Announcements

- Mid-term Exam follow-up:
  - Friday

- Assignment 5.75
  - Posted: Later Today
  - Early: Wednesday
  - Due: Friday

Objectives

- Recall the Expectation Maximization (EM) algorithm
- Briefly review the mixture of multinomials (MM) model
- Think about how to initialize MM parameters to get EM started.
- Gain insight into unsupervised learning for a real application

The Algorithm

EM Example: Iteration 1

Conceptually, EM is an iterative algorithm. What does it look like?

EM Example: Iteration 2
Summarize the key elements of the algorithm. What is happening in each iteration?

High Level: Expectation-Maximization (EM)

0. Choose value of K, number of clusters
1. Define the feature set
2. Choose initial parameter values for the model

- Iterative procedure:
  3. Use model to estimate partial label counts for all data points (E-step)
  4. Use the new complete data to estimate better model parameters (M-step)
  5. Repeat steps 4 and 5 until convergence
Recall: the Likelihood Surface ...

... is non-Convex:

Our approach

- Choose initial parameters
- Go uphill
- Try again?

Results

- Both:
- Assignment of data points to clusters (the mode of the fractional labels)
- Estimates of the model parameters!

The Model

- Parameters for the distribution of the hidden class variable (mixture weights)
- Parameters for the distribution of the word types given the k-th hidden class variable

Mixture of Multinomials Model

\[ V, 1 \leq i \leq N : \text{Cluster id of document } i | c_i \sim \text{Categorical}(\lambda) \]

where \( \lambda \) is a stochastic vector: \( \lambda_i \in \mathbb{R}^K, \sum_{i=1}^{K} \lambda_i = 1, \lambda_i \geq 0 \)

\[ P(c_i) = \lambda_i \]

\[ P(x_{ij} | c_i) = (\beta_{ij})^{x_{ij}} \]

Document \( i \) is a type-count vector: \( x_{ij} | c_i \sim \text{Multinomial}(\beta_{ij}, M_i) \)

where \( M_i = (K, V), \forall k \sum_{j=1}^{V} \beta_{ij} = 1 \)

Introduced Plates
Made the Parameters Explicit

How to Initialize?

\[ \lambda? \]
\[ \beta? \]
\[ c_i \sim \text{Categorical}(\lambda) \]
\[ x_i | c_i \sim \text{Multinomial}(\beta) \]

Intuitions about \( \lambda \)

- Your ideas:
  - We could guess that clusters are even.
  - We could guess that clusters are uneven.
  - We may not know anything about the balance of clusters in the data. Anything is possible.
  - We could look at similar data (or a subset of the data) and get a sense for the evenness of the data (or lack thereof).

How to capture the intuitions?

- Could assign point estimates for \( \lambda \) based on intuitions
- Could draw a point estimate for \( \lambda \) from a suitable distribution

Probability Vector Factory

- A Dirichlet distribution is like a factory for random probability vectors like \( \lambda \).

Intuitions about \( \lambda \)

- We could guess that clusters are even:
  - \( \lambda = \left[ \frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2} \right] \)
- We could guess that clusters are uneven:
  - \( \lambda \sim \text{Dirichlet}(\alpha) \)
  - \( \alpha = [0.1, 0.1, 0.1, \ldots, 0.1] \)
- We may not know anything about the balance of clusters in the data. Anything is possible:
  - \( \lambda \sim \text{Dirichlet}(\alpha) \)
  - \( \alpha = [1.11, 1.11, \ldots, 1] \)
- We could look at similar data (or a subset of the data) and get a sense for the evenness of the data (or lack thereof)…
Intuitions about $\beta_k$
(for a specific cluster $k$)

- Your ideas:
  - We could guess that the vocabulary is used evenly in a given cluster.
  - Intuitively some word types are more likely to be used than others in any given cluster.
    - I.e., vocabulary is used unevenly.
    - The number of word types used in a cluster may be relatively small.
  - We may not know about the balance of word types (vocabulary items) in the data for a given cluster.
  - We could look at similar data (or a subset of the data) and get a sense for whether certain word types are more likely than others.

Probability Vector Factory

- For each row $\beta_k$, we can draw a new random probability vectors from a Dirichlet distribution:

How to Initialize?

- $\alpha \sim \text{Dirichlet}(\alpha)$
- $\forall k \beta_k \sim \text{Dirichlet}(\alpha)$
- $c_i \sim \text{Categorical}(\beta_k)$
- $x_{ij} | c_i \sim \text{Multinomial}(\beta_k)$

Possibilities

- 3 possibilities explored by Marina Meila and David Heckerman in a well-known paper.
  1. Random
  2. Marginal
  3. HAC
Initialization #1: Random

- Initialize the parameters of the model independently of the data
- $\alpha$:
  - Initialize to be the uniform distribution
- $\beta_k (1 \leq k \leq K)$:
  - Sample from an uninformative distribution, namely from a uniform Dirichlet distribution:
  - $\gamma_j \sim \text{Categorical}(\beta)$

Hierarchical Agglomerative Clustering (HAC)

- Most popular heuristic clustering methods
- Big idea: pick up similar documents and stick them together, repeat
- Point Example (single link):

- You get a cluster hierarchy for free

Initialization #2: Noisy-Marginal (a.k.a. “Marginal”)

- $\Delta$:
  - Same as in Initialization #1
  - Initialize in a data-dependent fashion:
  - Estimate the parameters $y$ of a single Dirichlet distribution as follows:
    - Count each feature (word type) $t_j (1 \leq j \leq V)$ over the whole data set
    - Compute the MAP estimate of the counts using a Dirichlet prior (e.g., $(2,2, \ldots, 2)$)
    - Normalize by the total count of observed features
    - Use the normalized counts as an estimate of $y$
  - Multiply $y$ by an “equivalent sample size” (inventors use 2) to scale the height of the density peak around the mean.

- $\beta_k (1 \leq k \leq K)$:
  - Sample $\beta_k$ from this Dirichlet($y$)
  - (Thiesson, Meek, Chickering, and Heckerman)

Initialization #3: HAC

- Data-dependent
  - Perform HAC on random sample of the data
    - Using all $a$ is often intractable due to quadratic run time
  - $\Delta$:
    - Extract a set of sufficient statistics: Counts of clusters
    - Compute the MAP estimate of the counts using add-one smoothing
    - i.e., a Dirichlet prior (e.g., $(2,2, \ldots, 2)$)
    - Use this count vector as an estimate of $\Delta$
    - Sample $\Delta$ from this Dirichlet($y$)
  - $\beta_k$:
    - Same as above: Counts of words with each doc. cluster id
    - Compute the MAP estimate of the counts using add-one smoothing
    - i.e., a Dirichlet prior (e.g., $(2,2, \ldots, 2)$)
    - Use this count vector as an estimate of $\beta_k$
    - Then for each component $k$, sample parameters $\beta_k$ from this Dirichlet($y$)
  - Also known as “Cluster Refinement”

Zooming out ...

... for a Bayesian perspective

Hierarchical Model

$\Delta \sim \text{Dirichlet}(\omega)$

$\forall k \beta_k \sim \text{Dirichlet}(\omega)$

$c_i \sim \text{Categorical}(\Delta)$

$\mathbf{z} \mid c_i \sim \text{Multinomial}(\beta_{c_i})$
Turtles

- Yertle the Turtle and Other Stories
- By Dr. Seuss

Fleas

Great fleas have little fleas upon their backs to bite 'em,
And little fleas have lesser fleas, and so ad infinitum,
And the great fleas themselves, in turn, have greater fleas to go on,
While these again have greater still, and greater still, and so on.

--- Augustus De Morgan

Why Bayesian?

- Bayesian methods are attractive because
  - They manage uncertainty about model parameters
  - They allow one to incorporate prior knowledge during inference
  - They facilitate unsupervised learning
- But:
  - We're not quite ready to be fully Bayesian
  - For the moment, we are going to use inference methods for clustering that still involve point estimates of model parameters.
  - We'll try multiple values of the model parameters.

Next

- EM on Mixture of Gaussians