

Framework for Clustering Linguistic Data, Its Algorithm for Clustering and Implementation using PERL

Anshu Parashar*, Vinay Chopra**
Department of Computer Sc. & Engg.
*HCTM Kaithal, ** DAVIET Jalandhar

Abstract- Developing data mining algorithms for linguistic data has emerged as an important problem. For streaming data, the assumption is that the data records can be examined only once. Clustering plays a crucial role in organizing large document collections. As an example clustering can be used to structure query results, hence providing users with an overview of the results that is easier to understand and process than a flat list of documents. Clustering text data online as it comes in is a difficult problem. It is both hard to capture a meaningful notion of linguistic similarity and to cluster large amounts of data in a single pass. This problem is especially challenging because most known algorithms that ensure tight clustering are inefficient on large datasets. In this paper, we are implementing a single-pass text-clustering algorithm designed specifically for clustering of Linguistic Data like news stories etc. and examine its empirical behavior. The key to the approach we develop in this work is to use simple representations of documents and clusters as vectors in high dimensional vector spaces and to compute cosine distances between them. Using these distances, articles are clustered in a single pass, which gives certain running time guarantees, and increase cluster weights (or importance) when new articles are added and reduced the weights over time to simulate news events becoming less relevant. In this paper, we will present an overview of clustering and its development over time. We will then present the details of our clustering framework for linguistic data (especially news data).

I. INTRODUCTION

A. CLUSTER ANALYSIS

The problem of cluster analysis is a collection of statistical methods, which identifies groups of samples that behave similarly or show similar characteristics. In common parlance it is also called look-a-like groups. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. The objective is to cluster the data in such a way as to minimize the intra-cluster data point distances while maximizing inter-cluster distances. A cluster of data objects can be treated collectively as one group in many applications. In this way, clusters and groups are interchangeable words. Clustering is an unsupervised classification as it has no predefined classes and few of its typical applications include:

- As a stand-alone tool to get insight into data distribution
- As a pre-processing step for other algorithms

A good clustering method will produce high quality clusters with high intra-class similarity and low inter-class similarity. The quality of a clustering result depends on both

the similarity measure used by the method and its implementation. The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns. After computing these distances it cluster the news in a single pass and creating cluster as desire.

B. MAJOR CLUSTERING APPROACHES

There exist a large number of clustering algorithms in the literature. The choice of clustering algorithm depends both on the type of data available and on the particular purpose and application. If cluster analysis is used as a descriptive or exploratory tool, it is possible to try several algorithms on the same data to see what the data may disclose.

In general, major clustering methods can be classified into the following categories:--

Partitioning algorithms: Given k, the number of clusters to construct, a partitioning method creates an initial partitioning. Then, it uses an iterative reallocation technique that attempts to improve the partitioning by moving objects from one group to another. To achieve global optimality, partitioning based clustering would require the exhaustive enumeration of all of the possible partitions. Two popular heuristics:

- The k-means algorithm
- The k-medoids algorithm

Hierarchy algorithms: A hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed. The agglomerative approach, also called the bottom-up approach, starts with each object forming a separate group. It successively merges the objects or groups close to one another, until all of the groups are merged into one. The divisive approach, also called the top-down approach, starts with all the objects in the same cluster. In each successive iteration, a cluster is split up into smaller clusters, until eventually each object is in one cluster.

Density-based: the given cluster is growing as long as the density of objects in the "neighborhood" exceeds some threshold; i.e. for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points (DBSCAN, OPTICS)

Grid-based: the object space is divided into a finite number of cells that form a grid structure. All of the clustering

operations are performed on the grid structure (STING, CLIQUE).

Model-based: these methods hypothesize a model for each cluster and find the best fit of the data to the given model.

C. DISTANCE MEASURES

To compute whether a set of points are close enough to be considered a cluster, we need a distance measure - $D(x, y)$.

The usual axioms for a distance measure D are:

- $D(x, x) = 0$
- $D(x, y) = D(y, x)$
- $D(x, y) \leq D(x, z) + D(z, y)$ the triangle inequality

Assume a k -dimensional Euclidean space, the distance between two points,

$x = [x_1, x_2 \dots x_k]$ and $y = [y_1, y_2 \dots y_k]$ may be defined using one of the measures:

- *Euclidean distance:*

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

- *Manhattan distance:*

$$\sum_{i=1}^k |x_i - y_i|$$

- *Max of dimensions:*

$$\max_{i=1}^k |x_i - y_i|$$

- *Minkowski distance:*

$$\left(\sum_{i=1}^k (|x_i - y_i|)^p \right)^{1/p}$$

When there is no Euclidean space in which to place the points, clustering becomes more difficult: Web page accesses, DNA sequences, customer sequences, categorical attributes, documents, etc.

II. LINGUISTIC DATA CLUSTERING

A. APPROACH

Cluster linguistic data requires addressing the issues on linguistic similarity. A choice for distance function is not only based on its mathematic properties but also human interpretations of linguistic similarities. The specific instance of linguistic data we want to focus on is news stories. In study of news stories we find that News stories[2] are having their on linguistic notation generally used and while in clustering its require to change the clusters over the time as new news comes. Due to the huge size of news datasets , it is not feasible to store all of the data. we only focus to find a appropriate approach for clustering .In manual clustering human could be bias in selection of news to cluster , so computerization of clustering is more However, our goal is to be able to cluster

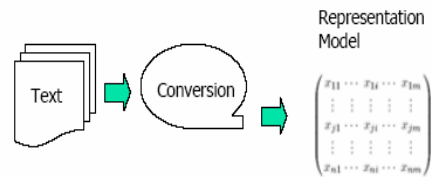
news very quickly using a standard desktop machine. Finding a good way to do computerized clustering would allow for many applications including helping people file text data on computer, find patterns in email [2], keep track of the most important news, etc. In our algorithm, we are using vector in high dimensional vector space for representing documents and clusters and then used cosine distances between vectors. Linguistic data is ubiquitous. As the volume of text data increases, management and analysis of data becomes very important. Linguistic data mining is an emerging technology for handling the increasing Linguistic data. Linguistic data clustering is one of the fundamental functions in text mining. Text clustering is to divide a collection of text documents into different category groups so that documents in the same category group describe the same topic, such as news, classic music or Chinese history.

B. REPRESENTATION MODEL

In information retrieval and text mining, text data [3] of different formats is represented in a common representation model, e.g.,

$$\begin{pmatrix} x_{11} & \dots & x_{1i} & \dots & x_{1m} \\ \vdots & & \vdots & & \vdots \\ x_{j1} & \dots & x_{ji} & \dots & x_{jm} \\ \vdots & & \vdots & & \vdots \\ x_{n1} & \dots & x_{ni} & \dots & x_{nm} \end{pmatrix}$$

Vector Space Model



Vector Space Model (VSM)

The most popular representation model used in information retrieval and text mining .In VSM, a text document is represented as a vector of terms

$$\langle t_1, t_2, \dots, t_i, \dots, t_n \rangle.$$

Each term t_i represents a word or a phrase.

The set of all n unique terms in a set of text documents forms the vocabulary for the set of documents. A set of documents is represented as a set of vectors that can be written as a matrix. where each row represents a document, each column indicates a term, and each element x_{ji} represents the frequency of the i^{th} term in the j^{th} document. Three ways to represent a term value x_{ji}

Frequency representation:

x_{ji} is the frequency of term i in document j

Binary representation:

$x_{ji} = 1$ indicates that term i occurs in document j , otherwise, $x_{ji} = 0$

Term frequency-inverted document frequency (tfidf):

$Tfidf(t_k, d_j) = \#(t_k, d_j) * \log(|Tr| / \#Tr(t_k))$, Where $\#(t_k, d_j)$ denotes the number of times term t_k occurs in document d_j

$\#Tr(t_k)$ denotes the number of documents in Tr in which t_k occurs.

$|Tr|$ denotes the number of documents

C.LINGUISTIC DISTANCES AND SIMILARITIES

The most commonly used distance measures in text clustering [2] are measures we have seen elsewhere. A few examples include taking the intersection distance of the two sets of words

$$d(A, B) = 1 - \frac{A \cap B}{A \cup B}$$

Converting the data to vectors and taking the Euclidian distance

$$d(\vec{x}, \vec{y}) = \sum_{i=1}^m (x_i - y_i)^2$$

less common is the L1 norm

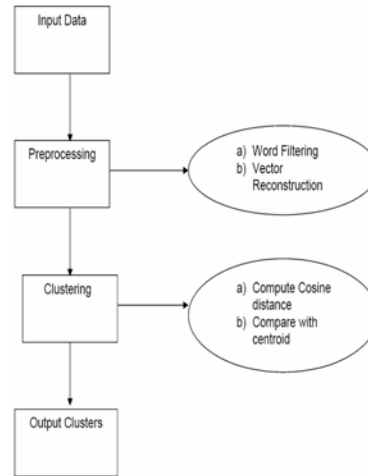
$$d(\vec{x}, \vec{y}) = \sum_{i=1}^m |x_i - y_i|$$

and cosine distance

$$d(\vec{x}, \vec{y}) = 1 - \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|}$$

On top of these basic distance measures, more recent research has included methods for using the word-sets of the articles to determine their similarities. Term Frequency Inverse Document Frequency [3], or tfidf weights each word according to how frequently it is expected to appear in the dataset, the intuition being that more common words should be weighted lower because they contain less information than rare ones. Doing this transformation before applying a known distance function would theoretically make the distance measures more meaningful.

III. LINGUISTIC CLUSTERING FRAMEWORK



Major tasks are: --

Pre-processing

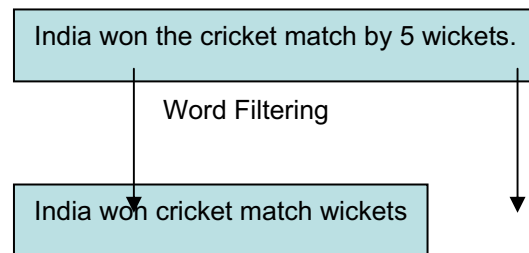
It involves following tasks:

- Word Filtering
- Vector construction

Word Filtering:

Algorithm uses TFIDF approach to remove most frequent words from the document. TFIDF refers to Term Frequency inverse document frequency. As the words, which are, occurring with a higher frequency, are not giving us any information. Therefore using some language information and some practical experiments, it removes some highly frequent words. The words in this category are mainly common prepositions, linking verbs, and pronouns.

Example:



Vector Construction:

After filtering the high frequency words it reconstructs the document, which now consists of very fewer words than the original document. Now this document needs to be converted to some vector (using Vector Space Representation) form for the purpose of calculate some distance with other document (or Cluster Centroids in this case). In this each article is stored as a vector in high dimensional space. Each possible word would represent a dimension in this vector space. If an article contained a word, it would have 1 added to that dimension. At the end, the vector would be normalized. Hence, articles containing many different words would be comprised of many components but each of small value since the vector would be

normalized. Articles with few words would contain few components, but each would have greater value.

Clustering

The clustering component of the algorithm is implemented in a straightforward manner. Each vector is compared to each other vector then added to the closest cluster if it passed the threshold. Since clusters were represented as vectors in high dimensional space[2], it would have been hard to not go through all of them for each new data point.

Each article is stored as a vector, which is implemented as a hash map, with words as keys and their values as the length of that dimension in the vector. The distance measure used is the Cosine Distance between two vectors.

The Cosine distance is

$$d(\vec{x}, \vec{y}) = 1 - \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|}$$

as example let two vectors are:--

x	
	1
	1
Cricket	1
Match	1
Wicket	1

y	
Pakistan	1
Lost	1
Game	2
Wickets	3

Cosine Distance =
 1- (1*0 + 1*0 + 1*0 + 1*0 + 1*3)
 = 1-3
 = -2

- Match with an adaptive threshold.
- Assign to the cluster
- Update Clusters

IV. ALGORITHM

For each document:

Repeat

Begin

Remove unwanted words from txt input files and create new files

For (each of the above files after removal)

Count number of occurrence of each word using *TFIDF* and store in a txt file

$Tfidf(t_k, d_j) = \#(t_k, d_j) * \log(|Tr|/\#Tr(t_k))$, Where

$\#(t_k, d_j)$ denotes the number of times term t_k occurs in document d_j

$\#Tr(t_k)$ denotes the number of documents in Tr in which t_k occurs

$|Tr|$ denotes the number of documents

For (each of the above output files)

Create vector representation of each file using *VSM*

In *VSM*, a text document is represented as a vector of terms

$\langle t_1, t_2, \dots, t_i, \dots, t_n \rangle$.

Where each term t_i represents a word or a phrase of the document.

Repeat

Read vector for each file and compute the cosine distance between vectors by using *cosine distance formula*

$$d(\vec{x}, \vec{y}) = 1 - \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|}$$

Where x, y are vector representation of documents

If (cosine distance is within threshold distance of the cluster)

Add to the cluster.

Else

Begin a new cluster.

End

V. RESULTS

We tested our algorithm on around 50 news articles and got accuracy around 60%. If the value of the threshold is properly chosen then we can improve it further. These results we get by implementing above algorithm in PERL.

CONCLUSIONS & ANALYSIS

Our Algorithm is data sensitive, in the sense that even if two news belong to same class they not belong to same cluster. While much of the notion of similarity can be captured by word counting techniques, we need some more sophisticated algorithms/Dictionary help, for this task.

REFERENCES

[1] Clustering Data Streams: Theory and Practice Sudipto Guha , Adam Meyerson Nina Mishra, Rajeev Motwani ,Liadan O’Callaghan
 [2] Online Clustering of Linguistic data”, Lev Reyzin, Princeton University.
 [3] Text Clustering:Algorithms, Semantics and Systems Joshua Zhexue Huang & Michael Ng. & Liping Jing.