Data Mining, Lecture 8

Text Data Mining

S. Nõmm

[1] Department of Software Science, Tallinn University of Technology

31.10.2018

Introduction

- A text document is a discrete sequence of words, also referred to as a string.
- In practice, text is usually represented as multidimensional data in the form of frequency annotated bag-of-words.
- Typically, a preprocessing approach is applied in which the very common words are removed, and the variations of the same word are consolidated. This approach leads to an unordered set of words, where normalized frequencies are associated with normalized words.
- The overall dimensionality is equal to the number of distinct words in the lexicon.

Specific characteristics of the text data

- **Number of zero attributes;** A single document may contain only a few hundred words (lexicon may be much larger few thousands). 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*. This has implications for the distance computation.
- The Euclidean distance function cannot compute the distance between two short documents in a comparable way to that between two long documents because the latter will usually be larger.

Specific characteristics of the text data

- **Nonnegativity:** The frequencies of words take on nonnegative values. When combined with high-dimensional sparsity, the nonnegativity property enables the use of specialized methods for document analysis. In general, all data mining algorithms must be cognizant of the fact that the presence of a word in a document is statistically more significant than its absence. Unlike traditional multidimensional techniques, incorporating the global statistical characteristics of the data set in pairwise distance computation is crucial for good distance function design.
- **Side information:** In some domains, such as the Web, additional side information is available. Examples include hyperlinks or other metadata associated with the document. These additional attributes can be leveraged to enhance the mining process further.

Document Preparation

- Stop word removal
- Stemming. Variations of the same word need to be consolidated. (Singular/plural,tenses, cases etc.)
- Punctuation marks removal.

Document Normalization (inverse document frequency)

Inverse document frequency: Higher frequency words tend to contribute noise to data mining operations such as similarity computation. The removal of stop words is motivated by this aspect. The concept of inverse document frequency generalizes this principle in a softer way, where words with higher frequency are weighted less.

$$f_i^{inv} = \log\left(\frac{n}{n_i}\right)$$

Document Normalization (frequency damping)

Frequency damping: The repeated presence of a word in a document will typically bias the similarity computation significantly. 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. (frequency damping is optional, and the effects vary with the application). For example clustering may show better results without damping.
Let $\bar{X} = \big(x_1, \ldots, x_d\big)$ be the word frequency vector of a document. Damping functions as logarithm or square root may be applied.

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

The normalized frequency $h(x_i)$ is defined as follows:

$$h(x_i) = f(x_i) * f_i^{inv}$$

Similarity measures

Let $\bar{X} = (x_1, \dots, x_d)$ and $\bar{Y} = (y_1, \dots, y_d)$ be the word representations of the two documents.

- Cosine similarity measure is defined as follows:

$$\cos\left(\bar{X}, \bar{Y}\right) = \frac{\sum_{i=1}^{d} h(x_i)h(y_i)}{\sqrt{\sum_{i=1}^{d} h^2(x_i)}\sqrt{\sum_{i=1}^{d} h^2(y_i)}}$$

- In some cases Jaccard coefficient is used as the similarity measure. (it is better suited for the binary data)

$$J\left(\bar{X}, \bar{Y}\right) = \frac{\sum_{i=1}^{d} h(x_i)h(y_i)}{\sum_{i=1}^{d} h^2(x_i) + \sum_{i=1}^{d} h^2(y_i) + \sum_{i=1}^{d} h(x_i)h(y_i)}$$

Clustering: representative based algorithms

Two major modification are required to adjust k-means type algorithms for text data:

- Instead of the Euclidean distance, the cosine similarity function is used.
- Modification of the computation of the cluster centroid. All words in the centroid are not retained. The low-frequency words in the cluster are projected out. Typically, a maximum of 200 to 400 words in each centroid are retained. This is also referred to as a cluster digest.
- A specialized variation of the k-means for text, which uses concepts from hierarchical clustering, will be discussed in this section.

Clustering: Scatter/Gather Approach

The scatter/gather approach uses a combination of hierarchical partitioning and k-means clustering in a two-phase approach.

1. Apply either the buckshot or fractionation procedures to create a robust set of initial seeds.
2. Apply a k-means approach on the resulting set of seeds to generate the final clusters.

Clustering: Scatter/Gather Approach

- *Buckshot:* Let k be the number of clusters to be found and n be the number of documents in the corpus. The buckshot method selects a seed superset of size $\sqrt{k \cdot n}$ and then agglomerates them to k seeds. Straightforward agglomerative hierarchical clustering algorithms are applied to this initial sample of seeds.
- *Fractionation:* method works with all the documents in the corpus. It initially breaks up the corpus into n/m buckets, each of size $m > k$ documents. An agglomerative algorithm is applied to each of these buckets to reduce them by a factor $\nu \in (0, 1)$. This step creates $\nu \cdot m$ agglomerated documents in each bucket, and therefore $\nu \cdot n$ agglomerated documents over all buckets. An agglomerated document is defined as the concatenation of the documents in a cluster. The process is repeated by treating each of these agglomerated documents as a single document. The approach terminates when a total of k seeds remains.

Fractionation: Partitioning into buckets

- Random partitioning of the documents.
- Sort the documents by the index of the j th most common word in the document.
- Contiguous groups of m documents in this sort order are mapped to clusters.

Possible enhancement procedures

- Split operation: The process of splitting can be used to further refine the clusters into groups of better granularity. This can be achieved by applying the buckshot procedure on the individual documents in a cluster by using $k = 2$ and then re clustering around these centers.
- Join operation: The join operation merges similar clusters into a single cluster. To perform the merging, the topical words of each cluster are computed, as the most frequent words in the centroid of the cluster. Two clusters are considered similar if there is significant overlap between the topical words of the two clusters.

Probabilistic approach

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.

- Let us suppose that available documents should be assigned to the k clusters $\left(\mathcal{G}_1, \ldots, \mathcal{G}_k\right)$.
- Bayes classifier is used to estimate the posterior probability $P(\mathcal{G}_m|\bar{X})$ in the E-step.
- The conditional feature distribution $P(w_j|\mathcal{G}_m$ for word w_j is estimated from these posterior probabilities in the M-step as follows:

$$P\left(w_j|\mathcal{G}_m\right) = \frac{\sum_{\bar{X}} P\left(\mathcal{G}_m|\bar{X}\right) I\left(\bar{X}, w_j\right)}{\sum_{\bar{X}} P\left(\mathcal{G}_m|\bar{X}\right)}$$

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

Probabilistic approach

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

$$P\left(\mathcal{G}_m|\bar{X}\right) =\propto P(\mathcal{G}_m) \prod_{w_j \in \bar{X}} P\left(w_j|\mathcal{G}_m\right) \prod_{w_j \notin \bar{X}} \left(1 - P\left(w_j|\mathcal{G}_m\right)\right)$$

- (M-step) Estimate conditional distribution$P\left(w_j|\mathcal{G}_m\right)$ of words

$$P\left(w_j|\mathcal{G}_m\right) = \frac{\sum_{\bar{X}} P\left(\mathcal{G}_m|\bar{X}\right) I\left(\bar{X}, w_j\right)}{\sum_{\bar{X}} P\left(\mathcal{G}_m|\bar{X}\right)}$$

and prior probabilities $P\left(\mathcal{G}_m\right)$ of different clusters using the estimated probabilities in the E-step.

Simultaneous Document and Word Cluster Discovery

Co-clustering

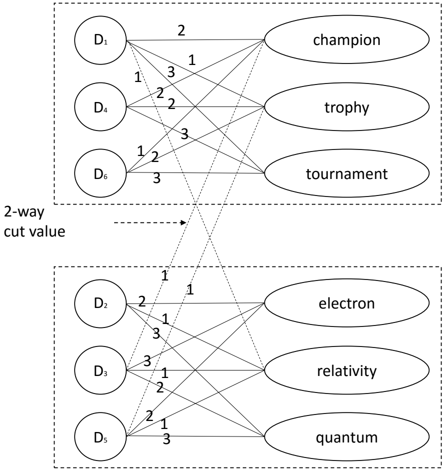
- The idea in co-clustering is to rearrange the rows and columns in the data matrix so that most of the nonzero entries become arranged into blocks.
- In the context of text data, this matrix is the $n \times d$ document term matrix D, where rows correspond to documents and columns correspond to words.
- The i th cluster is associated with a set of rows $\mathcal{R}_i$ (documents), and a set of columns $\mathcal{V}_i$ words.

Simultaneous Document and Word Cluster Discovery

	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

	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

Sports cluster
Physics cluster



CoClustering

- Convert the problem to a bipartite graph partitioning problem, such that 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 $j \in N_w$.
- A partitioning of this graph represents a simultaneous partitioning of the rows and columns.

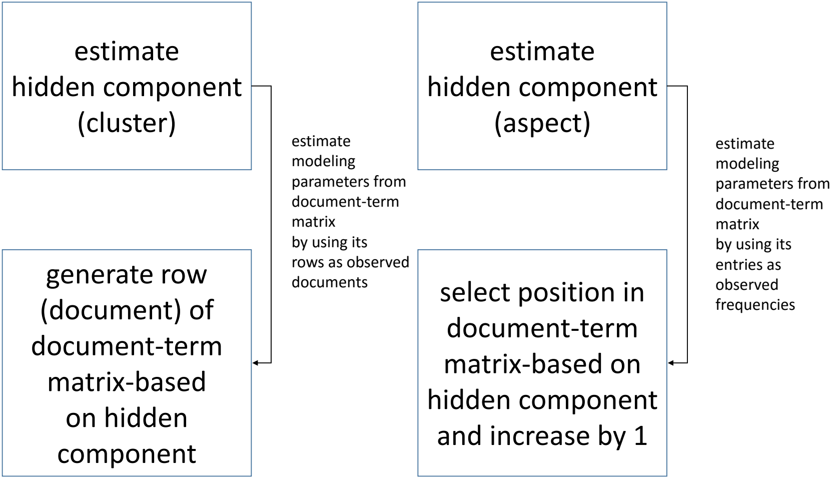
CoClustering, more formally

- Create a graph G $= G = (N_d \cup N_w, A)$ 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.
- Use a k-way graph partitioning algorithm to partition the nodes in $N_d \cup N_w$ into k groups.
- Report rowcolumn pairs $(\mathcal{R}_i, \mathcal{V}_i)$ for $i \in \{1, \ldots, k\}$. Here, $\mathcal{R}_i$ represents the rows corresponding to nodes in N_d for the ith cluster, and $\mathcal{V}_i$ represents the columns corresponding to the nodes in N_w for the ith cluster.

Topic Modeling

- Topic modeling can be viewed as a probabilistic version of the latent semantic analysis (LSA) method.
- Probabilistic latent semantic analysis is an expectation maximization-based mixture, EM - algorithm is used in a different way.
- The underlying generative process is different, and is optimized to discovering the correlation structure of the words rather than the clustering structure of the documents.
- Main difference is in the generative process.

EM versus PLSA



estimate hidden component (cluster)

estimate modeling parameters from document-term matrix by using its rows as observed documents

generate row (document) of document-term matrix-based on hidden component

estimate hidden component (aspect)

estimate modeling parameters from document-term matrix by using its entries as observed frequencies

select position in document-term matrix-based on hidden component and increase by 1

PLSA

- In PLSA, the generative process is designed for dimensionality reduction whereas different parts of the same document can be generated by different mixture components.
- Assume that there are k *aspects* latent topics $\mathcal{G}_1, \ldots, \mathcal{G}_k$.
- The generative process builds the document-term matrix as follows:
  1. Select an aspect (latent component) $\mathcal{G}_m$ with probability $P(\mathcal{G}_m)$.
  2. Generate the indices (i, j) of a documentword pair $(\bar{X}_i, w_j)$ with probabilities $P(\bar{X}_i|\mathcal{G}_m)$ and $P(w_j|\mathcal{G}_m)$, respectively. Increment the frequency of entry (i, j) in the document-term matrix by 1. The document and word indices are generated in a probabilistically independent way.
- All the parameters of this generative process, such as $P(\mathcal{G}_m)$, $P(\bar{X}_i|\mathcal{G}_m)$, and $P(w_j|\mathcal{G}_m)$, need to be estimated from the observed frequencies in the $n \times d$ document-term matrix.
- NB! aspects are analogues to the clusters but not equivalent.

PLSA

- PLSA assumes that the selected documents and words are conditionally independent after the latent topical component $\mathcal{G}_m$ has been fixed.

$$P\Big(\bar{X}_i, w_j|\mathcal{G}_m\Big) = P\Big(\bar{X}_i|\mathcal{G}_m\Big)P\Big(w_j|\mathcal{G}_m\Big)$$

- This implies that the joint probability for selecting a documentword pair can be expressed in the following way:

$$P\Big(\bar{X}_i, w_j\Big) = \sum_{m=1}^{k} P\Big(\mathcal{G}_m\Big)P\Big(\bar{X}_i, w_j|\mathcal{G}_m\Big) =$$

$$= \sum_{m=1}^{k} P\Big(\mathcal{G}_m\Big)P\Big(\bar{X}_i|\mathcal{G}_m\Big)P\Big(w_j|\mathcal{G}_m\Big)$$

- It is important to note that local independence between documents and words within a latent component does not imply global independence between the same pair over the entire corpus.

EM-algortithm

The EM-algorithm starts by initializing $P(\mathcal{G}_m)$, $P(\bar{X}_i|\mathcal{G}_m)$, and $P(w_j|\mathcal{G}_m)$ 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.

- (E-step) Estimate posterior probability $P(\mathcal{G}_m|\bar{X}_i, w_j)$ in terms of $P(Gm)$, $P(\bar{X}_i|\mathcal{G}_m)$, and $P(w_j|\mathcal{G}_m)$.
- (M-step) Estimate $P(\mathcal{G}_m$, $P(\bar{X}_i|\mathcal{G}_m)$ and $P(w_j|\mathcal{G}_m)$ in terms of the posterior probability $P(\mathcal{G}_m|\bar{X}_i, w_j)$, and observed data about word-document co-occurrence using log-likelihood maximization.
- The posterior probability estimated in the E-step can be expanded using the Bayes rule:

$$P\left(\mathcal{G}_m|\bar{X}_i, w_j\right) = \frac{P(\mathcal{G}_m)P\left(\bar{X}_i, w_j|\mathcal{G}_m\right)}{P\left(\bar{X}_i, w_j\right)}$$

Classification

- Instance-Based Classifiers: The simplest form of the nearest neighbor classifier returns the dominant class label of the top-k nearest neighbors with the cosine similarity. Weighting the vote with the cosine similarity value often provides more robust results.
  - Leveraging Latent Semantic Analysis (synonymy and polysemy)
  - Centroid-Based Classification
  - Rocchio Classification. Assumes that the documents in the same class form a contiguous region, and regions of different classes do not overlap. This allows to aggregate documents belonging to the same class into a single centroid. For a given document, the class label of the closest centroid is reported.
- Bayes Classifiers