CS 401R: Probabilistic Foundations of Machine Learning

Lecture #18: Text Clustering with Expectation Maximization

Announcements

- Reading Report #6
  - Due: Wednesday
- Assignment #5.5
  - Early: Tuesday
  - Due: Thursday
  - Finish before mid-term exam
- Mid-term Exam this week:
  - Topics: posted today after class
  - In-class Review: Wednesday
  - Testing Center: Thursday-Saturday

Objectives

- Understand the Expectation Maximization (EM) algorithm for one model: Mixture of Multinomials
- See unsupervised learning at work on a real application: text clustering
- Prepare to do EM on a Mixture of Gaussians

EM Example: Iteration 0

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

EM Example: Iteration 1

EM Example: Iteration 2
EM Example: Iteration 3

EM Example: Iteration 4

EM Example: Iteration 5

EM Example: Iteration 6

EM Example: Iteration 20

What Happened?
**High Level: Expectation-Maximization (EM)**

0. Choose value of $K$, number of clusters
1. Define $V$ (the vocabulary) to include all word types in all documents.
2. Reduce $V$ by feature selection
3. Choose some initial parameters for the model
4. Use model to estimate partial label counts for all docs (E-step)
5. Use the new complete data to learn better model (M-step)
6. Repeat steps 4 and 5 until convergence

- Iterative procedure:

**EM**

* $D$: Posterior distribution over clusters for each document computed using model $\theta$
* $C$: Total of the partial label counts (i.e., Expected counts) using all documents, computed directly from $D$
* $\theta$: Maximum likelihood parameter estimates, computed directly from $C$

**Likelihood Surface**

Stop at the top!

**E Step: Example**

- Once you have your partial counts, re-estimate parameters like you would with regular counts

- For every document, $x_i$:
  1. Compute posterior probability of each cluster id (answer the conditional queries!):
     \[ p(\theta = 1 | x_i) = 0.2 \]
     \[ p(\theta = 2 | x_i) = 0.8 \]
  2. Use as partial counts, total them up:
     \[ \sum_{c=1}^{C} c \cdot p(\theta = c | x_i) \]

- Estimate of $\theta$ is ML.

- Estimate of $\lambda$ is MAP.

**M Step: Example**

- Once you have your partial counts, re-estimate parameters like you would with regular counts

- For every document, $x_i$:
  1. Compute posterior probability of each cluster id (answer the conditional queries!):
     \[ p(\theta = 1 | x_i) = \frac{C(\theta = 1)}{N} \]
     \[ p(\theta = 2 | x_i) = \frac{C(\theta = 2)}{N} \]

- Estimate of $\lambda$ is ML.

- Estimate of $\theta$ is MAP.
Etc.

- In the next E step, the partial counts will be different, because the parameters have changed.
- E.g.,
  - \( p(c_j = 1|x_j) \propto p(c_j = 1) \cdot \prod_{t=1}^n p(t_j|c_j = 1) \)
  - And the likelihood will increase!

EM for Multinomial Mixture Model:

M-step

1. Re-estimate \( \hat{\lambda} \) and \( \hat{\beta} \) from the fractionally labeled data:
   \[
   \forall c \in C, \hat{\lambda}_c = \hat{P}(c) = \frac{C(c)}{N}, \quad \text{where } N = \text{number of documents}
   \]
   \[
   \forall c \in V, \hat{\beta}_{ij} = \hat{P}(t_j \mid c) = \frac{C(t_j,c)}{\sum_{c'} C(t_j,c')}
   \]

2. What about smoothing? Could use a MAP estimate:
   \[
   \forall c \in C, \hat{\beta}_{ij} = \hat{P}(t_j \mid c) = \frac{1 + C(t_j,c)}{n \cdot \sum_{c'} C(t_j,c')} = \frac{1 + C(t_j,c)}{\sum_{c'} C(t_j,c')}
   \]

Alternative: CEM = Classification EM

= Greedy EM = Hard EM

- The "E" step is different, but initialization and the M-step are the same.

  - For each document \( x_i \):
    1. Choose the best cluster for document \( x_i \): 
       \[
       c_i^* = \arg \max_{c} \hat{P}(c \mid x_i) = \arg \max_{c} \hat{P}(c) \prod_{j=1}^{n} P(x_{ij} \mid c)
       \]
       \[
       = \arg \max \lambda_c \prod_{j=1}^{n} \beta_{ij}^{x_{ij}}
       \]
    2. Total up your new counts:
       \[
       \forall c \in C, \hat{C}(c) = \sum_{x_i} \hat{P}(c \mid x_i)
       \]

Log Likelihood of the Data

- We want the probability of the unlabeled data \( D \) according to a model with parameters \( \theta \):
  \[
  P(D \mid \theta) = \prod_{x \in D} P(x \mid \theta) = \prod_{x \in D} \prod_{c \in C} \lambda_c \cdot \prod_{j=1}^n \beta_{ij}^{x_{ij}}
  \]
- Independence of data (see previous lecture)
  \[
  \log P(D \mid \theta) = \log \prod_{x \in D} \prod_{c \in C} \lambda_c \cdot \prod_{j=1}^n \beta_{ij}^{x_{ij}}
  \]
  \[
  = \sum_{x \in D} \sum_{c \in C} \log \lambda_c \cdot \sum_{j=1}^n \log \beta_{ij}^{x_{ij}}
  \]
- Logarithm
  \[
  \log \prod_{x \in D} \prod_{c \in C} \lambda_c \cdot \prod_{j=1}^n \beta_{ij}^{x_{ij}}
  \]
- Log of product
  \[
  = \sum_{x \in D} \sum_{c \in C} \log \lambda_c \cdot \sum_{j=1}^n \log \beta_{ij}^{x_{ij}}
  \]
- Log of sum
  \[
  \log \sum_{x \in D} \prod_{c \in C} \lambda_c \cdot \prod_{j=1}^n \beta_{ij}^{x_{ij}}
  \]
- Log of product
  \[
  \sum_{x \in D} \sum_{c \in C} \log \lambda_c \cdot \sum_{j=1}^n \log \beta_{ij}^{x_{ij}}
  \]
Model Parameterizations

Data Likelihood

EM Properties

- Each step of EM is guaranteed to increase data likelihood - a hill climbing procedure
- Not guaranteed to find global maximum of data likelihood
- Data likelihood typically has many local maxima for a general model class and rich feature set
- Many “patterns” in the data that we can fit our model to...

EM in General

- EM is a technique for learning anytime we have incomplete data \((x, y)\)
- **Induction Scenario** (ours): we actually want to know \(y\) (e.g., clustering)
- **Convenience Scenario**: we want the marginal, \(P(x)\), and including \(y\) just makes the model simpler (e.g., mixing weights)

General Approach

- Learn \(y\) and \(\theta\)
  - **E-step**: make a guess at posteriors \(P(y | x, \theta)\)
    - This means scoring all completions with the current parameters
    - Treat the completions as (fractional) complete data, and count.
  - **M-step**: re-estimate \(\theta\) to maximize log \(P(x, y | \theta)\)
    - Then compute ML (or smoothed) estimates of model parameters

Advertisement

- Learn more about Natural Language Processing, topic models, and text mining:

Register for CS 679 – Fall 2015

Next

- Mixture of Gaussians