An Online Semantic-enhanced Dirichlet Model for Short Text Stream Clustering

Jay Kumar and **Junming Shao**[*] and **Salah ud din**
Data Mining Lab, School of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu, 611731, China
jay_tharwani1992@yahoo.com, junmshao@uestc.edu.cn

Wazir Ali
SMILE Lab, School of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu, 611731, China

Abstract

Clustering short text streams is a challenging task due to its unique properties: infinite length, sparse data representation and cluster evolution. Existing approaches often exploit short text streams in a batch way. However, determine the optimal batch size is usually a difficult task since we have no prior knowledge when the topics evolve. In addition, traditional independent word representation in the graphical model tends to cause "term ambiguity" problem in short text clustering. Therefore, in this paper, we propose an Online Semantic-enhanced Dirichlet Model for short text stream clustering, called OSDM, which integrates the word-occurrence semantic information (i.e., context) into a new graphical model and clusters for each arriving short text automatically in an online way. Extensive results have demonstrated that OSDM gives better performance compared to many state-of-the-art algorithms on both synthetic and real-world data sets.

1 Introduction

A massive amount of short text data is constantly generated with online social platforms such as microblogs, Twitter and Facebook. Clustering of such short text streams has thus gained increasing attention in recent years due to many real-world applications like event tracking, hot topic detection, and news recommendation (Hadifar et al., 2019). However, due to the unique properties of short text streams such as infinite length, evolving patterns and sparse data representation, short text stream clustering is still a big challenge (Aggarwal et al., 2003; Mahdiraji, 2009).

During the past decade, many approaches have been proposed to address the text stream clustering problem from different points of view, and each method comes with specific advantages and drawbacks. Initially, traditional clustering algorithms for static data were enhanced and transformed for text streams (Zhong, 2005). Very soon, they are replaced by model-based algorithms such as LDA (Blei et al., 2003), DTM (Blei and Lafferty, 2006), TDPM (Ahmed and Xing, 2008), GSDMM(Yin and Wang, 2016b), DPMFP (Huang et al., 2013), TM-LDA (Wang et al., 2012), NPMM (Chen et al., 2019) and MStream (Yin et al., 2018), to mention a few. However, for most established approaches, they often work in a batch way, and assume the instances within a batch are interchangeable. This assumption usually cannot hold for topic-evolving text data corpus. Determining an optimal batch size is also a non-trivial task for different text streams (Howard and Ruder, 2018).

Additionally, unlike long text documents, short text clustering further suffers from the lack of supportive term occurrence to capture semantics (Gong et al., 2018). For most existing short text clustering algorithms like Sumblr (Shou et al., 2013), DCT (Liang et al., 2016) and MStreamF (Yin et al., 2018), exploiting independent word representation in their cluster models tends to cause ambiguity. Let us show the following four tweets, for example:

T1: *"A regular intake of an Apple can improve your health and muscle stamina."*

T1: *"A glass of fresh apple juice is recommended for breakfast."*

T2: *"New Apple Watch can monitor your health."*

[*]Corresponding author: Junming Shao

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 766–776
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

T2: *"Apple will launch new smartphone iPhoneX this december."*

Tweets of these two topics share few common terms, i.e., '*health*' or '*apple*'. It creates an ambiguity if the model deals with only single term representation to calculate the similarity. However, the co-occurring terms representation (i.e., context) helps a model to identify the topic[1] correctly.

To solve these aforementioned issues, we propose an online semantic-enhanced dirichlet model for short text stream clustering. Compared to existing approaches, it has following advantages. (1) It allows processing each arriving short text in an online way. The online model is not only free of determining the optimal batch size, but also lends itself to handling large-scale data streams efficiently; (2) To the best of our knowledge, it is the first work to integrate semantic information for model-based online clustering, which is able to handle "term ambiguity" problem effectively and finally support high-quality clustering; (3) Equipped with Poly Urn Scheme, the number of clusters (topics) are determined automatically in our cluster model.

2 Related Work

During the past decade, many text stream clustering algorithms have been proposed. Here, due to the space limitation, we only report some model-based approaches which are highly related to our work. For more details, please refer to comprehensive surveys, e.g., (Mahdiraji, 2009; Silva et al., 2013; Nguyen et al., 2015; Aggarwal, 2018).

The early classical attempt for text clustering is Latent Dirichlet Allocation (LDA) (Blei et al., 2003). However, it cannot handle the temporal data for text streams. For this purpose, many LDA variants have been proposed to consider the text streams such as dynamic topic model (DTM) (Blei and Lafferty, 2006), dynamic mixture model (DMM) (Wei et al., 2007), temporal LDA (T-LDA) (Wang et al., 2012), streaming LDA (S-LDA) (Amoualian et al., 2016), and dirichlet mixture model with feature partition (DPMFP) (Zhao et al., 2016). These models assume that each document contains rich content, and thus they are not suitable for dealing with the short text streams. Later, Dirichlet multinomial mixture model-based dynamic clustering topic (DCT) model was designed to deal with short text streams by assigning each document with single topic (Liang et al., 2016). Very soon, GSDMM was proposed to extend DMM with collapsed gibbs sampling to infer the number of clusters (Yin and Wang, 2014). However, most of these models did not investigate the evolving topics (clusters) in text streams where the number of topics usually evolves over time.

To automatically detecting the number of clusters, (Ahmed and Xing, 2008) proposed a temporal dirichlet process mixture model (TDMP). It divides the text stream into many chunks (batches), and assumes that the documents inside each batch are interchangeable. Later, GSDPMM was proposed with collapsed gibbs sampling to infer the number of clusters in each batch. In contrast to LDA, GSDPMM not only converges faster but also dynamically assigns the number of clusters over time (Yin and Wang, 2016a). However, both TDMP and GSDPMM models do not examine the evolving topics, and, these models process the text stream for multiple times. Thereafter, MStreamF (Yin et al., 2018) was thus proposed by incorporating a forgetting mechanism to cope with cluster evolution, and allows processing each batch only one time. The NPMM model (Chen et al., 2019) was recently introduced by using the word-embeddings to eliminate a cluster generating parameter of the model.

In summary, for most existing approaches, they usually work in a batch way. However, determining optimal batch sizes for different text streams is usually a difficult task. More importantly, due to the intrinsic sparse data representation of short-text data, the semantics, is little investigated in established approaches. Actually, they need to be carefully considered to decrease the term ambiguity in short text clustering.

3 Preliminaries

Here, the problem statement is first given, followed with a brief introduction about dirichlet process and Poly Urn scheme.

3.1 Problem Formulation

Formally, a text stream is continuous arrival of text documents over time: $S_t = \{d_t\}_{t=1}^{\infty}$. Where d_t denotes a document arrived at time t. Each document contains specific words $d_t = \{w_1, w_2, \dots, w_n\}$ and may have different length. The key objective of the clustering task is to group similar documents into clusters: $Z = \{z_t\}_{t=1}^{\infty}$, and each clus-

[1] Topic and cluster will be interchangeably used in this paper

ter z_t contains documents represented as $z_t = \{d_1^{z_t}, d_2^{z_t}, \ldots, d_n^{z_t}\}$. For short text clustering, each document is the member of only one topic, so $z_i \cap z_j = \phi$, where $i \neq j$.

3.2 Dirichlet Process

Dirichlet Process (DP) is a non-parametric stochastic processes to model the data (Teh et al., 2006). It is the process to draw a sample from (base) distribution, where each sample itself is a distribution, denoted as $\mathcal{N} \sim \text{DP}(\alpha, \mathcal{N}_0)$. Here, $\mathcal{N}$ is the drawn sample from the base distribution $\mathcal{N}_0$. The drawing procedure of a sample from the distribution is controlled by a concentration parameter α.

3.3 Poly Urn Scheme (PUS)

The procedure to draw the sequential samples $\mathcal{N}_1, \mathcal{N}_2 \ldots$ from a distribution is described by the *poly urn scheme* (Blackwell et al., 1973). It can be summarized as:

$$\mathcal{N}_n | \mathcal{N}_{1:n-1} \sim \frac{\alpha}{\alpha + n - 1} + \frac{\sum_{k=1}^{n-1} \delta\left(\mathcal{N}_n - \mathcal{N}_k\right)}{\alpha + n - 1}$$

Here, $\delta(x) = 1$ if $x = 0$ and $\delta(x) = 0$ otherwise. Initially, the urn is empty, so we draw a color from the base distribution i.e. $\mathcal{N}_1 \sim \mathcal{N}_0$, and put a ball of drawn color into the urn. In the next turn, either we draw a color from the distribution which is already drawn with probability of $\frac{n-1}{\alpha+n-1}$, or draw a new color with probability of $\frac{\alpha \mathcal{N}_0}{\alpha+n-1}$. Since, drawing samples from distribution is repeated, so the same color may appear more than once. This defines that we have K number of distinct colors and n number of draws. This condition is defined by a well-known process called Chinese restaurant process (CRP) (Ferguson and Thomas S Ferguson, 1973). In CRP, we suppose that there are infinite number of tables in a restaurant, and each table surrounds infinite number of empty chairs. The first customer sits on first table, and later on the next customer either chooses to sit on any occupied table with probability of $\frac{n_k}{\alpha+n-1}$ or chooses an empty table with probability of $\frac{\alpha}{\alpha+n-1}$. Here, n_k is number of customers sitting on a specific table. A new customer is tend to be attracted towards a highly crowded table. This phenomenon is one part of our equation to understand creation of clusters over time. The CRP represents the draws from distribution $\mathcal{G}$, while the stick-breaking process shows the property of $\mathcal{G}$ explicitly:

$$\mathcal{G}(\mathcal{N}) = \sum_{k=1}^{\infty} \theta_k \delta\left(\mathcal{N} - \mathcal{N}_k\right), \quad \mathcal{N}_k \sim \mathcal{N}_0 \qquad (1)$$

The mixture weights $\theta = \{\theta_k\}_{k=1}^{\infty}$ can be formalized by $\theta \sim GEM(\gamma)$ (Neal, 2000). We exploit Equation (1) for the generative process of the Dirichlet process multinomial mixture model (DPMM) as follows.

$$z_d | \theta \sim \text{Mult}(\theta) \quad d = 1, \ldots, \infty$$

$$\mathcal{N}_k | \beta \sim \text{Dir}(\beta) \quad k = 1, \ldots, \infty$$

$$d \left| z_d, \{\mathcal{N}_k\}_{k=1}^{\infty} \sim p\left(d | \mathcal{N}_{z_d}\right)\right.$$

Here, z_d is the assigned documents to the cluster, which are multinomial distributed. The probability of document d generated by topic z is summarized as:

$$p\left(d | \mathcal{N}_z\right) = \prod_{w \in d} \text{Mult}\left(w | \mathcal{N}_z\right) \qquad (2)$$

Here, the naive Bayes assumption is considered where words in a document are independently generated by the topic. Whereas, the sequential draw of the sample can be derived by following the CRP. It is also assumed that the position of words in a document is not considered while calculating the probability.

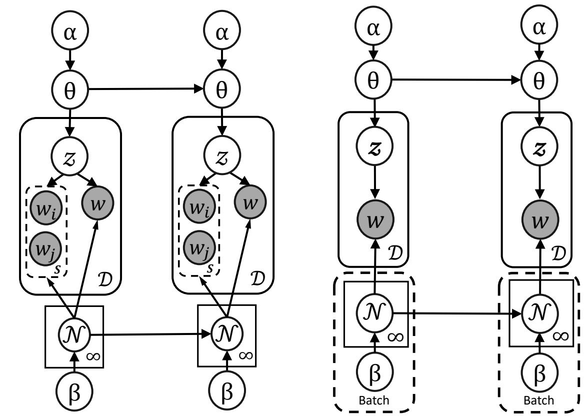
4 Proposed Approach

This section gives a brief discussion about the representation and formulation of the proposed algorithm.

4.1 Model Representation

We build our model upon the DPMM (Yin and Wang, 2016a), which is an extension of the DMM model to deal with evolving clusters. We call our model as OSDM (Online Semantic-enhanced Dirichlet Model), aiming at incorporating the semantic information and cluster evolution simultaneously for short text stream clustering in *an online way*. The graphical model of OSDM is given in Figure 1a.

We show two major differences in our model to highlight the novelty. First, for word-topic distribution, we embed semantic information by capturing the ratio of word co-occurrence. Thereby, independent word generating process and word co-occurrence weight are well considered in topic generation. Secondly, our model works instance



(a) OSDM (b) MStreamF

Figure 1: The graphical representation of OSDM and MStream. Here MStream works in a batch way while OSDM works in an online way.

by instance fashion to cluster the documents, instead of batch by batch. For comparison, Figure 1b further show the MStreamF (Yin et al., 2018) model. At initial stage before clustering documents of a batch, MStreamF update vocabulary set (active terms) from all the documents in a batch, then it starts the clustering each document of the batch. However, OSDM does not consider fixed number of documents to create vocabulary set, instead it incrementally updates with each arriving document.

4.2 Model Formulation

Defining the relationship between documents and clusters is the most crucial task while dealing with the text stream clustering problem. The threshold-based methodology (Nguyen et al., 2015) adapts similarity measures to define the homogeneity threshold between a cluster and a document. If the dissimilarity between the exiting clusters and a new arriving document is above the threshold, then a new cluster is created. However, due to the dynamic nature of the stream, it is very hard to define the similarity threshold manually.

In contrast, we assume that documents are generated by DPMM (see Section 3). Most recent algorithm MStreamF improved DPMM to cluster short text documents in the stream. As a further study, we integrate the semantic component in DPMM model. Additionally, we integrate term importance on the basis of cluster frequency. The derived equation for calculating the probability of a document d choosing existing cluster z is given in Equation (3).

$$p\left(z_d = z|\vec{z}, \vec{d}, \alpha, \beta\right) = \left(\frac{m_z}{D-1+\alpha D}\right) \cdot \left(\frac{\prod_{w \in d} \prod_{j=1}^{N_d^w} (n_z^w \cdot ICF_w) + \beta + j - 1}{\prod_{i=1}^{N_d} n_z + V\beta + i - 1}\right) \cdot \left(1 + \sum_{w_i \in d \wedge w_j \in d} cw_{ij}\right) \quad (3)$$

The first term of this Equation $\left(\frac{m_z}{D-1+\alpha D}\right)$ represents completeness of the cluster. Here, m_z is the number of documents contained by the cluster z and D is the number of current documents in active clusters[2]. Whereas, α is the concentration parameter of the model. The middle term of the equation based on multinomial distribution (see Equation (2)) with psuedo weight of words β defines the homogeneity between a cluster and a document. N_d and N_d^w represents total number of words and term frequency of word w in document d, respectively. The symbol n_z^w is the term frequency of the word w in the cluster z. The current vocabulary size of the model is represented by V. n_z is the number of words in the cluster z. ICF_w calculates the term importance over the active clusters in the model, which is defined as follows.

$$ICF(w \in d) = \log\left(\frac{|Z|}{|w \epsilon Z|}\right) \quad (4)$$

Here, $|Z|$ represents the number of active clusters in the model. The denominator part of Equation (4) is the number of those cluster which contains the word w. The term $\left(1 + \sum_{w_i \in d \wedge w_j \in d} cw_{ij}\right)$ defines the semantic weight of term co-occurrence between the cluster and a document. Formally, we define a value of an entry cw_{ij} in the co-occurrence matrix as follows.

$$cw_{ij} = \frac{\sum\limits_{d' \subseteq z} n_{d'}^{w_i}}{\sum\limits_{d' \subseteq z} n_{d'}^{w_i} + \sum\limits_{d' \subseteq z} n_{d'}^{w_j}} \quad (w_i, w_j) \in d' \quad (5)$$

Here, $n_z^{d'}$ is frequency count of word w_i in document d'. The ratio between w_i and w_j must satisfy the property $cw_{ij} + cw_{ji} = 1$. We calculate the term co-occurrence weight of those terms which are

[2] Active clusters refer to those clusters which are not yet deleted from the model.

common in the cluster z and document d. Term co-occurrence matrix is constructed where two terms are co-occurred in a single document. Therefore, if the size of cluster feature set (discussed in Section 4.3) is $|V_z|$, then it is not necessary that the co-occurrence matrix would be $|V_z| \times |V_z|$.

So far, we have defined the probability of a document choosing existing cluster, then we have to define the probability for a document to creating a new cluster. By following the DPMM for infinite number of clusters, which transform $\theta \sim GEM(\gamma)$ into $\theta \sim GEM(\alpha D)$, because the hyper-parameter for the mixture model should be dynamically change over time. Therefore, the probability of creating a new cluster is as follows.

$$p\left(z_d = z|\vec{z}_{\neg d}, \vec{d}, \alpha, \beta\right) = \left(\frac{\alpha D}{D-1+\alpha D}\right) \cdot \left(\frac{\prod_{w\in d}\prod_{j=1}^{N_d^w} \beta + j - 1}{\prod_{i=1}^{N_d} V\beta + i - 1}\right) \qquad (6)$$

Here, the pseudo number of clusters related documents in the model is represented as αD , and β is the pseudo term frequency of each word (exist in document) of the new cluster.

4.3 The cluster feature (CF) set

The similarity-based text clustering approaches usually follow vector space model (VSM) to represent the cluster feature space (Din and Shao, 2020). However, a topic needs to be represented as the subspace of global feature space. Here, we use a micro-cluster feature set to represent each cluster. Namely, a cluster is represented as the summary statistics of a set of words of related documents. In our model, a cluster feature (CF) set is defined as a 6-tuple $\{m_z, n_z^w, cw_z, len_z, l_z, u_z\}$, where m_z is the number of documents in the cluster z, n_z^w is the number of frequency of the word w in the cluster, cw_z is the word to word co-occurrence matrix, len_z is the number of words in the cluster z which is sum of all frequencies of words, l_z is the cluster weight, and u_z is the last updated time stamp.

The desirable addition property of cluster feature allows updating each micro-cluster in an online way.

Definition 1: A document d can be added to a cluster z by using the *addition property*.

$$m_z = m_z + 1$$
$$n_z^w = n_z^w + N_d^w \quad \forall w \in d$$
$$cw_z = cw_z \cup cw_d$$
$$len_z = len_z + len_d$$

Here, cw_d is word to word co-occurrence of the document, and len_d represents the number of total words in the document. The complexity of updating a cluster by adding a document is $O(\mathcal{L})$, where $\mathcal{L}$ is the average length of the document. This property is useful to update evolving micro-clusters in the text stream clustering procedure.

Algorithm 1: OSDM

```
Input: $S_t : \{d_t\}_{t=1}^{\infty}$ , $\alpha$ : concentration
       parameter, $\beta$ : pseudo weight of
       term in cluster, $\lambda$ : decay factor
Output: Cluster assignments $z_d$
$K = \phi$
while $d_t$ in $S_t$ do
    $t = t + 1$
    $K = removeOldZ_i(K)$
    $K = reduceClusterWeight(\lambda, K)$
    foreach $z_i \in K$ do
        $P_{Z_i} = prob(z_i, d_t)$ using Eq. (3)
    end
    $i = \arg\max_i(P_{Z_i})$
    $P_{Z_n}$ = calculate the probability of new
     cluster using Eq. (6)
    if $P_{Z_i} < P_{Z_n}$ then
        $m_{z_n} = 1$
        $n_{z_n}^w = N_{d_t}^w$
        $cw_{z_n} = cw_{d_t}$
        $len_{z_n} = len_{d_t}$
        $l_{z_n} = 1, u_{z_n} = t$
        $K = K \cup z_n$
    else
        $m_{z_i} = m_{z_i} + 1$
        $n_{z_i}^w = n_{z_i}^w + N_{d_t}^w$
        $cw_{z_i} = cw_{z_i} \cup cw_{d_t}$
        $len_{z_i} = len_{z_i} + len_{d_t}$
        $l_{z_i} = 1, u_{z_i} = t$
    end
end
```

4.4 OSDM Algorithm

We propose a semantic-enhanced non-parametric dirichlet model to cluster the short text streams in an online way, called OSDM. The proposed algorithm allows processing each instance incrementally and updates the model accordingly.

The procedure of OSDM is given in Algorithm 1. Initially, it creates a new cluster for the first document and the document is assigned to the newly created CF set. Afterward, each arriving document in the stream either choose an existing cluster or generate a new cluster. The corresponding probability for choosing either of an existing cluster or a new cluster is computed using Equation (6) and (3), respectively. The CF vector with the highest probability is updated using the addition property.

To deal with the cluster evolution (i.e., evolving topics) in text streams, many existing approaches often delete the old clusters by using some of the forgetting mechanisms (e.g., decay rate) (Zhong, 2005; Aggarwal and Yu, 2010; Islam et al., 2019). Instead of deleting old clusters, MStreamF (Yin et al., 2018) deletes old batches. In this study, we investigate the importance of each micro-cluster to handle the cluster evolution problem. Specifically, the importance of each micro-cluster is decreased over time if it is not updated. l_z in CF stores weight of each cluster. If the weight is approximately equals to zero, then the cluster is removed from the model, i.e., it cannot capture recent topics in the text stream. For this purpose, we applied the exponential decay function, $l_z = l_z \times 2^{-\lambda \times (\triangle t)}$. Here, $\triangle t$ is the elapsed time from the last update, and λ is the decay rate. The decay rate must be adjusted depending upon the applications at hand. The initial value of l_z (See Line 16 of Algorithm 1) is set to 1. Afterward, the importance of micro-cluster is exponentially decreases over time. We can also store the deleted clusters in a permanent disk for offline analysis.

Complexity Analysis. The OSDM algorithm always maintains the average $\bar{K}$ number of current topics (CF sets). Every CF set store average $\bar{V}$ number of words in n_z^w and at most $|\bar{V}_z| \times |\bar{V}_z|$ in cw_z. Thus the space complexity of OSDM is $O(\bar{K}(\bar{V} + \bar{V}^2) + VD)$, where V is the size of active vocabulary and D is the number of active documents. On other side, OSDM calculates the probability of arriving document with each cluster (see Line 6 of Algorithm 1). Therefore, the time complexity of OSDM is $O(\bar{K}(\mathcal{L}\bar{V}))$, where $\mathcal{L}$ is the average size of arriving document.

5 Experimental Study

5.1 Datasets and evaluation metrics

To evaluate the performance of the proposed algorithm, we conduct experiments on three real and two synthetic datasets. These datasets were also used in (Yin and Wang, 2016a; Liang et al., 2016; Qiang et al., 2018; Yin et al., 2018; Jia et al., 2018; Chen et al., 2019) to evaluate short text clustering models. In the preprocessing step, we removed stop words, converted all text into lowercase, and stemming. The description of the datasets is as follows.

- **News (Ns):** This dataset is collected by (Yin and Wang, 2014), which contains 11,109 news title belong to 152 topics.
- **Reuters (Rs):** Similar to (Yin and Wang, 2016b) we skip the documents with more than one class and obtained the dataset consists of 9,447 documents from 66 topics.
- **Tweets (Ts):** This dataset contain 30,322 tweets which are relevant to 269 topics in the TREC [3] microblog.
- **News-T (Ns-T) and Reuters-T (Rs-T):** Naturally, we may find a situation where topics in social media appear only for a certain time period and then disappear. However, the documents of each topic in original dataset is observed for long period of time. Therefore, to construct synthetic dataset we sorted documents datasets by topic in two datasets including *Reuters* and *News*. After sorting, we then divide each dataset into sixteen equal chunks and shuffled them.

We adopted five different evaluation metrics for deep analysis of all algorithms, which include Normalized Mutual Information (NMI), Homogeneity (Ho.), V-Measure (VM), Accuracy (Acc.) and cluster Purity (Pur.). We utilized sklearn[4] API to implement these metrics. We compute the measures on overall clustering results (Yin and Wang, 2014). Homogeneity measures that each cluster should have only members of a single class. Whereas, V-measure calculates how successfully the criteria of completeness and homogeneity are satisfied. Cluster purity measures the true positive instances in each cluster. The typical NMI measure calculates the overall clustering quality.

5.2 Baselines

We have selected four state-of-the-art representative algorithms for stream text clustering to com-

[3] http://trec.nist.gov/data/microblog.html

[4] http://scikit-learn.org

pare OSDM (Os). A brief description of these algorithms are given as follows.

(1) **DTM** (Blei and Lafferty, 2006) is an extension of Latent Dirichlet Allocation which traces the evolution of hidden topics from corpus over time. It was designed to deal with the sequential documents.

(2) **Sumblr (Sb)** (Shou et al., 2013) is an online stream clustering algorithm for tweets. With only one pass, it enables the model to cluster the tweets efficiently while maintaining cluster statistics.

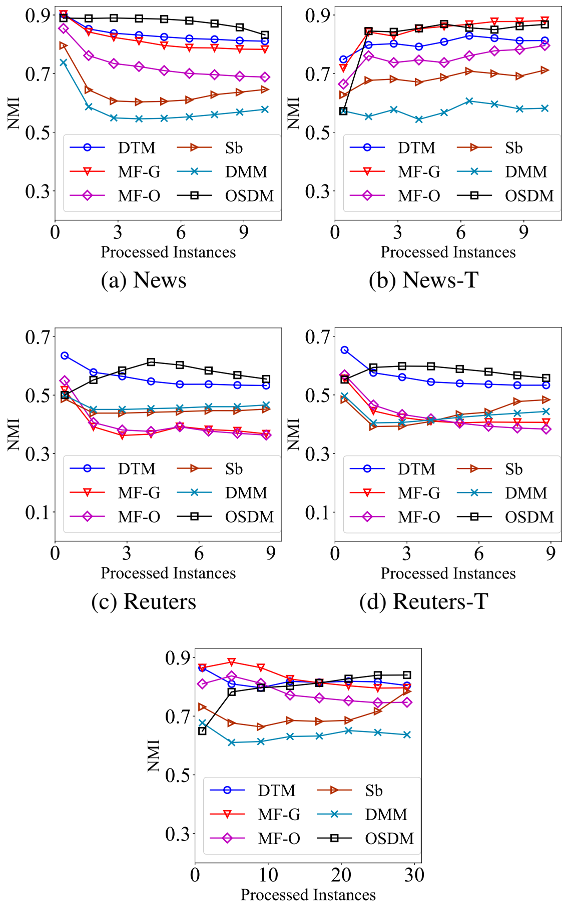
(3) **DMM** (Yin and Wang, 2014) is a Dirichlet multinomial mixture model for short text clustering, which does not consider temporal dependency of instances.

(4) **MStreamF** (Yin et al., 2018) is the latest model to deal with infinite number of latent topics in short text while processing one batch at a time. Two models of MStreamF were proposed, one with one-pass clustering process, and another with gibbs sampling. We refer to the former algorithm as MStreamF-O (MF-O) and the latter as MStreamF-G (MF-G).

We try to find the optimal parameter values of all baseline algorithms with grid search. Finally, we set $\alpha = 0.01$ for DTM, $\beta = 0.02$ for Sumblr. For MStreamF-O and MStreamF-G, we set $\alpha = 0.03$ and $\beta = 0.03$. As defined in (Yin et al., 2018), we set the number of iterations to 10 and $saved\text{-}batches = 2$ for MStreamF-G. We set $\alpha = 0.3$ and $\beta = 0.3$ for DMM. The DTM, DMM and Sumblr needs fixed number of cluster as input therefore we set $K = 300$, $K = 170$ and $K = 80$ for Tweets, News and Reuters datasets, respectively. We set $\alpha = 2e^{-3}$, $\beta = 4e^{-5}$ and $\lambda = 6e^{-6}$ for OSDM. The source code of OSDM is publicly available at: `https://github.com/JayKumarr/OSDM`.

5.3 Comparison with state-of-the-art methods

In this section, we provide a detailed comparative analysis of OSDM with state-of-the-art algorithms. The overall results are summarized in Table 1. We report NMI, Homogeneity, v-measure, purity and accuracy of each algorithm. Additionally, we also evaluate the performance of each algorithm over different time-stamps of the stream (see Figure 2).



(a) News (b) News-T

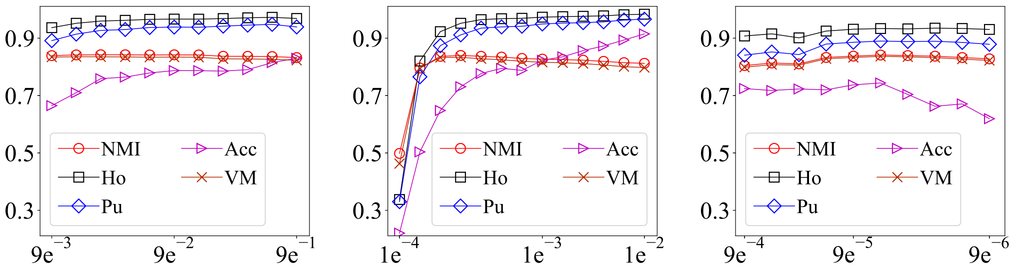
(c) Reuters (d) Reuters-T

(e) Tweets

Figure 2: The performance of different text steam clustering algorithm over time (in thousand points) in terms of NMI measure.

Further, we studied the parameter sensitivity and runtime of OSDM, respectively.

From Table 1, we can see that OSDM outperformed all baseline algorithms on almost every dataset in terms of all measures. Here, MStreamF-G yielded much better results on the Ns-T data in terms of NMI measure. The reason behind might be the multiple iterations of each batch in the stream. However, MStreamF-G requires more execution time to process the data. In contrast, our proposed algorithm OSDM processes the data only once. And we can also observe that OSDM achieves the highest NMI in other data sets. In addition, the crucial part of evaluating the cluster similarity is measured by the homogeneity measure. We can see that OSDM outperformed all previous algorithms. It also shows the same statistics except for v-measure of DTM. Likewise, our model generates more pure clusters. Furthermore, to investigate the performance over time, we plot the performance of



(a) α (b) β (c) λ

Figure 3: The sensitivity analysis with different parameters, including α, β and λ.

Alg.	Eva.	Datasets				
		Ns	Ts	Rs	Ns-T	Rs-T
OSDM	NMI	**0.815**	**0.836**	**0.552**	0.858	**0.554**
MF-O		0.685	0.746	0.361	0.803	0.381
MF-G		0.780	0.795	0.364	**0.888**	0.405
Sb		0.575	0.698	0.464	0.723	0.494
DTM		0.808	0.800	0.537	0.810	0.537
DMM		0.586	0.636	0.448	0.582	0.476
OSDM	Ho.	**0.951**	**0.936**	**0.954**	**0.900**	**0.964**
MF-O		0.654	0.695	0.374	0.778	0.385
MF-G		0.751	0.738	0.319	0.900	0.343
Sb		0.547	0.758	0.402	0.747	0.574
DTM		0.833	0.822	0.659	0.837	0.657
DMM		0.588	0.622	0.466	0.565	0.497
OSDM	VM	0.805	**0.831**	**0.479**	0.857	0.478
MF-O		0.684	0.744	0.361	0.803	0.380
MF-G		0.779	0.793	0.361	**0.888**	0.400
Sb		0.575	0.696	0.458	0.723	0.436
DTM		**0.808**	0.800	0.526	0.810	**0.527**
DMM		0.586	0.636	0.448	0.582	0.476
OSDM	Pur.	**0.907**	**0.890**	**0.962**	**0.851**	**0.972**
MF-O		0.552	0.529	0.602	0.636	0.608
MF-G		0.653	0.801	0.530	0.835	0.606
Sb		0.414	0.609	0.609	0.580	0.770
DTM		0.767	0.749	0.793	0.765	0.795
DMM		0.456	0.473	0.673	0.398	0.694
OSDM	Acc.	**0.880**	0.665	**0.927**	**0.769**	**0.952**
MF-O		0.420	0.246	0.577	0.584	0.447
MF-G		0.517	**0.707**	0.452	0.606	0.461
Sb		0.606	0.539	0.652	0.653	0.620
DTM		0.647	0.246	0.669	0.294	0.644
DMM		0.334	0.150	0.649	0.073	0.500

Table 1: The performance of different algorithms on five data sets in terms of different measures including Mutual Information (NMI), Homogeneity (Ho.), V-Measure (VM), Accuracy (Acc.) and cluster Purity (Pur.).

all algorithms over time in Figure 2.

5.4 Sensitivity Analysis

We perform sensitivity analysis for OSDM with respects to three input parameters: concentration parameter α, β, and decay function parameter λ on the *Tweets* dataset. From Figure 3a, we can observe the effect of α, which ranges from $9e^{-3}$ to $9e^{-1}$. The performance in terms of all evaluation measures is stable over the different values of parameters. The α parameter is responsible for finer clustering, that is why we can observe a little fluctuation in initial values. Figure 3b shows the performance on different values of β, which ranges from $1e^{-4}$ to $1e^{-2}$. As we already defined that we modified homogeneity part of the clustering model (see Equation (3)), and β is the related hyper-parameter. We can observe that after a certain range, the values of all the evaluation measure become stable. The crucial point to be observed is the stability of homogeneity on different values of β. Figure 3c shows effect of λ ranges from $9e^{-4}$ to $9e^{-6}$. Our model follows the forgetting mechanism on decay factor λ and the clusters are deleted from model when the value is approximately equals to zero. We can observe the performance of OSDM

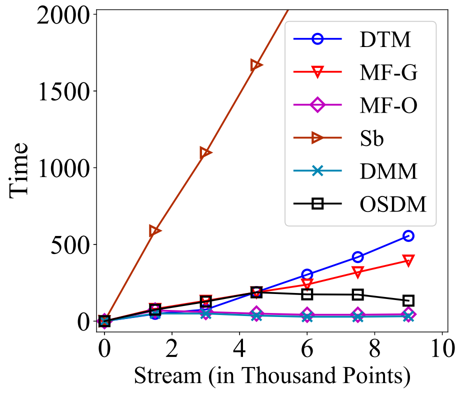


Figure 4: The runtime of different text stream clustering algorithms.

on different decay factors. It can be observed that the behavior of a given evaluation measure is stable over time.

5.5 Runtime

To compare the runtime of different algorithms, we performed all experiments on a PC with *core i5-3470* and 8GB memory. Figure 4 shows the runtime of all algorithms on the tweets dataset. We can observe that Sumblr required the highest execution time to cluster the instances. Whereas, the runtime of other algorithms are comparable. Due to simple execution process of each instance MStreamF-O took least time because it does not need to maintain semantic similarity. Comparatively, MStreamF-G required much higher time than OSDM. The reason is that it needs to execute each batch data multiple times. Due to online nature, the overall speed of OSDM is more efficient than most existing algorithms, and the benefit is strengthened with more and more arriving instances.

6 Conclusion

In this paper, we propose a new online semantic-enhanced dirichlet model for short text stream clustering. In contrast to existing approaches, OSDM does not require to specify the batch size and the dynamic number evolving clusters. It dynamically assigns each arriving document into an existing cluster or generating a new cluster based on the poly urn scheme. More importantly, OSDM tried to incorporate semantic information in the proposed graphical representation model to remove the term ambiguity problem in short-text clustering. Building upon the semantic embedding and online learning, our method allows finding high-quality evolving clusters. Extensive results further demonstrate that OSDM has better performance compared to many state-of-the-art algorithms.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61976044), Fundamental Research Funds for the Central Universities (ZYGX2019Z014), Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China (161062), National key research and development program (2016YFB0502300).

References

Charu C. Aggarwal. 2018. A survey of stream clustering algorithms. In *Data Clustering*, pages 231–258.

Charu C Aggarwal, Jiawei Han, Jianyong Wang, Philip S. Yu, Jianyong Wang Jiawei Han, Philip S. Yu, Jiawei Han, Jianyong Wang, and Philip S. Yu. 2003. A Framework for Clustering Evolving Data Streams. In *International conference on Very large data bases*, pages 81–92.

Charu C Aggarwal and Philip S. Yu. 2010. On Clustering Massive Text and Categorical Data Streams. *Knowledge and Information Systems*, 24(2):171–196.

Amr Ahmed and Eric P Xing. 2008. Dynamic Non-Parametric Mixture Models and The Recurrent Chinese Restaurant Process: with Applications to Evolutionary Clustering. In *Proceedings of SIAM International Conference on Data Mining*, pages 219–230.

Hesam Amoualian, Marianne Clausel, Eric Gaussier, Massih-Reza Amini, Marianne Clausel, Massih-Reza Amini, Éric Gaussier, and Massih-Reza Amini. 2016. Streaming-LDA: A Copula-based Approach to Modeling Topic Dependencies in Document Streams. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 695–704. ACM.

David Blackwell, James B MacQueen, Others, and David R. Brillinger. 1973. Ferguson distributions via Pólya urn schemes. *The annals of statistics*, 1(2):353–355.

David M. Blei and John D. Lafferty. 2006. Dynamic topic models. *ACM International Conference Proceeding Series*, 148:113–120.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022.

Junyang Chen, Zhiguo Gong, and Weiwen Liu. 2019. A nonparametric model for online topic discovery with word embeddings. *Information Sciences*, 504:32–47.

Salah Ud Din and Junming Shao. 2020. Exploiting evolving micro-clusters for data stream classification with emerging class detection. *Information Sciences*, 507:404–420.

Salah Ud Din, Junming Shao, Jay Kumar, Waqar Ali, Jiaming Liu, and Yu Ye. 2020. Online Reliable Semi-supervised Learning on Evolving Data Streams. *Information Sciences*, 507.

Thomas S Ferguson and Thomas S Ferguson. 1973. A Bayesian analysis of some nonparametric problems. *The annals of statistics*, 1(2):209–230.

Hongyu Gong, Tarek Sakakini, Suma Bhat, and Jinjun Xiong. 2018. Document similarity for texts of varying lengths via hidden topics. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 2341–2351.

Amir Hadifar, Lucas Sterckx, Thomas Demeester, and Chris Develder. 2019. A Self-Training Approach for Short Text Clustering. In *Proceedings of the 4th Workshop on Representation Learning for NLP (ACL)*, pages 194–199.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *Association for Computational Linguistics*, 1:328–339.

Ruizhang Rui-zhang Huang, Guan Yu, Zhaojun Wang, Jun Zhang, and Liangxing Shi. 2013. Dirichlet process mixture model for document clustering with feature partition. *IEEE Transactions on Knowledge and Data Engineering*, 25(8):1748–1759.

Md. Kamrul Islam, Md. Manjur Ahmed, and Kamal Z. Zamli. 2019. A buffer-based online clustering for evolving data stream. *Information Sciences*, 489:113–135.

Caiyan Jia, Matthew B. Carson, Xiaoyang Wang, and Jian Yu. 2018. Concept decompositions for short text clustering by identifying word communities. *Pattern Recognition*, 76:691–703.

Shangsong Liang, Emine Yilmaz, and Evangelos Kanoulas. 2016. Dynamic Clustering of Streaming Short Documents. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 995–1004. ACM.

Alireza Rezaei Mahdiraji. 2009. Clustering data stream: A survey of algorithms. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 13(2):39–44.

Radford M Neal. 2000. Markov chain sampling methods for Dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265.

Hai Long Nguyen, Yew Kwong Woon, and Wee Keong Ng. 2015. A survey on data stream clustering and classification. *Knowledge and Information Systems*, 45(3):535–569.

Jipeng Qiang, Yun Li, Yunhao Yuan, and Xindong Wu. 2018. Short text clustering based on Pitman-Yor process mixture model. *Applied Intelligence*, 48(7):1802–1812.

Junming Shao, Yue Tan, Lianli Gao, Qinli Yang, Claudia Plant, and Ira Assent. 2019. Synchronization-based clustering on evolving data stream. *Inf. Sci.*, 501:573–587.

Lidan Shou, Zhenhua Wang, Ke Chen, and Gang Chen. 2013. Sumblr Continuous Summarization of Evolving Tweet Streams. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 533–542. ACM.

Jonathan A. Silva, Elaine R. Faria, Rodrigo C. Barros, Eduardo R. Hruschka, André C. P. L. F. de Carvalho, and João Gama. 2013. Data stream clustering: A Survey. *ACM Computing Surveys*, 46(1):1–31.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Yu Wang, Eugene Agichtein, and Michele Benzi. 2012. TM-LDA: Efficient Online Modeling of Latent Topic Transitions in Social Media. In *International conference on Knowledge discovery and data mining*, pages 123–131. ACM.

Xing Wei, Jimeng Sun, and Xuerui Wang. 2007. Dynamic mixture models for multiple time series. In *International Joint Conference on Artificial Intelligence*, pages 2909–2914.

Jianhua Yin, Daren Chao, Zhongkun Liu, Wei Zhang, Xiaohui Yu, and Jianyong Wang. 2018. Model-based Clustering of Short Text Streams. In *ACM International Conference on Knowledge Discovery and Data Mining*, pages 2634–2642.

Jianhua Yin and Jianyong Wang. 2014. A dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 233–242. ACM.

Jianhua Yin and Jianyong Wang. 2016a. A model-based approach for text clustering with outlier detection. In *IEEE International Conference on Data Engineering*, pages 625–636.

Jianhua Yin and Jianyong Wang. 2016b. A Text Clustering Algorithm Using an Online Clustering Scheme for Initialization. In *ACM International Conference on Knowledge Discovery and Data Mining*, pages 1995–2004.

Zhong Zhang, Chongming Gao, Chongzhi Liu, Qinli Yang, and Junming Shao. 2019. Towards robust arbitrarily oriented subspace clustering. In *Database Systems for Advanced Applications - 24th International Conference, DASFAA 2019, Chiang Mai, Thailand, April 22-25, 2019, Proceedings, Part I*, volume 11446 of *Lecture Notes in Computer Science*, pages 276–291. Springer.

Yukun Zhao, Shangsong Liang, Zhaochun Ren, Jun Ma, Emine Yilmaz, and Maarten de Rijke. 2016. Explainable User Clustering in Short Text Streams. In *International ACM conference on Research and Development in Information Retrieval*, pages 155–164.

Shi Zhong. 2005. Efficient streaming text clustering. *Neural Networks*, 18(5-6):790–798.