Expectation Maximization cont’d

Recap of EM so far...
Part 1:
What if we knew the cluster parameters \( \{ \pi_k, \mu_k, \Sigma_k \} \)?

Responsibilities in equations

\[
 r_{ik} = \frac{\pi_k \mathcal{N}(x_i | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(x_i | \mu_j, \Sigma_j)}
\]

Responsibility cluster \( k \) takes for observation \( i \)

\( \text{"prior"} = \text{prob. of class} \)

\( \text{"likelihood"} = \text{like of data given class} \)

\( \text{Normalized over all possible cluster assignments} \)
Part 1 summary

Desired soft assignments (responsibilities) are easy to compute when cluster parameters \( \{ \pi_k, \mu_k, \Sigma_k \} \) are known.

But, we don’t know these!

Part 2b:
What can we do with just soft assignments \( r_{ij} \)?
Estimating cluster parameters from soft assignments

Instead of having a full observation \( x_i \) in cluster \( k \), just allocate a portion \( r_{ik} \)

\( x_i \) divided across all clusters, as determined by \( r_{ik} \)

Maximum likelihood estimation from soft assignments

Just like in boosting with weighted observations...

<table>
<thead>
<tr>
<th>( R )</th>
<th>( G )</th>
<th>( B )</th>
<th>( r_{11} )</th>
<th>( r_{12} )</th>
<th>( r_{13} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_i[1] \times x_i[2] \times x_i[3] )</td>
<td>0.30</td>
<td>0.18</td>
<td>0.52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x_i[1] \times x_i[2] \times x_i[3] )</td>
<td>0.01</td>
<td>0.26</td>
<td>0.73</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x_i[1] \times x_i[2] \times x_i[3] )</td>
<td>0.002</td>
<td>0.008</td>
<td>0.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x_i[1] \times x_i[2] \times x_i[3] )</td>
<td>0.75</td>
<td>0.10</td>
<td>0.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x_i[1] \times x_i[2] \times x_i[3] )</td>
<td>0.05</td>
<td>0.93</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x_i[1] \times x_i[2] \times x_i[3] )</td>
<td>0.13</td>
<td>0.86</td>
<td>0.01</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total weight in cluster: (effective # of obs) 1.242 2.8 2.42

52% chance this obs is in cluster 3
Maximum likelihood estimation from soft assignments

Cluster-specific location/shape MLE

Compute cluster parameter estimates with weights on each row operation
Estimate cluster proportions from relative weights

\[ \hat{\pi}_k = \frac{N_{\text{soft}}^k}{N} \]

Total weight in cluster \(k\) = effective # obs

\[ N_{\text{soft}}^k = \sum_{i=1}^{N} r_{ik} \]

Part 2b summary

Still straightforward to compute cluster parameter estimates from soft assignments
Expectation maximization (EM)

Expectation maximization (EM): An iterative algorithm

Motivates an iterative algorithm:

1. **E-step**: estimate cluster responsibilities given current parameter estimates

   \[
   \hat{r}_{ik} = \frac{\hat{\pi}_k N(x_i \mid \hat{\mu}_k, \hat{\Sigma}_k)}{\sum_{j=1}^{K} \hat{\pi}_j N(x_i \mid \hat{\mu}_j, \hat{\Sigma}_j)}
   \]

2. **M-step**: maximize likelihood over parameters given current responsibilities

   \[
   \hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k \mid \{\hat{r}_{ik}, x_i\}
   \]
EM for mixtures of Gaussians in pictures – initialization

EM for mixtures of Gaussians in pictures – after 1\textsuperscript{st} iteration
EM for mixtures of Gaussians in pictures – after 2\textsuperscript{nd} iteration

EM for mixtures of Gaussians in pictures – converged solution
EM for mixtures of Gaussians in pictures – replay

The nitty gritty of EM
Convergence and initialization of EM

Convergence of EM

• EM is a coordinate-ascent algorithm
  – Can equate E- and M-steps with alternating maximizations of an objective function
• Convergences to a local mode

• We will assess via (log) likelihood of data under current parameter and responsibility estimates
Initialization

- Many ways to initialize the EM algorithm
- Important for convergence rates & quality of local mode found
- Examples:
  - Choose K observations at random to define K "centroids". Assign other observations to nearest centroid to form initial parameter estimates.
  - Pick centers sequentially to provide good coverage of data like in k-means++
  - Initialize from k-means solution
  - Grow mixture model by splitting (and sometimes removing) clusters until K clusters are formed

Potential of vanilla EM to overfit
Overfitting of MLE

Maximizing likelihood can **overfit to data**

Imagine at $K=2$ example with one obs assigned to **cluster 1** and others assigned to **cluster 2**
- What parameter values maximize likelihood?

![Set center equal to point and shrink variance to 0](image)

Likelihood goes to $\infty$!

Overfitting in high dims

**Doc-clustering example:**
Imagine only 1 doc assigned to cluster $k$ has word $w$ (or all docs in cluster agree on count of word $w$)

Likelihood maximized by setting $\mu_k[w] = x_i[w]$ and $\sigma_{w,k}^2 = 0$

Likelihood of any doc with different count on word $w$ being in cluster $k$ is 0!
Simple regularization of M-step for mixtures of Gaussians

Simple fix: Don’t let variances → 0!

Add small amount to diagonal of covariance estimate

Alternatively, take Bayesian approach & place prior on parameters.

Similar idea, but all parameter estimates are “smoothed” via cluster pseudo-observations.

Formally relating k-means and EM for mixtures of Gaussians
Relationship to k-means

Consider Gaussian mixture model with

\[ \Sigma = \begin{pmatrix} \sigma^2 & & \\ & \sigma^2 & \\ & & \sigma^2 \end{pmatrix} \]

Spherically symmetric clusters

and let the variance parameter \( \sigma \to 0 \)

- Spherical clusters w/ equal variances, so relative likelihoods just fcn of dist to cluster center
- As variances \( \to 0 \), likelihood ratio becomes 0 or 1
- Responsibilities weigh in cluster proportions, but dominated by likelihood disparity

\[ \hat{r}_{ik} = \frac{\hat{\pi}_k N(x_i \mid \hat{\mu}_k, \sigma^2 I)}{\sum_{j=1}^{K} \hat{\pi}_j N(x_i \mid \hat{\mu}_j, \sigma^2 I)} \in \{0, 1\} \]

Datapoint gets fully assigned to nearest center, just as in k-means

Infinitesimally small variance EM = k-means

1. **E-step:** estimate cluster responsibilities given current parameter estimates

\[ \hat{r}_{ik} = \frac{\hat{\pi}_k N(x_i \mid \hat{\mu}_k, \sigma^2 I)}{\sum_{j=1}^{K} \hat{\pi}_j N(x_i \mid \hat{\mu}_j, \sigma^2 I)} \in \{0, 1\} \]

Infinitesimally small

Decision based on distance to nearest cluster center

2. **M-step:** maximize likelihood over parameters given current responsibilities (hard assignments!)

\[ \hat{\pi}_k, \hat{\mu}_k \mid \{\hat{r}_{ik}, x_i\} \]
Summary for the EM algorithm

What you can do now...

• Estimate soft assignments (responsibilities) given mixture model parameters
• Solve maximum likelihood parameter estimation using soft assignments (weighted data)
• Implement an EM algorithm for inferring soft assignments and cluster parameters
  – Determine an initialization strategy
  – Implement a variant that helps avoid overfitting issues
• Compare and contrast with k-means
  – Soft vs. hard assignments
  – k-means as a limiting special case of EM for mixtures of Gaussians
Bayes Optimal Classifier & Naïve Bayes

Classification

Learn: $f: X \mapsto Y$
- $X$ – features
- $Y$ – target classes

Suppose you know $P(Y|X)$ exactly, how should you classify?
- Bayes optimal classifier:
Recall: Bayes rule

\[ P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)} \]

Which is shorthand for:

\[ (\forall i, j) \quad P(Y = i \mid X = j) = \frac{P(X = j \mid Y = i)P(Y = i)}{P(X = j)} \]

How hard is it to learn the optimal classifier?

- Data =

<table>
<thead>
<tr>
<th>Sky</th>
<th>Temp</th>
<th>Humid</th>
<th>Wind</th>
<th>Water</th>
<th>Forecast</th>
<th>EnjoySpt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Warm</td>
<td>Normal</td>
<td>Strong</td>
<td>Warm</td>
<td>Same</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Warm</td>
<td>High</td>
<td>Strong</td>
<td>Warm</td>
<td>Same</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cold</td>
<td>High</td>
<td>Strong</td>
<td>Warm</td>
<td>Change</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Warm</td>
<td>High</td>
<td>Strong</td>
<td>Cold</td>
<td>Change</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- How do we represent these? How many parameters?
  - Prior, P(Y):
    - Suppose Y is composed of k classes

- Likelihood, P(X|Y):
  - Suppose X is composed of d binary features

- Complex model! High variance with limited data!!!
Conditional Independence

X is conditionally independent of Y given Z, if the probability distribution governing X is independent of the value of Y, given the value of Z.

\[(\forall i, j, k) \quad P(X = i \mid Y = j, Z = k) = P(X = i \mid Z = k)\]

e.g.,

\[P(\text{Thunder} \mid \text{Rain, Lightening}) = P(\text{Thunder} \mid \text{Lightening})\]

Equivalent to:

\[P(X, Y \mid Z) = P(X \mid Z)P(Y \mid Z)\]

What if features are independent?

• Predict Lightening
• From two conditionally independent features
  – Thunder
  – Rain
The Naïve Bayes assumption

• Naïve Bayes assumption:
  – Features are independent given class:
    \[
    \]
  – More generally:
    \[
    P(X[1], \ldots, X[d] | Y) = \prod_{j} P(X[j] | Y)
    \]
• How many parameters now?
  • Suppose X is composed of d binary features

The Naïve Bayes classifier

• Given:
  – Prior P(Y)
  – d conditionally independent features X[j] given the class Y
  – For each X[j], we have likelihood P(X[j]|Y)

• Decision rule:
  \[
  \hat{y} = f_{NB}(x) = \arg \max_{y} P(y)P(x[1], \ldots, x[d] | y)
  = \arg \max_{y} P(y) \prod_{j} P(x[j] | y)
  \]
• If assumption holds, NB is optimal classifier!
MLE for the parameters of NB

• Given dataset
  - Count(A=a,B=b) == # examples where A=a and B=b

• MLE for NB, simply:
  - Prior: P(Y=y) =


Subtleties of NB classifier 1 – Violating the NB assumption

• Usually, features are not conditionally independent:
  \[ P(X[1], \ldots, X[d] | Y) \neq \prod_j P(X[j] | Y) \]

• Actual probabilities P(Y|X) often biased towards 0 or 1

• Nonetheless, NB is one of the most used classifier out there
  - NB often performs well, even when assumption is violated
  - [Domingos & Pazzani ’96] discuss some conditions for good performance
Subtleties of NB classifier 2 – Insufficient training data

- What if you never see a training instance where $X[1]=a$ when $Y=b$?
  - e.g., $Y=\text{SpamEmail}$, $X[1]=\text{Viagra}$
  - $P(X[1]=a \mid Y=b) = 0$

- Thus, no matter what the values $X[2], \ldots, X[d]$ take:
  - $P(Y=b \mid X[1]=a, X[2], \ldots, X[d]) = 0$

- “Solution”: smoothing
  - Add “fake” counts, usually uniformly distributed
  - Equivalent to Bayesian learning

Text classification

- Classify e-mails
  - $Y = \{\text{Spam,NotSpam}\}$
- Classify news articles
  - $Y = \{\text{what is the topic of the article?}\}$
- Classify webpages
  - $Y = \{\text{Student, professor, project, ...}\}$

- What about the features $X$?
  - The text!
Features $X$ are entire document – $X[j]$ for jth word in article

**Article from rec.sport.hockey**

Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.e
From: xxx@yyy.zzz.edu (John Doe)
Subject: Re: This year's biggest and worst (opinio
Date: 5 Apr 93 09:53:39 GMT

I can only comment on the Kings, but the most
obvious candidate for pleasant surprise is Alex
Zhritnik. He came highly touted as a defensive
defenseman, but he's clearly much more than that.
Great skater and hard shot (though wish he were
more accurate). In fact, he pretty much allowed
the Kings to trade away that huge defensive
liability Paul Coffey. Kelly Hruday is only the
biggest disappointment if you thought he was any
good to begin with. But, at best, he's only a
mediocre goaltender. A better choice would be
Tomas Sandstrom, though not through any fault of
his own, but because some thugs in Toronto decided

### NB for text classification

- $P(X|Y)$ is huge!!!
  - Article at least 1000 words, $X=\{X[1],\ldots, X[1000]\}$
  - $X[j]$ represents jth word in document
    - i.e., the domain of $X[j]$ is entire vocabulary, e.g., Webster Dictionary
    - (or more), 10,000 words, etc.

- NB assumption helps a lot!!!
  - $P(X[j]=x[j]|Y=y)$ is the probability of observing word $x[j]$ in a
document on topic $y$

$$f_{NB}(x) = \arg \max_y \prod_{j=1}^{\text{LengthDoc}} P(x[j] \mid y) P(y)$$
Bag of words model

• Typical additional assumption: **Position in document doesn’t matter**
  
  \[ P(\mathbf{X}[j]=\mathbf{x}[j] | Y=y) = P(\mathbf{X}[k]=\mathbf{x}[j] | Y=y) \]
  
  - “Bag of words” model – order of words on the page ignored
  - Sounds really silly, but often works very well!

\[
P(y) \prod_{j=1}^{\text{LengthDoc}} P(\mathbf{x}[j] | y)
\]

When the lecture is over, remember to wake up the person sitting next to you in the lecture room.
Abstract

Patients with epilepsy can manifest short, sub-clinical epileptic “bursts” in addition to full-blown clinical seizures. We believe the relationship between these two classes of events—something not previously studied quantitatively—could hold important insights into the nature and intrinsic dynamics of seizures. A goal of our work is to parse these complex epileptic events into distinct dynamic regimes. A challenge posed by the intracranial EEG (iEEG) data we study is the fact that the number and placement of electrodes can vary between patients. We develop a Bayesian nonparametric Markov switching process that allows for (i) shared dynamic regimes between a variable number of channels, (ii) asynchronous regime-switching, and (iii) an unknown dictionary of dynamic regimes. We encode a sparse and changing set of dependencies between the channels using a Markov-switching Gaussian graphical model for the innovations process driving the channel dynamics and demonstrate the importance of this model in parsing and out-of-sample predictions of iEEG data. We show that our model produces interpretable state assignments that can help automate clinical analysis of seizures and enable the comparison of sub-clinical bursts and full clinical seizures.

Keywords: Bayesian nonparametric, EEG, factorial hidden Markov model, graphical model, time series.

1. Introduction

Despite over three decades of research, we still have very little idea of what defines a seizure. This ignorance stems from the complexity of epilepsy as a disease and a paucity of quantitative tools that are flexible...
NB with bag of words for text classification

• Learning phase:
  - Prior $P(Y)$
    • Count how many documents you have from each topic (+ prior)
  - $P(X[j] | Y)$
    • For each topic, count how many times you saw word in documents of this topic (+ prior)

• Test phase:
  - For each document
    • Use naïve Bayes decision rule
      \[
      f_{NB}(x) = \arg \max_y P(y) \prod_{j=1}^{LengthDoc} P(x[j] | y)
      \]

Twenty News Groups results

Given 1000 training documents from each group
Learn to classify new documents according to which newsgroup it came from

- comp.graphics
- comp.os.ms-windows.misc
- comp.sys.ibm.pc.hardware
- comp.sys.mac.hardware
- comp.windows.x
- misc.forsale
- rec.autos
- rec.motorcycles
- rec.sport.baseball
- rec.sport.hockey
- alt.atheism
- sci.space
- sci.crypt
- sci.electronics
- sci.med
- soc.religion.christian
- talk.religion.misc
- talk.politics.mideast
- talk.politics.misc
- talk.politics.guns

Naïve Bayes: 89% classification accuracy
Learning curve for Twenty News Groups

Accuracy vs. Training set size (1/3 withheld for test)