Model-Based Clustering of Short Text Streams

By

Bellamkonda Sachin

17MA20008

Under the supervision of

Pawan Kumar

Department of Mathematics

Indian Institute of Technology Kharagpur

Abstract

The importance of short text stream clustering has increased due to major growth of short text in social medias. Here we discuss about a streaming algorithm that works on short text streams, so unlike normal text streams the problem with short text streams is that the frequency of words occurring is less and this faces the sparsity problem. Many Classification models like Dynamic Topic Model (DTM), Topic Tracking Model (TTM) etc. are proposed but these models assume that each document has rich contents, most of these don’t work on short text streams and also most of them assume fixed number of clusters so they can’t deal with the topic drift property efficiently and also the problem arrived due to the common words in different clusters (multi-Cluster words). Our results have shown that MStream algorithm with removing the common cluster words has given us better results than MStream algorithm.

Introduction

Short text streams emerging on the internet have become more and more popular in the recent times. The task on topic modelling these short text streams from the documents incoming has many applications such as text summarization, recommendation engines, google search snippets etc.

The problem of general streaming data is that the data will be coming continuously to our model so storing the documents and iterating them again and again is out of equation. Each document may come at a onetime which is one pass scheme also more generally they come in batches which is batch scheme. Most of the data coming will be in batches and new clusters are evolving from this continuous inflow of batches. The problem of short text streams over large text data documents is that the words in short texts appear less frequent this led to data sparsity problem. Since this is a streaming data there can be new topics evolving over the period of time. Therefore, the documents coming may not set into the clusters that are already created from the previous documents, this is called concept drift problem. This concept drift problem makes the prediction less accurate as time progresses because of the change in distribution of data over the time. And also, when the clusters are created there can be some homograph words in the clusters that are already belong to different cluster, but due to these words there is a probability that a new document may go into the existing cluster although it belongs to another new cluster.

So, we are going to work on an algorithm introduced to tackle the above-mentioned problems. First, we introduce a MStream clustering Algorithm works completely fine with both the one pass scheme and batch scheme. With experimental results shown that even with the one pass scheme MStream algorithm is getting good results and with the batch scheme as we can iterate every batch multiple times until next batch arrives, we can get even better results compared to one pass scheme. Instead of giving the probability of each document belong to different topics we rather assign each document belongs to one topic. By this way we can handle data sparsity problem in incoming data.

The number of clusters are not fixed before running the model so, the data drift problem is solved because when the new topic arises the document can go into newly created clusters. As in the daily life application the twitter data (short text data) coming will increase day by day so running time to cluster these documents and the space used to store these documents will increase exponentially and also, we are interested in topics of specific period of time. So, for this we introduce a new algorithm called MStreamF algorithm which solve these problems. We will build this on MStream algorithm with removing old texts which is done by removing old clusters in the previous batches. We will remove the outdated clusters by having a fixed number of batches (fixed before running the model) and these batch of batches need to be stored for the estimation of new batches when number of batches exceeds this threshold value, we remove those clusters that are associated with old batches that is we will be removing the information from the cluster feature.

In this model to handle the problem of multi cluster words existing in the clusters we remove the we remove the multi cluster words with high entropy which can reduce the topic ambiguity problem.

Related Work

Text stream data clustering, we can classify them as shown below.

1.Similarity Based Stream Clustering

This type of method mainly chooses vector-based model to find the similarity between the documents by representing documents as vectors and finding similarity between them using cosine similarity. For example, CluStream[10] based on the concept of micro clusters. Micro clusters are data structures which summarize a set of instances from the stream. CluStream has both online and offline phase and uses pyramidal time frame.

The drawback of this type of clustering is that there is a threshold value that needs to be fixed for the documents to go into new cluster or join the existing cluster. Unlike this in our method we do not fix any threshold for our documents to go into existing cluster or not, instead in the proposed method it calculates probabilities whether a new cluster need to be created fir the document or it can fit into the existing one. By following this method, we can overcome the concept drift problem in the streaming data.

2.Model Based Stream Clustering

Model based clustering considers the data are generated by a mixture of underlying probability distributions. Differently from k-means, in model-based clustering each document has a probability to which cluster the document needs to be assigned.

There are many different methods like Latent Dirichlet Allocation (LDA) [11], streaming LDA(ST-LDA) [12], topic tracking model (TTM) [13]. But these methods do not handle well for the short text streams since the information in document is not enough for the document multinomial distributions in clusters There is a method Dynamic Clustering Topic Model (DCT) which can be used for short text streams in which each document coming is assigned to single topic unlike the probabilistic to each and every cluster.

In the traditional methods we can see that the number of topics (or) clusters are fixed from starting which is not true in case of text streams where we have the concept drift problem and the short text problem.

Approach

We have divided this into three parts initially we introduce how the documents, clusters are shown in this model and then we implement MStream algorithm which works with data sparsity and data drift problem was extended to MStream Algorithm with some rules to remove the outdated clusters in the cluster feature.

Representation

In general similarity based clustering methods keep term frequency-inverse document frequency (TF-IDF) which are used for estimating the corresponding weights to the words that are in the clusters. Differently, in this method each document is a collection of words from the streaming data and the frequencies for each of the words that are present in the document and the other difference is that in our representation we assume documents are from a multinomial distribution.

Here to represent a document a cluster feature is used. Here cluster feature is tuple with three characters co-occurrences of the words in the cluster z, number of documents that are present in each of the cluster (m_z) and number of words (words from the documents) in each cluster z(n_z). The cluster feature is very useful when a new document is updated into the cluster or deleted from the cluster, we just need to update the above three variables.

MStream Algorithm

This is a model based clustering algorithm. This can work for one pass scheme and updated version of this can work for the batch process as well and the results will get even better after multiple iterations of the batches until the next batch arrives.