STEVENS INSTITUTE of TECHNOLOGY
School of Business

FE 582

Lecture 8: Mining Text Data

Dragos Bozdog

Specific characteristics unique to text data

Number of “zero” attributes

- If each word in the lexicon is viewed as an attribute, and the document word frequency is viewed as the attribute value, most attribute values are 0.
- This phenomenon is referred to as high-dimensional *sparsity*.

Nonnegativity

- The frequencies of words take on nonnegative values.

Side information

- In some domains, such as the Web, additional side information is available.
- These additional attributes can be leveraged to enhance the mining process further.

Document Preparation and Similarity Computation

Convert raw text documents to the multidimensional format.

Stop word removal

- Stop words are frequently occurring words in a language that are not very discriminative for mining applications.

Stemming

- Variations of the same word need to be consolidated.

Punctuation marks

- After stemming has been performed, punctuation marks, such as commas and semicolons, are removed.
- Numeric digits are removed.
- Hyphens are removed, if the removal results in distinct and meaningful words.

Document Preparation and Similarity Computation

The problem of document normalization is closely related to that of similarity computation.

Two primary types of normalization are applied to documents:

Inverse document frequency

- Higher frequency words tend to contribute noise to data mining operations such as similarity computation.

Frequency damping

- To provide greater stability to the similarity computation, a damping function is applied to word frequencies so that the frequencies of different words become more similar to one another.

Document Preparation and Similarity Computation

Types of normalization

- The inverse document frequency id_i of the i^{th} term is a decreasing function of the number of documents n_i in which it occurs:

$$id_i = \log\left(\frac{n}{n_i}\right)$$

with n number of documents in the collection

Frequency damping normalization ensures that the excessive presence of a single word does not throw off the similarity computation.

- Consider a document with word-frequency vector $\bar{X} = (x_1 \cdots x_d)$, where *d* is the size of the lexicon.

- A damping function $f(.)$, such as the square root or the logarithm, is optionally applied to the frequencies before similarity computation:

$$f(x_i) = \sqrt{x_i} \quad \text{or} \quad f(x_i) = \log(x_i)$$

Document Preparation and Similarity Computation

- The *normalized* frequency $h(x_i)$ for the i^{th} word may be defined as follows:

$$h(x_i) = f(x_i) \cdot id_i$$

- We already discussed cosine and Jaccard coefficient as popular measures.

Specialized Clustering Methods for Text

We will first focus on generic multidimensional clustering algorithms, and then present specific algorithms in these contexts.

Representative-Based Algorithms

- These correspond to the family of algorithms such as *k*-means, *k*-modes, and *k*-median algorithms. Among these, the *k*-means algorithms are the most popularly used for text data.

- Two major modifications are required for effectively applying these algorithms to text data:

-The choice of the similarity function. Instead of the Euclidean distance, the cosine similarity function is used.

-Modifications are made to the computation of the cluster centroid. All words in the centroid are not retained. The low-frequency words in the cluster are projected out.

Specialized Clustering Methods for Text

k-means clustering

- *Kernel k-means:* The key idea is that the Euclidean distance between a data point *X* and the cluster centroid *μ* of cluster *C* can be computed as a function of the dot product between *X* and the data points in *C*:

$$\boxed{||\overline{X}-\overline{\mu}||^2 = ||\overline{X}-\frac{\sum_{\overline{X_i}\in\mathcal{C}}\overline{X_i}}{|\mathcal{C}|}||^2 = \overline{X}\cdot\overline{X}-2\frac{\sum_{\overline{X_i}\in\mathcal{C}}\overline{X}\cdot\overline{X_i}}{|\mathcal{C}|}+\frac{\sum_{\overline{X_i},\overline{X_j}\in\mathcal{C}}\overline{X_i}\cdot\overline{X_j}}{|\mathcal{C}|^2}}$$

- In kernel *k*-means, the dot products $\bar{X}_i \cdot \bar{X}_j$ are replaced with kernel similarity values $K(\bar{X}_i, \bar{X}_j)$.

Probabilistic Algorithms

Probabilistic text clustering can be considered an unsupervised version of the naive Bayes classification method.

- It is assumed that the documents need to be assigned to one of *k* clusters $G_1 \cdots G_n$. The basic idea is to use the following generative process:

1) Select a cluster G_m, where $m \in \{1 \dots k\}$.

2) Generate the term distribution of G_m based on a generative model.

- The observed data are then used to estimate the parameters of the Bernoulli or multinomial distributions in the generative process.

Probabilistic Algorithms

The clustering is done in an iterative way with the EM algorithm, where cluster assignments of documents are determined from conditional word distributions in the E-step with the Bayes rule, and the conditional word distributions are inferred from cluster assignments in the M-step.

For initialization:

- The documents are randomly assigned to clusters.
- Initial prior probabilities $\mathrm{P}(G_m)$ and conditional feature distributions $\mathrm{P}\left(w_j \mid G_m\right)$ are estimated from the statistical distribution of this random assignment.
- A Bayes classifier is used to estimate the posterior probability $\mathrm{P}(G_m \mid X)$ in the E-step.

Probabilistic Algorithms

- The conditional feature distribution $\mathrm{P}(w_j \mid G_m)$ for word w_j is estimated from these posterior probabilities in the M-step as follows:

$$\boxed{P(w_j|\mathcal{G}_m) = \frac{\sum_{\overline{X}} P(\mathcal{G}_m|\overline{X}) \cdot I(\overline{X}, w_j)}{\sum_{\overline{X}} P(\mathcal{G}_m|\overline{X})}}$$

where $I(\bar{X}, w_j)$ an indicator variable that takes on the value of 1, if the word w_j present in *X*, and 0, otherwise.

- The next E-step uses these modified values of $\mathrm{P}(w_j \mid G_m)$ and the prior probability to derive the posterior Bayes probability with a standard Bayes classifier. Therefore, the following two iterative steps are repeated to convergence:

Probabilistic Algorithms

1) (E-step) Estimate posterior probability of membership of documents to clusters using Bayes rule:

$$\boxed{P(\mathcal{G}_m|\overline{X}) \propto P(\mathcal{G}_m) \prod_{w_j \in \overline{X}} P(w_j|\mathcal{G}_m) \prod_{w_j \notin \overline{X}} (1 - P(w_j|\mathcal{G}_m)) \,.}$$

2) (M-step) Estimate conditional distribution $\mathrm{P}(w_j \mid G_m)$ of words and prior probabilities $\mathrm{P}(G_m)$ of different clusters using the estimated probabilities in the E-step.

- At the end of the process, the estimated value of $\mathrm{P}(G_m \mid \bar{X})$ provides a cluster assignment probability and the estimated value of $\mathrm{P}(w_j \mid G_m)$ provides the term distribution of each cluster.

Co-clustering

Simultaneous Document and Word Cluster Discovery

- Co-clustering is most effective for nonnegative matrices in which many entries have zero values.

- Co-clustering is also referred to as *bi-clustering* or *two-mode clustering* because of its exploitation of both “modes” (words and documents).

Co-clustering

- In the context of text data, this matrix is the $n \times d$ document term matrix *D*, where rows correspond to documents and columns correspond to words.

- Thus, the i^{th} cluster is associated with a set of rows R_i (documents), and a set of columns V_i (words).

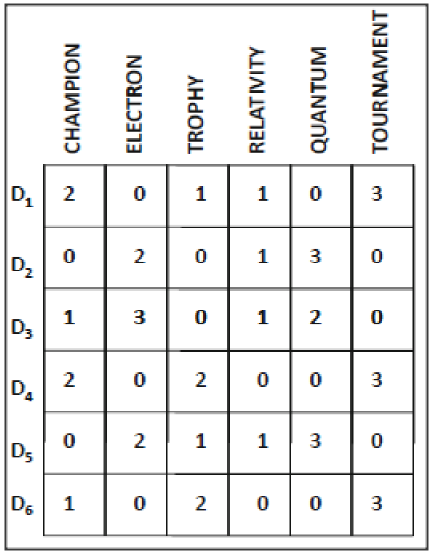
- The rows R_i are disjoint from one another over different values of *i*, and the columns V_i are also disjoint from one another over different values of *i*.

- The co-clustering method simultaneously leads to document clusters and word clusters.

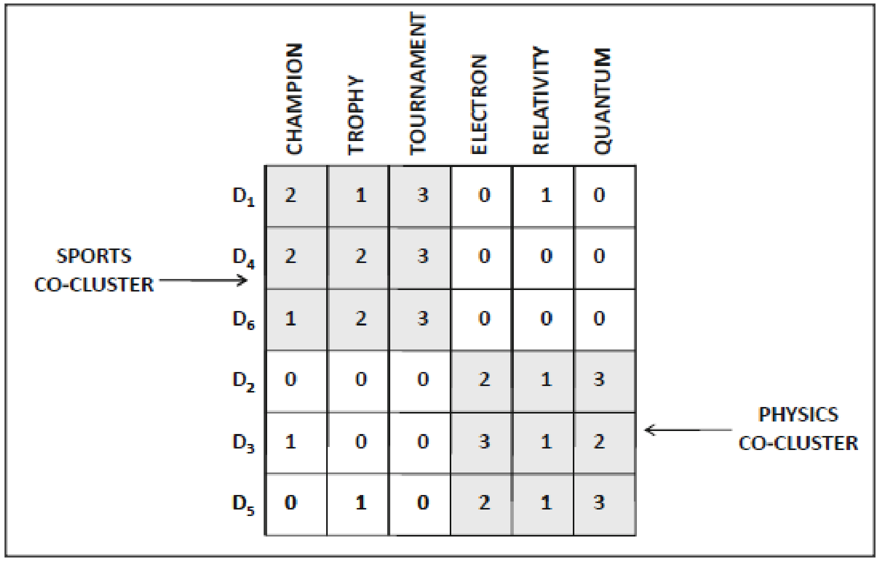
Co-clustering

Ex: 6 × 6 document-word matrix.

- The entries in the matrix correspond to the word frequencies in six documents denoted by $D_1 \cdots D_6$.
- The six words in this case are *champion, electron, trophy, relativity, quantum,* and *tournament*.
- If the rows and columns were permuted, so that all the sports-related rows/columns occur earlier than all the science-related rows/columns, then the resulting matrix is shown on the right



	CHAMPION	ELECTRON	TROPHY	RELATIVITY	QUANTUM	TOURNAMENT
D_1	2	0	1	1	0	3
D_2	0	2	0	1	3	0
D_3	1	3	0	1	2	0
D_4	2	0	2	0	0	3
D_5	0	2	1	1	3	0
D_6	1	0	2	0	0	3


SPORTS CO-CLUSTER
PHYSICS CO-CLUSTER

	CHAMPION	TROPHY	TOURNAMENT	ELECTRON	RELATIVITY	QUANTUM
D_1	2	1	3	0	1	0
D_4	2	2	3	0	0	0
D_6	1	2	3	0	0	0
D_2	0	0	0	2	1	3
D_3	1	0	0	3	1	2
D_5	0	1	0	2	1	3

Co-clustering

The goal is to minimize the weights of the nonzero entries outside these shaded blocks.

Solution?

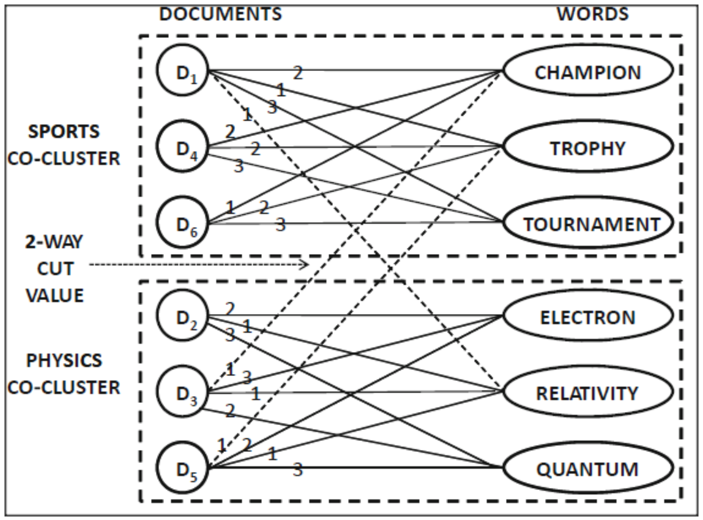
- Convert the problem to a graph partitioning problem, so that the aggregate weight of the nonzero entries in the nonshaded regions is equal to the aggregate weight of the edges across the partitions.

- A node set N_d is created, in which each node represents a document in the collection. A node set N_w is created, in which each node represents a word in the collection.

- An undirected bipartite graph $G = (N_d \cup N_w, A)$ is created, such that an edge (i, j) in *A* corresponds to a nonzero entry in the matrix, where $i \in N_d$ and $j \in N_w$.

Co-clustering

Bipartite graph for the co-cluster



Co-clustering steps:

1) Create a graph $G = (N_d \cup N_w, A)$ with nodes in N_d representing documents, nodes in N_w representing words, and edges in *A* with weights representing nonzero entries in matrix *D*.

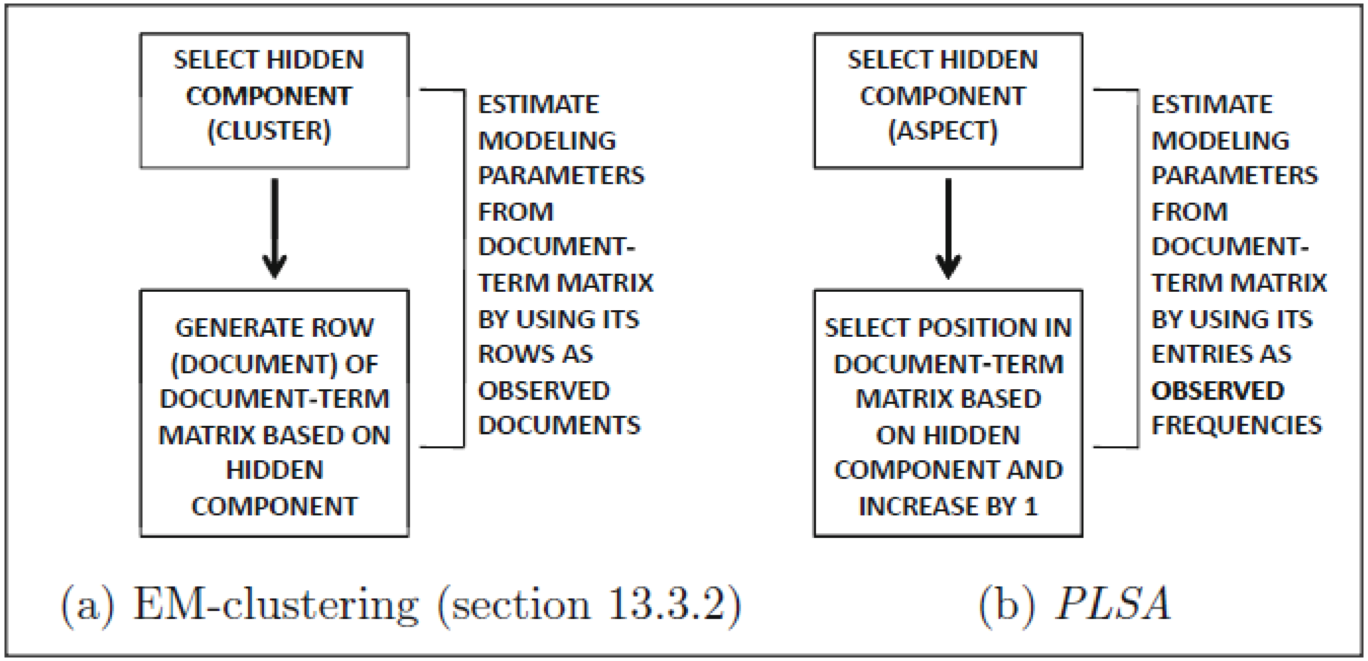
2) Use a *k*-way graph partitioning algorithm to partition the nodes in $N_d \cup N_w$ into *k* groups.

3) Report row–column pairs $(R_i, V_i)$for $i \in \{1 . . . k\}$.

Topic Modeling

Probabilistic Latent Semantic Analysis (PLSA)

- Is an expectation maximization-based mixture modeling algorithm.
- The underlying generative process is optimized to discovering the *correlation structure* of the words rather than the clustering structure of the documents.


SELECT HIDDEN COMPONENT (CLUSTER)
GENERATE ROW (DOCUMENT) OF DOCUMENT-TERM MATRIX BASED ON HIDDEN COMPONENT
ESTIMATE MODELING PARAMETERS FROM DOCUMENT-TERM MATRIX BY USING ITS ROWS AS OBSERVED DOCUMENTS
(a) EM-clustering (section 13.3.2)
SELECT HIDDEN COMPONENT (ASPECT)
SELECT POSITION IN DOCUMENT-TERM MATRIX BASED ON HIDDEN COMPONENT AND INCREASE BY 1
ESTIMATE MODELING PARAMETERS FROM DOCUMENT-TERM MATRIX BY USING ITS ENTRIES AS OBSERVED FREQUENCIES
(b) *PLSA*

Topic Modeling

It is assumed that there are *k aspects* (or latent topics) denoted by $G_1 \cdots G_k$. The generative process builds the document-term matrix as follows:

1) Select a latent component (aspect) G_m with probability $\mathrm{P}(G_m)$

2) Generate the indices (*i, j*) of a document–word pair $(\bar{X}, w_j)$with probabilities $\mathrm{P}(\bar{X}_i \mid G_m)$ and $\mathrm{P}(w_j \mid G_m)$, respectively. Increment the frequency of entry (*i, j*) in the document-term matrix by 1.

$\mathrm{P}(G_m)$, $\mathrm{P}(\bar{X}_i \mid G_m)$ and $\mathrm{P}(w_j \mid G_m)$ need to be estimated from the observed frequencies in the $n \times d$ document-term matrix.

Topic Modeling

- An important assumption in *PLSA* is that the selected documents and words are *conditionally* independent after the latent topical component G_m has been fixed. In other words, the following is assumed:

$$\mathrm{P}\left(\bar{X}_i, w_j \mid G_m\right) = \mathrm{P}(\bar{X}_i \mid G_m) \cdot \mathrm{P}\left(w_j \mid G_m\right)$$

- This implies

$$\mathrm{P}\left(\bar{X}_i, w_j\right) = \sum_{m=1}^{k} \mathrm{P}(G_m) \cdot \mathrm{P}\left(\bar{X}_i, w_j \mid G_m\right)$$

$$= \sum_{m=1}^{k} \mathrm{P}(G_m) \cdot \mathrm{P}(\bar{X}_i \mid G_m) \cdot \mathrm{P}\left(w_j \mid G_m\right)$$

Topic Modeling

- In *PLSA*, the posterior probability $\mathrm{P}\left(G_m|\overline{X}_i, w_j\right)$ of the latent component associated with a particular *document–word pair* is estimated. The EM algorithm starts by initializing $\mathrm{P}(G_m)$, $\mathrm{P}(\overline{X}_i \mid G_m)$ and $\mathrm{P}\left(w_j \mid G_m\right)$ to 1/*k*, 1/*n*, and 1/*d*, respectively. Here, *k*, *n*, and *d* denote the number of clusters, number of documents, and number of words, respectively.

- The algorithm iteratively executes the following E- and M-steps to convergence:

1) (E-step) Estimate posterior probability $\mathrm{P}\left(G_m|\overline{X}_i, w_j\right)$ in terms of $\mathrm{P}(G_m)$, $\mathrm{P}(\overline{X}_i \mid G_m)$ and $\mathrm{P}\left(w_j \mid G_m\right)$

2) (M-step) Estimate $\mathrm{P}(G_m)$, $\mathrm{P}(\overline{X}_i \mid G_m)$ and $\mathrm{P}\left(w_j \mid G_m\right)$ in terms of the posterior probability $\mathrm{P}\left(G_m|\overline{X}_i, w_j\right)$ and observed data about word-document co-occurrence using log-likelihood maximization.

- These steps are iteratively repeated to convergence.

Topic Modeling

The posterior probability estimated in the E-step can be expanded using the Bayes rule:

$$\boxed{P(\mathcal{G}_m|\overline{X_i}, w_j) = \frac{P(\mathcal{G}_m) \cdot P(\overline{X_i}, w_j|\mathcal{G}_m)}{P(\overline{X_i}, w_j)}.}$$

and using previous equation for $\mathrm{P}(\bar{X}_i\ , w_j)$

$$\boxed{P(\mathcal{G}_m|\overline{X_i}, w_j) = \frac{P(\mathcal{G}_m) \cdot P(\overline{X_i}|\mathcal{G}_m) \cdot P(w_j|\mathcal{G}_m)}{\sum_{r=1}^{k} P(\mathcal{G}_r) \cdot P(\overline{X_i}|\mathcal{G}_r) \cdot P(w_j|\mathcal{G}_r)}.}$$

This shows that the E-step can be implemented in terms of the estimated values $\mathrm{P}(G_m)$, $\mathrm{P}(\bar{X}_i \mid G_m)$ and $\mathrm{P}(w_j \mid G_m)$

Topic Modeling

- It remains to show how these values can be estimated using the *observed* word-document co-occurrences in the M-step.

- The posterior probabilities $\mathrm{P}\left(G_m|\overline{X}_i\,, w_j\right)$ may be viewed as weights attached with word-document co-occurrence pairs for each aspect G_m. These weights can be leveraged to estimate the values $\mathrm{P}(G_m)$, $\mathrm{P}(\overline{X}_i \mid G_m)$ and $\mathrm{P}\left(w_j \mid G_m\right)$ for each aspect using maximization of the log-likelihood function.

- The final estimated values will be presented on the next slide.

Topic Modeling

- Let $f(\overline{X}_i, w_j)$ represent the observed frequency of the occurrence of word w_j in document $\overline{X}_i$ in the corpus. Then, the estimations in the M-step are as follows:

$$\boxed{\begin{aligned}
&P(\overline{X_i}|\mathcal{G}_m) \propto \sum_{w_j} f(\overline{X_i}, w_j) \cdot P(\mathcal{G}_m|\overline{X_i}, w_j) \quad \forall i \in \{1 \ldots n\}, m \in \{1 \ldots k\} \\
&P(w_j|\mathcal{G}_m) \propto \sum_{\overline{X_i}} f(\overline{X_i}, w_j) \cdot P(\mathcal{G}_m|\overline{X_i}, w_j) \quad \forall j \in \{1 \ldots d\}, m \in \{1 \ldots k\} \\
&P(\mathcal{G}_m) \propto \sum_{\overline{X_i}} \sum_{w_j} f(\overline{X_i}, w_j) \cdot P(\mathcal{G}_m|\overline{X_i}, w_j) \quad \forall m \in \{1 \ldots k\}.
\end{aligned}}$$

Limitations of PLSA

- The number of parameters grows linearly with the number of documents. Therefore, such an approach can be slow and may overfit the training data because of the large number of estimated parameters.

Specialized Classification Methods for Text

Instance-Based Classifiers

- It works well for text, especially when a preprocessing phase of clustering or dimensionality reduction is performed.

- However because of the sparse and high-dimensional nature of text collections, this basic procedure can be modified in two ways to improve both the efficiency and the effectiveness:

- The first method uses dimensionality reduction in the form of latent semantic indexing.

- The second method uses fine-grained clustering to perform centroid based classification.

Specialized Classification Methods for Text

Leveraging Latent Semantic Analysis

- A major source of error in instance-based classification is the noise inherent in text collections. This noise is often a result of *synonymy* and *polysemy*.

  - *Synonymy* refers to the fact that two words mean approximately the same thing.
  - *Polysemy* refers to the fact that the same word may mean two different things.

- Latent semantic analysis (*LSA*) is an approach that relies on singular value decomposition (*SVD*) to create a reduced representation for the text collection.

  - (*LSA*) method is an application of the *SVD* method to the $n \times d$ document-term matrix *D*, where *d* is the size of the lexicon, and *n* is the number of documents.
  - The eigenvectors with the largest eigenvalues of the square $d \times d$ matrix $D^T D$ are used for data representation. The sparsity of the data set results in a low intrinsic dimensionality.

Specialized Classification Methods for Text

- In the text domain, the reduction in dimensionality resulting from *LSA* is rather drastic.

- For example, we may be able to represent a corpus drawn on a lexicon of size 100,000 in less than 300 dimensions.

- The removal of the dimensions with small eigenvalues typically leads to a reduction in the noise effects of synonymy and polysemy.

Specialized Classification Methods for Text

Centroid-Based Classification

- Fast alternative to *k*-nearest neighbor classifiers.
- The basic idea is to use a clustering algorithm to partition the documents of each class into clusters.
- The number of clusters derived from the documents of each class is proportional to the number of documents in that class.
- Class labels are associated with individual clusters rather than the actual documents.

Specialized Classification Methods for Text

Rocchio Classification

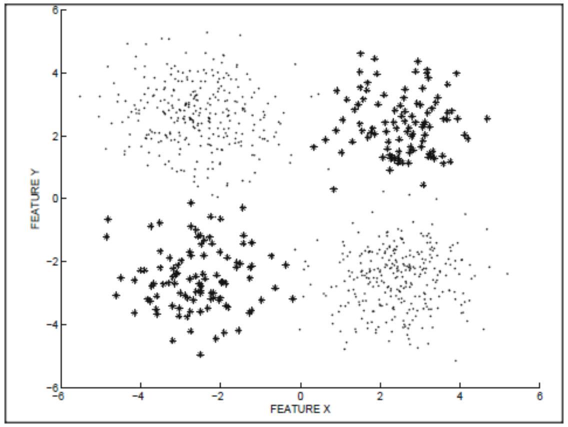
Special case of the of the centroid-based classifier.

- all documents belonging to the same class are aggregated into a *single* centroid.

- For a given document, the class label of the closest centroid is reported.

- Rocchio's method would not work very well if documents of the same class were separated into distinct clusters. In such cases, the centroid of a class of documents may not even lie in one of the clusters of that class.

- A bad case for Rocchio's method is illustrated in Figure



Specialized Classification Methods for Text

Multinomial Bayes Model

Revisit Bernoulli model

- Let *C* be the random variable representing the class variable of an unseen test instance, with *d*-dimensional feature values $\bar{X}(a_1 \cdots a_d)$
- For the Bernoulli model on text data, each value of a_i is 1 or 0, depending on whether or not the i^{th} word of the lexicon is present in the document $\bar{X}$.
- Goal: estimate the posterior probability $P\left(C = c \mid \bar{X} = (a_1 \cdots a_d)\right)$.

- Let the random variables for the individual dimensions of $\bar{X}$ be denoted by $\bar{X} = (x_1 \cdots x_d)$. Then, it is desired to estimate the conditional probability $P(C = c \mid x_1 = a_1, \cdots, x_d = a_d)$

Specialized Classification Methods for Text

Then, by using Bayes' theorem, the following equivalence can be inferred:

$$
\begin{aligned}
&P(C = c \mid x_1 = a_1, \cdots, x_d = a_d) \\
&= \frac{P(C = c)P(x_1 = a_1, \cdots, x_d = a_d \mid C = c)}{P(x_1 = a_1, \cdots, x_d = a_d)} \\
&\propto P(C = c)P(x_1 = a_1, \cdots, x_d = a_d \mid C = c) \\
&\approx P(C = c)\prod_{i=1}^{d} P(x_i = a_i \mid C = c)
\end{aligned}
$$

- In the binary model, each attribute value a_i takes on the value of 1 or 0 depending on the presence or the absence of a word.

- If the fraction of the documents in class *c* containing word *i* is denoted by $p(i, c)$, then the value of $P(x_i = a_i \mid C = c)$ is estimated as either $p(i, c)$ or $1 - p(i, c)$ depending upon whether a_i is 1 or 0, respectively.

Specialized Classification Methods for Text

- Note that this approach explicitly penalizes nonoccurrence of words in documents.
- Word absence is usually weakly related to class labels. This leads to greater noise in the evaluation. Furthermore, differential frequencies of words are ignored by this approach.
- Longer documents are more likely to have repeated words.
- The multinomial model is designed to address these issues.

Specialized Classification Methods for Text

- In the multinomial model, the *L* terms in a document are treated as samples from a multinomial distribution.

- The total number of terms in the document (or document length) is denoted by $L = \sum_{j=1}^{d} a_i$.

  a_i is assumed to be the raw frequency of the term in the document.

- The posterior class probabilities of a test document with the frequency vector $(a_1 \cdots a_d)$ are defined and estimated using the following generative approach:

  1) Sample a class *c* with a class-specific prior probability.

  2) Sample *L* terms *with replacement* from the term distribution of the chosen class *c*.

  3) *Test instance classification:* What is the posterior probability that the class *c* is selected in the first generative step, conditional on the *observed* word frequency $(a_1 \cdots a_d)$ in the test document?

Specialized Classification Methods for Text

- When the sequential ordering of the *L* different samples are considered, the number of possible ways to sample the different terms to result in the representation $(a_1 \cdots a_d)$ is given by

$$\frac{L!}{\prod_{i:a_i>0} a_i!}$$

- The probability of *each* of these sequences is given by

$$\prod_{i:a_i>0} p(i,c)^{a_i}$$

- In this case, $p(i,c)$ is estimated as the fractional number of occurrences of word *i* in class *c including repetitions*.

- Therefore, unlike the Bernoulli model, repeated presence of word *i* in a document belonging to class *c* will increase $p(i,c)$.

Specialized Classification Methods for Text

- If $\mathrm{n}(i, c)$ is the number of occurrences of word *i* in all documents belonging to class *c*, then

$$p(i, c) = \frac{\mathrm{n}(i, c)}{\sum_i \mathrm{n}(i, c)}$$

- Then, the class conditional feature distribution is estimated as follows:

$$P(x_1 = a_1, \cdots, x_d = a_d \mid C = c) \approx \frac{L!}{\prod_{i:a_i>0} a_i!} \prod_{i:a_i>0} p(i, c)^{a_i}$$

Specialized Classification Methods for Text

Using the Bayes rule, the multinomial Bayes model computes the posterior probability for a test document as follows:

$$\begin{aligned}
&P(C = c \,|x_1 = a_1, \cdots, x_d = a_d) \\
&\propto P(C = c\,) \cdot P(x_1 = a_1, \cdots, x_d = a_d \,|\, C = c) \\
&\approx P(C = c\,) \cdot \frac{L!}{\prod_{i:a_i>0} a_i!} \prod_{i:a_i>0} p(i,c)^{a_i} \\
&\propto P(C = c\,) \cdot \prod_{i:a_i>0} p(i,c)^{a_i}
\end{aligned}$$

- The constant factor $\frac{L!}{\prod_{i:a_i>0} a_i!}$ has been removed from the last condition because it is the same across all classes.
- Note that in this case, the product on the right-hand side only uses those words *i*, for which a_i is strictly larger than 0. Therefore, nonoccurrence of words is ignored.

Specialized Classification Methods for Text

- In this case, we have assumed that each a_i is the raw frequency of a word, which is an integer.

- It is also possible to use the multinomial Bayes model in which the frequency a_i might be fractional. However, the generative explanation becomes less intuitive in such a case.

Thank You

Dragos Bozdog

For academic use only.