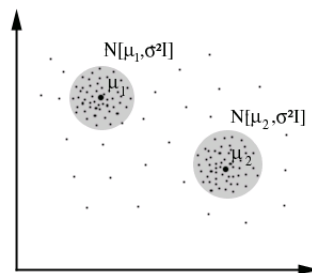


Discriminative vs. Generative Clustering

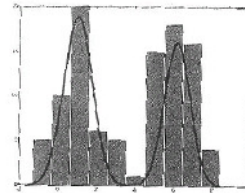
- Discriminative:
 - Cluster data points by similarities
 - Example: k -means and hierarchical clustering
- Generative
 - Generate models to describe different clusters, and use these models to classify points
 - Example: Clustering via finite Gaussian mixture models

Gaussian Mixture Models (GMMs)



To generate data:

- Choose one of the Gaussians with probability α_i
- Sample a point from that Gaussian



From: http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/mixture.html

Gaussian Mixture Models (GMMs): A “soft” version of K -means

K -means:

Assumption: Data is clustered about K *centroids*, which are the means of the data in the cluster

Goal: Find the centroids that best cluster the data.

Method: Guess the centroids. Then, alternate:

1. Assign each data point to a centroid
2. Update the centroid to be the mean of the new cluster

Stopping criteria:

- Data points don't change clusters
- Centroids don't change location

GMMs:

Assumption: Each data point is generated by sampling from one of K Gaussians, with some probability distribution over the Gaussians.

Goal: Find parameters (μ , σ^2) of these Gaussians and probability distribution over these Gaussians that had the highest probability (“maximum likelihood”) of generating the given data.

Method: Expectation-Maximization (EM) algorithm: Guess the parameters and probability distribution. Then, alternate:

E step: For each data point, calculate the “responsibility” of each Gaussian, using the current parameter values.

M step: Using the calculated “responsibilities”, reestimate the current parameters.

Stopping criteria: Analogous to K -means

Gaussian Mixture Models

- Let X be a set of multivariate data points (vectors):

$$X = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}.$$

- General expression for Gaussian mixture model (with K Gaussians):

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\text{where } \sum_{k=1}^K \pi_k = 1$$

and $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is a Gaussian with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$.

Text classification from labeled and unlabeled documents using EM

K. Nigam et al., *Machine Learning*, 2000

- Big problem with text classification: need labeled data.
- What we have: lots of unlabeled data.
- Question of this paper: Can unlabeled data be used to increase classification accuracy?
- I.e.: Any information implicit in unlabeled data? Any way to take advantage of this implicit information?

General idea: A version of EM algorithm

- Train a classifier with small set of available labeled documents.
- Use this classifier to assign probabilistically-weighted class labels to unlabeled documents.
- Then train a new classifier using all the documents, both originally labeled and formerly unlabeled.
- Iterate.

Probabilistic framework

- Assumes data are generated with Gaussian mixture model
- Assumes one-to-one correspondence between mixture components and classes.
- “These assumptions rarely hold in real-world text data”

Probabilistic framework

Let $C = \{c_1, \dots, c_K\}$ be the classes / mixture components

Let $\theta = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\} \cup \{\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_K\} \cup \{\pi_1, \dots, \pi_K\}$ be the mixture parameters.

Assumptions: A document d_i is created by first selecting a mixture component according to the mixture weights π_j , then having this selected mixture component generate a document according to its own parameters, with distribution

$$p(d_i | c_j; \theta).$$

- Likelihood of document d_i :

$$p(d_i | \theta) = \sum_{j=1}^k \pi_k p(d_i | c_j; \theta)$$

- Now, we will apply EM to a naive Bayes Classifier

Recall naive Bayes classifier: Assume each attribute is conditionally independent, given c_j .

Let $\mathbf{x} = (a_1, a_2, \dots, a_D)$

We have:

$$p(a_1, a_2, \dots, a_D | c_j) = p(a_1 | c_j) p(a_2 | c_j) \cdots p(a_D | c_j)$$

$$p(c_j | \mathbf{x}) = p(c_j) \prod_i p(a_i | c_j), i = 1, \dots, D; j = 1, \dots, K$$

To “train” naive Bayes from labeled data, estimate

$$p(c_j) \text{ and } p(a_i | c_j), j = 1, \dots, K; i = 1, \dots, D$$

These values are estimates of the parameters in θ . Call these values $\hat{\theta}$.

Note that Naive Bayes can be thought of as a generative mixture model.

Document d_i is represented as a vector of word frequencies $(w_1, \dots, w_{|V|})$, where V is the vocabulary (all known words).

There is an assumed probability distribution over words associated with each class, parameterized by θ .

We need to find estimate $\hat{\theta}$ to determine what probability distribution document $d_i = (w_1, \dots, w_{|V|})$ is most likely to come from.

Applying EM to Naive Bayes

- We have a small number of labeled documents S_{labeled} and a large number of unlabeled documents, $S_{\text{unlabeled}}$.
- The initial parameters $\hat{\theta}$ are estimated from the labeled documents S_{labeled} .
- **Expectation step:** The resulting classifier is used to assign probabilistically-weighted class labels $p(c_j | \mathbf{x})$ to each unlabeled document $\mathbf{x} \in S_{\text{unlabeled}}$.
- **Maximization step:** Re-estimate $\hat{\theta}$ using $p(c_j | \mathbf{x})$ values for $\mathbf{x} \in S_{\text{unlabeled}} \cup S_{\text{labeled}}$.
- Repeat until $p(c_j | \mathbf{x})$ or $\hat{\theta}$ has converged.

Data

- 20 UseNet newsgroups
- Web pages (WebKB)
- Newswire articles (Reuters)

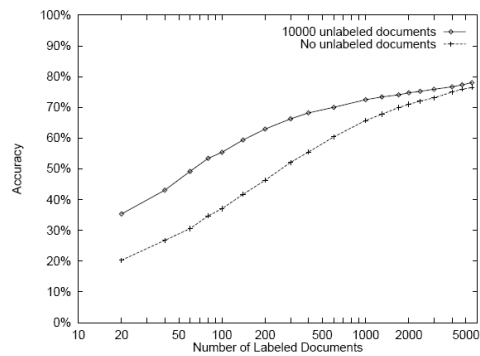


Figure 2. Classification accuracy on the 20 Newsgroups data set, both with and without 10,000 unlabeled documents. With small amounts of training data, using EM yields more accurate classifiers. With large amounts of labeled training data, accurate parameter estimates can be obtained without the use of unlabeled data, and the two methods begin to converge.

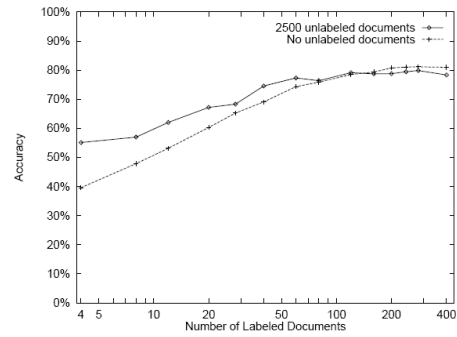


Figure 4. Classification accuracy on the WebKB data set, both with and without 2500 unlabeled documents. When there are small numbers of labeled documents, EM improves accuracy. When there are many labeled documents, however, EM degrades performance slightly—indicating a misfit between the data and the assumed generative model.