N_R \) is no. of types seen \( R \) times
- \( N \): Total tokens
In keeping with Zipf’s law, the number of types that occur \( R \) times in the corpus is typically more than the number of types that occur \( R + 1 \) times.

- The total probability mass of words falls slowly with \( R \).
- The total probability mass of rare words is greater than the total probability mass of common words, because of the large number of rare words.
Good Turing smoothing reallocates probabilities

- The total probability mass of all words that occurred $R$ times is assigned to words that occurred $R-1$ times
- The total probability mass of words that occurred once is reallocated to words that were never observed in training
Good Turing Discounting

- Assign total probability mass of all words seen $R + 1$ times to all words seen $R$ times.
  \[ P_{GT}(w \in C_R) = \frac{(R + 1)N_{R+1}}{N} \]

- Share this probability among the $N_R$ words
  \[ P_{GT}(w) = \frac{N}{N_R} \times \frac{(R + 1)N_{R+1}}{N_R} \]
  \[ = \frac{(R + 1)N_{R+1}}{N} \]

- Equivalent computation: Discounted count for words in $C_R$ is:
  \[ R_{GT} = \frac{(R + 1)N_{R+1}}{N_R} \]
  - Modified probability: Use discounted count as the count for the word
    \[ P_{GT}(w) = \frac{R_{GT}}{N} \]
The probability mass curve cannot simply be shifted left directly due to two potential problems:

- Directly shifting the probability mass curve assigns 0 probability to the most frequently occurring words.

Good Turing Discounting

- The probability mass curve cannot simply be shifted left directly due to two potential problems.
- Directly shifting the probability mass curve assigns 0 probability to the most frequently occurring words.
The count of counts curve is often not continuous. We may have words that occurred $L$ times, and words that occurred $L+2$ times, but none that occurred $L+1$ times. Good Turing Discounting

No. of types

True count of counts curve

No words observed with this count. Total probability mass here is 0.

- The count of counts curve is often not continuous
  - We may have words that occurred $L$ times, and words that occurred $L+2$ times, but none that occurred $L+1$ times
Good Turing Discounting

- The count of counts curve is often not continuous
  - We may have words that occurred \( L \) times, and words that occurred \( L+2 \) times, but none that occurred \( L+1 \) times
  - By simply reassigning probability masses backward, words that occurred \( L \) times are assigned the total probability of words that occurred \( L+1 \) times = 0!

Shifed curve assigns zero probability to words that were actually seen a large number of times!
The count of counts curve is smoothed and extrapolated
- Smoothing fills in “holes” – intermediate counts for which the curve went to 0
- Smoothing may also vary the counts of events that were observed
- Extrapolation extends the curve to one step beyond the maximum count observed in the data

Smoothing and extrapolation can be done by linear interpolation and extrapolation, or by fitting polynomials or splines

Probability masses are computed from the smoothed count-of-counts and reassigned
Good Turing Discounting

- **Step 1**: Compute count-of-counts curve
  - Let $N_R$ be the number of words that occurred $R$ times

- **Step 2**: Smooth and extend count-of-count curve
  - Let $N'_R$ be the smoothed count of the number of words that occurred $R$ times.

- The total smoothed count of all words that occurred $R$ times is $RN'_R$.
  - We operate entirely with the smoothed counts from here on
• **Step 3:** Compute discounted counts of words from smoothed counts.

\[ R_{\text{smooth}}^{GT} = (R + 1)N_{R+1}' / N_R' \]

• **Step 4:** Compute modified total count from smoothed counts

\[ N_{\text{total}}' = \sum_w R_{\text{smooth}}^{GT}(w) \]

• **Step 5:** A word \( w \) with count \( k \) is assigned probability

\[ P(w) = R_k^{GT} / N_{\text{total}}' \]
• **Step 6:** Compute a probability for unseen terms!!!!

• A probability mass \( P(C_0) = \frac{N_1}{N} \) is left over
  - The left-over probability mass is reassigned to words that were not seen in the training corpus

\[
P(\text{any unseen word}) = \frac{P(C_0)}{N_0}
\]
Where does GT apply

• Anywhere where the above kind of count-of-counts curve occurs
  – Any situation where the size of the multinomial set is very large and
data sparsity is an issue
    • Regardless of the true distribution

• Turns out this situation occurs everywhere
  – Not just unigrams, but also bigrams, trigrams, Ngram..
  – Not just language
Expectation Maximization
A Thought Experiment

- A person shoots a loaded dice repeatedly
- You observe the series of outcomes
- **Problem:** Estimate $p_1, p_2, \ldots, p_6$ (probability of the faces)

6 3 1 5 4 1 2 4 …
Recap: MLE of 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
  \[ \log(P(n_1, n_2, n_3, n_4, n_5, n_6)) = \log(\text{Const}) + \sum_i n_i \log(p_i) \]

- Solving for the probabilities gives us
  \[ p_i = \frac{n_i}{\sum_j n_j} \]
Recap: MLE of Multinomial

- Multiplying the counts by a large factor \( N \) does not change the estimate.
- Counts: \( O = (Nn_1, Nn_2, \ldots, Nn_6) \)
- Maximizing \( P(O) \) gives us

\[
p_i = \frac{Nn_i}{N \sum_j n_j} \quad \Rightarrow \quad p_i = \frac{n_i}{\sum_j n_j}
\]
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
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

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

6 3 1 5 4 1 2 4 ...

4 4 1 6 3 2 1 2 ...

6 4 1 5 3 2 2 2 ...

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
Mixture Distributions

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

\[
P(X) = \sum_{Z} P(Z)P(X | Z)
\]

Mixture Gaussian

\[
P(X) = \sum_{Z} P(Z)N(X; \mu_{z}, \Theta_{z})
\]

Mixture Unigram probabilities

\[
P(W) = \sum_{i} \lambda_{i} P_i(W)
\]
Maximum Likelihood Estimation

• For our problem:

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

– \( Z \) = color of dice

\[ P(n_1, n_2, n_3, n_4, n_5, n_6) = \text{Const} \prod_X P(X)^{n_X} = \text{Const} \prod_X \left( \sum_Z P(Z)P(X | Z) \right)^{n_X} \]

• Maximum likelihood solution: Maximize

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

• No closed form solution (summation inside log)!
– In general ML estimates for mixtures do not have a closed form
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
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
Estimation with unseen information

- **Mixture 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 including the observation in the right bin

- **Problem: $Z$ is unseen**
  - How do we estimate probabilities
  - *Which bin do we drop the number into?*
Finding the right “unseen” bin

- Scale up the problem: Assume we obtained a very large number $N$ of 6s
- Assume we know all parameters of the distribution
  - We know mixture weights $P(Z)$ and individual multinomials $P(X|Z)$
- How many of the 6s came from the blue and how many from the red dice?
The Expected behavior

- The expected number of 6s from each of the two bins is:

\[ E[N_Z] = N P(Z|X) = N \frac{P(Z)P(X|Z)}{\sum_Z P(Z)P(X|Z)} \]

- \( X = 6 \) for our example
A Thought Experiment

- We know the color of *some* outcomes, but not other 2
- "Scale" up all counts by N
- Count *scaled* up counts
  - Use *expected* counts where the actual color is unknown

<table>
<thead>
<tr>
<th></th>
<th>Blue bin</th>
<th></th>
<th>Red bin</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>1N</td>
</tr>
<tr>
<td>1</td>
<td>.2N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>1N</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>.2N</td>
<td>1N</td>
<td>0</td>
</tr>
</tbody>
</table>
A Thought Experiment

- The scaling factor is irrelevant and unnecessary.

<table>
<thead>
<tr>
<th>Blue bin</th>
<th>Red bin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6</td>
<td>1 2 3 4 5 6</td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1 1</td>
</tr>
<tr>
<td>1</td>
<td>.2 1</td>
</tr>
<tr>
<td>5</td>
<td>1 1</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>.6 1</td>
</tr>
<tr>
<td>2</td>
<td>1 1</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>.2 1 0 1 1 1 .6 1 1 1 0 0 1.4</td>
</tr>
</tbody>
</table>
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

- **We do not know $Z$ – it is hidden from us!**

- **Solution:** FRAGMENT THE OBSERVATION into every $Z$
Interpretation: Fragmented observations

- EM is an iterative algorithm
  - At each time there is a current estimate of parameters
- Observations are fragmented into all possible values of unobserved attribute $Z$ using current parameter estimates
  - The “size” of the fragments of any observation attributed to any $Z$ is the expected fraction of instances of that value from each $Z$
    - For multinomial, equal to the a posteriori probabilities of the various values of $Z$. Computed using Bayes’ rule:
      \[
P(Z | X) = \frac{P(X | Z)P(Z)}{P(X)} = CP(X | Z)P(Z)
\]
- Model parameters can be updated with statistics obtained from fragmented observations
Expectation Maximization

• Hypothetical Dice Shooter Example:
  • We obtain an initial estimate for the probability distribution of the two sets of dice (somehow):

  ![Probability Distribution](image1)

  ![Probability Distribution](image2)

  - We obtain an initial estimate for the probability with which the caller calls out the two shooters (somehow)

  ![Probability Distribution](image3)
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:
  \[
P(\text{red} \mid X = 4) = CP(X = 4 \mid Z = \text{red})P(Z = \text{red}) = C \times 0.05 \times 0.5 = C 0.025 \]
  \[
P(\text{blue} \mid X = 4) = CP(X = 4 \mid Z = \text{blue})P(Z = \text{blue}) = C \times 0.1 \times 0.5 = C 0.05 \]

\[
P(\text{red} \mid X = 4) = \frac{C 0.025}{C 0.025 + C 0.05} \]

\[
P(\text{red} \mid X = 4) = 0.33 \quad P(\text{blue} \mid X = 4) = 0.67
\]
Expectation Maximization

4 (0.33)  4 (0.67)
Every observed roll of the dice contributes to both “Red” and “Blue”
Every observed roll of the dice contributes to both “Red” and “Blue”
Every observed roll of the dice contributes to both “Red” and “Blue”
• 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

| Called | P(red|X) | P(blue|X) | Instances | Total red count | Total blue count |
|--------|--------|----------|-----------|----------------|-----------------|
| 1      | .57    | .43      | 3         | 1.71           | 1.29            |
| 2      | .14    | .86      | 4         | 0.56           | 3.44            |
| 3      | .33    | .67      | 2         | 0.66           | 1.34            |
| 4      | .33    | .67      | 4         | 1.32           | 2.68            |
| 5      | .33    | .67      | 2         | 0.66           | 1.34            |
| 6      | .8     | .2       | 3         | 2.4            | 0.60            |
| Total  |        |          | 18        | 7.31           | 10.69           |
**Expectation Maximization**

| Called | P(red|X) | P(blue|X) | Instances | Total red count | Total blue count |
|--------|--------|----------|-----------|----------------|-----------------|
| 1      | .57    | .43      | 3         | 1.71           | 1.29            |
| 2      | .14    | .86      | 4         | 0.56           | 3.44            |
| 3      | .33    | .67      | 2         | 0.66           | 1.34            |
| 4      | .33    | .67      | 4         | 1.32           | 2.68            |
| 5      | .33    | .67      | 2         | 0.66           | 1.34            |
| 6      | .8     | .2       | 3         | 2.4            | 0.60            |
| Total  |        |          | 18        | 7.31           | 10.69           |

- **Updated probability of Red dice:**
  - \( P(1 \mid Red) = \frac{1.71}{7.31} = 0.234 \)
  - \( P(2 \mid Red) = \frac{0.56}{7.31} = 0.077 \)
  - \( P(3 \mid Red) = \frac{0.66}{7.31} = 0.090 \)
  - \( P(4 \mid Red) = \frac{1.32}{7.31} = 0.181 \)
  - \( P(5 \mid Red) = \frac{0.66}{7.31} = 0.090 \)
  - \( P(6 \mid Red) = \frac{2.40}{7.31} = 0.328 \)

- **Updated prior probability of red:**
  - \( \frac{7.31}{18} = 0.41 \)

- **Updated probability of Blue dice:**
  - \( P(1 \mid Blue) = \frac{1.29}{11.69} = 0.122 \)
  - \( P(2 \mid Blue) = \frac{0.56}{11.69} = 0.322 \)
  - \( P(3 \mid Blue) = \frac{0.66}{11.69} = 0.125 \)
  - \( P(4 \mid Blue) = \frac{1.32}{11.69} = 0.250 \)
  - \( P(5 \mid Blue) = \frac{0.66}{11.69} = 0.125 \)
  - \( P(6 \mid Blue) = \frac{2.40}{11.69} = 0.056 \)

- **Updated prior probability of blue:**
  - \( \frac{10.69}{18} = 0.59 \)
### Expectation Maximization

| Called | P(red|X) | P(blue|X) | Instances | Total red count | Total blue count |
|--------|--------|----------|-----------|----------------|-----------------|
| 1      | .57    | .43      | 3         | 1.71           | 1.29            |
| 2      | .14    | .86      | 4         | 0.56           | 3.44            |
| 3      | .33    | .67      | 2         | 0.66           | 1.34            |
| 4      | .33    | .67      | 4         | 1.32           | 2.68            |
| 5      | .33    | .67      | 2         | 0.66           | 1.34            |
| 6      | .8     | .2       | 3         | 2.4            | 0.60            |
| Total  |        |          | 18        | 7.31           | 10.69           |

- Updated probability of Red dice:
  - $P(1 \mid \text{Red}) = 1.71/7.31 = 0.234$
  - $P(2 \mid \text{Red}) = 0.56/7.31 = 0.077$
  - $P(3 \mid \text{Red}) = 0.66/7.31 = 0.090$
  - $P(4 \mid \text{Red}) = 1.32/7.31 = 0.181$
  - $P(5 \mid \text{Red}) = 0.66/7.31 = 0.090$
  - $P(6 \mid \text{Red}) = 2.40/7.31 = 0.328$

- Updated prior probability of red:
  - $7.31/18 = 0.41$

- Updated probability of Blue dice:
  - $P(1 \mid \text{Blue}) = 1.29/11.69 = 0.122$
  - $P(2 \mid \text{Blue}) = 0.56/11.69 = 0.322$
  - $P(3 \mid \text{Blue}) = 0.66/11.69 = 0.125$
  - $P(4 \mid \text{Blue}) = 1.32/11.69 = 0.250$
  - $P(5 \mid \text{Blue}) = 0.66/11.69 = 0.125$
  - $P(6 \mid \text{Blue}) = 2.40/11.69 = 0.056$

- Updated prior probability of blue:
  - $10.69/18 = 0.59$

The updated values can be used to repeat the process. Estimation is an iterative process.
The Dice Shooter Example

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

• Given a sequence of observations $O_1, O_2, ..$
  – $N_X$ is the number of observations of number $X$

• Initialize $P(Z)$, $P(X|Z)$ for dice $Z$ and outcomes $X$

• Iterate:
  – For each outcome $X$:
    
    $P(Z | X) = \frac{P(X | Z)P(Z)}{\sum_{Z'} P(Z')P(X | Z')}$

  – Update:

    $P(X | Z) = \frac{N_X P(Z | X)}{\sum_X N_X P(Z | X)}$

    $P(Z) = \frac{\sum_X N_X P(Z | X)}{\sum_{Z'} \sum_X N_X P(Z' | X)}$
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:

\[ P(6) = \alpha P(6 | \text{red}) + \beta P(6 | \text{blue}) = \alpha P_r + \beta P_b \]

  • 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 [FIX]

\[ P(6) = 1.0(\alpha P_r + \beta P_b) + 0.0 \text{anything} \]

  • 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

\[ P(X) = \sum_k P(k) N(X; \mu_k, \Theta_k) = \sum_k \frac{P(k)}{\sqrt{(2\pi)^d | \Theta_k |}} \exp \left( -0.5 (X - \mu_k)^T \Theta_k^{-1} (X - \mu_k) \right) \]

• 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

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

- 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)
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

\[
\mu_{\text{red}} = \frac{1}{N_{\text{red}}} \sum_{i \in \text{red}} X_i \\
\Theta_{\text{red}} = \frac{1}{N_{\text{red}}} \sum_{i \in \text{red}} (X_i - \mu_{\text{red}})(X_i - \mu_{\text{red}})^T \\
P(\text{red}) = \frac{N_{\text{red}}}{N}
\]
Gaussian Mixtures: Generating model

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

- 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

\[
P(k \mid X) = \frac{P(X \mid k)P(k)}{\sum_{k'} P(k')P(X \mid k')} = \frac{P(k)N(X; \mu_k, \Theta_k)}{\sum_{k'} P(k')N(X; \mu_{k'}, \Theta_{k'})}
\]
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
- Compute fragment sizes for each Gaussian, for each observation

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

| Number | $P(\text{red}|X)$ | $P(\text{blue}|X)$ |
|--------|------------------|------------------|
| 6.1    | .81              | .19              |
| 1.4    | .33              | .67              |
| 5.3    | .75              | .25              |
| 1.9    | .41              | .59              |
| 4.2    | .64              | .36              |
| 2.2    | .43              | .57              |
| 4.9    | .66              | .34              |
| 0.5    | .05              | .95              |
Each observation contributes only as much as its fragment size to each statistic.

Mean(red) = 
\[
\frac{(6.1 \times 0.81 + 1.4 \times 0.33 + 5.3 \times 0.75 + 1.9 \times 0.41 + 4.2 \times 0.64 + 2.2 \times 0.43 + 4.9 \times 0.66 + 0.5 \times 0.05)}{(0.81 + 0.33 + 0.75 + 0.41 + 0.64 + 0.43 + 0.66 + 0.05)}
\]
= 17.05 / 4.08 = 4.18

Var(red) = 
\[
\frac{(6.1-4.18)^2 \times 0.81 + (1.4-4.18)^2 \times 0.33 + (5.3-4.18)^2 \times 0.75 + (1.9-4.18)^2 \times 0.41 + (4.2-4.18)^2 \times 0.64 + (2.2-4.18)^2 \times 0.43 + (4.9-4.18)^2 \times 0.66 + (0.5-4.18)^2 \times 0.05)}{(0.81 + 0.33 + 0.75 + 0.41 + 0.64 + 0.43 + 0.66 + 0.05)}
\]

\[
P(red) = \frac{4.08}{8}
\]
EM for Gaussian Mixtures

1. Initialize $P(k)$, $\mu_k$ and $\Theta_k$ for all Gaussians

2. For each observation $X$ compute \textit{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

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

4. If not converged, return to 2
EM estimation of Gaussian Mixtures

• An Example

Histogram of 4000 instances of a randomly generated data

Individual parameters of a two-Gaussian mixture estimated by EM

Two-Gaussian mixture estimated by EM
Typical GMM caveats

• Generic GMMs do not have maximum likelihood estimates
  – A GMM where one of the Gaussians is centered exactly on a training point, and has 0 variance, has infinite likelihood
  – A theoretically perfect ML solution would diverge towards such bogus “solutions”
  – We will always be computed “impermissible” local maximum solutions

• Solution: Floor variance to a reasonable value
  – Ensure the likelihood does not blow up
  – A too-low floor can still result in bogus estimates
    • Multiple single-training-point Gaussians with floored variance
  – A too-high floor can result in bogus estimates
    • Poor fits to actual modes in data

• Even with constraints, GMMs are non-identifiable
  – Permuting the Gaussians will result in identical likelihoods
Expectation Maximization

• The EM algorithm is used whenever proper statistical analysis of a phenomenon requires the knowledge of a hidden or missing variable (or a set of hidden/missing variables)
  – The hidden variable is often called a “latent” variable

• Some examples:
  – Estimating mixtures of distributions
    • Only data are observed. The individual distributions and mixing proportions must both be learnt.
  – Estimating the distribution of data, when some attributes are missing
  – Estimating the dynamics of a system, based only on observations that may be a complex function of system state
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 $p(\text{heads})$ and $p(\text{number})$ for the dice from a collection of outputs

• Problem 2:
  – Caller rolls two dice
  – He calls out the sum
  – Determine $P(\text{dice})$ from a collection of outputs
The dice and the coin

- Unknown: Whether it was head or tails
The dice and the coin

• Unknown: Whether it was head or tails

\[
P(\text{heads} \mid N) = \frac{P(N)P(\text{heads})}{P(N)P(\text{heads}) + P(N-1)P(\text{tails})}
\]

\[
\text{count}(N) = \# N \cdot P(\text{heads} \mid N) + \# (N - 1) \cdot P(\text{tails} \mid N - 1)
\]
The two dice

- Unknown: How to partition the number
- $\text{Count}_{\text{blue}}(3) += P(3,1 \mid 4)$
- $\text{Count}_{\text{blue}}(2) += P(2,2 \mid 4)$
- $\text{Count}_{\text{blue}}(1) += P(1,3 \mid 4)$
The two dice

- Update rules

\[ P(N, K - N | K) = \frac{P_1(N)P_2(K - N)}{\sum_{J=1}^{6} P_1(J)P_2(K - J)} \]

\[ \text{count}_1(N) = \sum_{K=2}^{12} \#K \cdot P(N, K - N | K) \]
Fragmentation can be hierarchical

\[ P(X) = \sum_k P(k) \sum_Z P(Z \mid k) P(X \mid Z, k) \]

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