A Soft Subspace Clustering Method for Text Data Using a Probability based Feature Weighting Scheme

Abdul Wahid, Xiaoying Gao, and Peter Andreae

School of Engineering and Computer Science,
Victoria University of Wellington,
19 Kelburn Parade 6012. Wellington, New Zealand
{abdul.wahid,xgao,pondy}@ecs.vuw.ac.nz
http://ecs.victoria.ac.nz

Abstract. Clustering methods aim to find clusters or groups of similar objects in a given set of data. Common soft subspace clustering methods for text data find different clusters in subspaces using a weighted distance measure. The weighting scheme heavily affects the clustering performance and requires special consideration. Since text data has semantic information along with syntactic information, a weighting scheme, which uses semantic information, is more likely to generate a better clustering solution.
This paper introduces a novel soft subspace clustering method that uses a probabilistic model to extract semantic information from documents for weighting features. We created a feature weight matrix from the probability distribution of terms in subspaces and developed a weighted distance measure for finding similar documents in relevant subspaces. Our experiment results on synthetic and real-world datasets show that our newly developed method outperforms other state-of-the-art soft subspace clustering methods.

Keywords: Clustering Algorithms, Soft Subspace Clustering, Latent Dirichlet Allocation

1 Introduction

Clustering methods try to find similar documents and group them together in clusters. Documents are generally represented in a Vector Space Model, where each distinct term is treated as a feature. Hence the feature space becomes very large. Traditional clustering methods such as k-means, consider all features at the same time to cluster the data and are only suitable for data with a small number of features.

Subspace clustering methods are widely applied when the number of features is very large. They try to group similar objects using a subset of features (i.e. subspace) instead of all features. In subspace clustering, each cluster represents a set of objects clustered according to a subspace of features. The problem of subspace clustering is often divided into two sub-problems: determining the subspaces and clustering the data. Based on how these problems are addressed, there are two main categories of subspace clustering methods: *hard subspace clustering* and *soft subspace clustering*. In hard subspace clustering, a feature in a subspace is either present or not present (1 or 0), whereas in soft subspace clustering, a feature in a subspace is determined by its degree of presence (i.e. a weight between 0-1). A feature is considered relevant (i.e. present) if its weight is high in a subspace and considered irrelevant if its weight is low in a subspace.

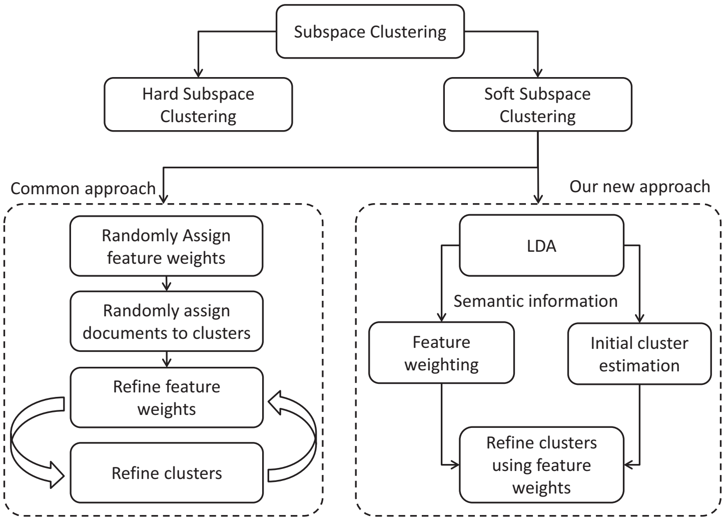

Fig. 1: Differences between of subspace clustering approaches and our new approach

In text datasets, some features can be considered to be partially presented in subspaces. Therefore, soft subspace clustering methods, which assign weights to features instead of determining the exact presence of features in a subspace, are becoming more popular in text clustering.

The most popular soft subspace clustering methods are FWKM [20], EWKM [19] and FGKM [9]. These methods use modified version of k-means to cluster the data in different subspaces according to feature weights. These methods mainly differ in terms of how they compute the feature weights. The main issue with these methods is that they ignore the semantic information of the documents, which might be helpful in improving the clustering process.

Latent Dirichlet Allocation (LDA) is a popular topic modeling method which can be used to extract semantic information from a collection of documents. LDA is based on a generative model, where a document is assumed to be generated from the distribution of terms which form a special theme or topic. The main idea of our method is to treat topics generated from the LDA model as subspaces because each topic specifies a soft subset of related terms (features). Subspaces generated by the LDA were utilized in initializing the clusters in our method.

We use LDA model to compute a probability that a term is relevant in a subspace (topic/subset of terms). These probabilities can represent the semantic information and is used as term or feature weightings in our soft subspace clustering to improve the clustering process. Figure 1 shows the difference between existing clustering methods and our new method. The common existing soft subspace clustering methods use a random approach to initialize weightings and randomly assign objects to clusters. Then the feature weightings and clusters are refined iteratively. In our method, we first use LDA to assign the feature weights and assign objects to the initial clusters. Then we iteratively refine the clusters according to the feature weights.

The main contribution of this paper is a new soft subspace clustering algorithm for documents using semantically weighted terms for different subspaces that are derived from the LDA model. The main novelty of the method is the development of a new weighted distance measure from the LDA probability matrices to compute the distances between the documents in different subspaces.

The paper is organized as follows: section 2 discusses the related work; section 3 describes our proposed method; section 4 explains the experimental design and section 5 presents results along with discussion; and section 6 provides a conclusion of the paper along with the future directions.

2 Related Work

2.1 Hard Subspace Clustering

Hard subspace clustering methods divide the feature space into different subspaces where each feature is either present or absent in a subspace. Hard subspace clustering methods can be further categorized by their search approaches i.e. bottom-up and top-down. The examples of bottom-up hard subspace clustering methods are CLIQUE [3], ENCLUS [10], MAFIA [18] and FINDIT [29]. The examples of top-down hard subspace clustering methods are PROCLUS [1], ORCLUS [2] and δ-Clusters [30]. Our method differs from these methods because it belongs to soft subspace clustering methods.

2.2 Soft Subspace Clustering

In soft subspace clustering, each feature is assigned different weights for different subspaces. Hence some proportion of a feature is present in all subspaces. In clustering process, the features that have higher weight values in a subspace contribute more to form a cluster than the features that have lower weights. Generally the soft subspace clustering methods employ variable weighting scheme and iteratively update the feature weights in the clustering process.

Variable weighting schemes are widely applied in data mining [13, 22, 11, 12, 21]. Some of the variable weighting methods can be extended, especially k-means type variable weighting, to develop soft subspace clustering algorithms [16, 7, 20, 17, 14, 15].

Recent approaches such as FWKM [20], EWKM [19] and FGKM [9, 8] use k-means type variable weighting algorithms and formulate a minimization problem for data clustering. FWKM uses Lagrange multiplier and forms a polynomial weighting formula to compute the feature weights and iteratively refines the clusters using the following objective function.

$$\min J(U, W, \Lambda) = \sum_{i=1}^{k} \sum_{j=1}^{n} u_{ij} \sum_{t=1}^{m} \lambda_{it} [(\mu_{it} - d_{jt})^2 + \sigma] \tag{1}$$

where

- u is a $k \times n$ binary matrix representing the assignment of objects to clusters. $u_{ij} = 1$ iff object j is in cluster i, $u_{ij} = 0$ otherwise.
- λ is $k \times m$ feature weight matrix. It represents k subspaces in rows and m features in columns. The value in a cell is a weight of the feature to its corresponding subspace and the value ranges from 0-1. The sum of the weights of all features in a subspace is 1. i.e. $\sum_{t=1}^{m} \lambda_{it} = 1, 1 \leq i \leq k, 0 < \lambda_{it} < 1$
- μ is a $k \times m$ matrix representing the mean value of a feature in a cluster.
- d_{jt} represents a feature t of the j^{th} object[1].
- σ is an average spread/variance of all the features in a dataset.

EWKM clusters the data in a similar fashion but uses the exponential weighting formula to compute the feature weights. Its objective function is similar to equation 1, but instead of using σ, it uses Shanon entropy to control the weights. FGKM has a slightly different approach, it not only uses the individual feature weightings but also uses the feature group weightings scheme. The feature group weightings is computed by combining features into different groups and then assigning weights to those groups.

The above soft subspace clustering methods ignore the semantic information of the documents in a clustering process. The main motivation of our research work is to investigate the use of semantic information (e.g. topics) of documents in soft subspace clustering process.

2.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [6] extracts topics/themes from documents, which have semantic information. It is widely used in other domains such as topic modeling [5] and Entity Resolution [4]. The topics generated by LDA can be considered as subspaces and for each subspace, LDA facilitates to compute a term weight. Our soft subspace clustering method is related to FWKM and

[1] for clustering a collection of documents, d_{jt} is often the term-frequency of a term in a document

EWKM, however our method uses LDA based weighting scheme to utilize the semantic information of the documents.

LDA is a probabilistic model with an assumption that a document is a random mixture over latent topics and each topic is a distribution over terms. The two main parameters in this model are topic-document distributions θ and topic-term distributions ϕ.

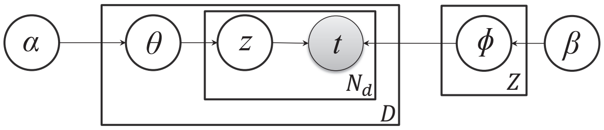

Fig. 2: A common LDA graphical model using plate notation.

Figure 2[1] represents a graphical model for LDA. Arrows represent conditional dependencies between two variables and plates/rectangles represent loop or repetition of the variable mentioned in the corner of the plate. The shaded circle represents the observed variable while unshaded represent unobserved variables. Hyperparameter α is a prior on topic distribution. High value of α favors topic distributions with more topics and low value (<1) of α favors topic distribution with a few topics. Hyperparameter β is a prior on term distribution in every topic, which controls the number of times terms are sampled from a topic. The LDA model infers three latent variables θ, ϕ and z (topics) while observing t (terms) in a document set D.

In Figure 2, the inner plate (z and t) denotes the continuous sampling of topics and terms until N_d terms are created from document d. The out plate (which is surrounding θ) denotes the continuous sampling of a topic distribution for each document d in a document set D. The plate surrounding ϕ denotes the continuous sampling of a term distribution over each topic z until a total of Z topics are generated. More details of LDA can be found in [5].

To the best of our knowledge, our research work is the first attempt that applies LDA to assign weights and use it in text soft subspace clustering.

3 Our LDA Weighted K-Means Model

This section presents our new subspace clustering method which builds on LDA for document clustering[2]. Figure 3 shows the overall design of our method. The documents are pre-processed by implementing stop words filtration, low frequency words filtration and WordNet lemmatization. Then we use LDA based

[1] This figure is created by the author. However, similar figures are commonly used in literature to describe LDA.

[2] The code of our method was implemented using lingpipe toolkit (http://alias-i.com/lingpipe/)

on Gibbs sampling to generate two matrices: topic-document matrix θ and topic-term matrix ϕ. θ is then used for initializing the clusters and ϕ is used as feature weights for refining the clusters.

3.1 Gibbs Sampling

We implemented LDA model in an unsupervised way (without using training datasets) using Gibbs sampling algorithm explained in [24]. The Gibbs sampling iteratively computes the conditional probability of assigning an occurrence of a term (token of a term) to each topic. The common Gibbs sampling method provides the estimates of the posterior distribution over z (topics) but does not provides θ and ϕ. However, we can use the Gibbs sampling technique to approximate θ and ϕ from posterior estimates of z.

For each token i (an occurrence of a term), let v_i, d_i, z_i denote the term for the token, the document for the token and the topic of the token respectively in a document collection. The Gibbs sampling iteratively processes each term token in the document collection and estimates the conditional probability of assigning the current term token to an individual topic, based on the topic assignments to all other term tokens. The conditional distribution is formalized as:

$$Prb(z_i = r|\mathbf{z}_{-i}, ...) \tag{2}$$

where $z_i = r$ is the assignment of i^{th} token to topic r. $\mathbf{z}_{-i}$ denotes the topic assignment of all the tokens excluding the i^{th} token. Other variables for equation 2 represented by (...) are v_i, d_i, $\mathbf{v}_{-i}$, $\mathbf{d}_{-i}$, α and β. $\mathbf{v}_{-i}$ represents all terms tokens except the i^{th} term token and $\mathbf{d}_{-i}$ represents document tokens except the i^{th} document token. Griffiths and Steyvers [24] provided a simple way to compute Equation 2 as:

$$Prb(z_i = r|\mathbf{z}_{-i}, ...) \propto \frac{\mathcal{C}^{(1)}_{rv_i} + \beta}{\sum_{l=1}^{m} \mathcal{C}^{(1)}_{rl} + m\beta} \frac{\mathcal{C}^{(2)}_{rd_i} + \alpha}{\sum_{z=1}^{Z} \mathcal{C}^{(2)}_{zd_i} + Z\alpha} \tag{3}$$

where $\mathcal{C}^{(1)}$ and $\mathcal{C}^{(2)}$ are $Z \times m$ and $Z \times D$ matrices respectively and Z, m, D are the number of topics, terms and documents respectively. The cell values of these matrices represent the frequency of the term/document for the corresponding topics. $\mathcal{C}^{(1)}_{rv_i}$ denotes the number of times the term v_i is assigned to the topic r excluding the i^{th} instance and $\mathcal{C}^{(2)}_{rd_i}$ denotes the number of times a term token in document d is assigned to the topic r excluding the i^{th} instance.

3.2 Generating θ and ϕ

After applying the Gibbs sampling algorithm, we create two matrices: 1) ϕ topic-term matrix and 1) θ topic-document matrix. These matrices are generated from the two count matrices $\mathcal{C}^{(1)}$ and $\mathcal{C}^{(2)}$ according to [24] as follows:

$$\phi_{rt} = \frac{\mathcal{C}^{(1)}_{rt} + \beta}{\sum_{l=1}^{m} \mathcal{C}^{(1)}_{rl} + m\beta}, \ \theta_{rj} = \frac{\mathcal{C}^{(2)}_{rj} + \alpha}{\sum_{z=1}^{Z} \mathcal{C}^{(2)}_{zj} + Z\alpha} \tag{4}$$

ϕ corresponds to the probability that a term t is assigned to topic r and θ corresponds to the probability that a document j is assigned to topic r.

The rows of topic-document matrix θ represent topics and the columns represent documents. The cells of θ represent the probability that a document has the corresponding topic. We use this matrix to form the initial clusters. One should note that LDA naturally provides a simple way for clustering the documents. However, this clustering is not soft subspace clustering. Following is a way to improve the clusters generated from LDA by utilizing the information from LDA and forming soft subspace clustering method.

In LDA model, each term is a feature and each topic corresponds to a subspace, therefore topic-term matrix ϕ can be considered of a feature weight matrix for different subspaces where each feature or term has a degree of presence in all subspaces or topics. We used the values of topic-term matrix ϕ for determining relevant subspaces and developed a new weighted distance measure, which finds similar documents in relevant subspaces.

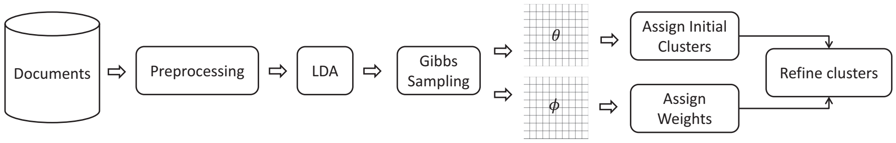

Fig. 3: System diagram of our new method. θ and ϕ are the topic-document and topic-term matrices respectively.

3.3 Objective Function

We perform clustering by formulating the clustering as a minimization problem and our objective is to minimize the sum of squared distances between documents and the nearest cluster centers weighted by different subspaces. The objective function is similar to the objective functions (equation 1) of the FWKM or EWKM, however, we do not include σ or Shanon entropy because we are already controlling the feature weighting using two hyper parameters of LDA model (α and β). Moreover, the objective function uses previously computed LDA based feature weights instead of computing the feature weights in iterative manner.

Let $D = \{d_1, d_2, d_3, ..., d_n\}$ be a set of n documents and $T = \{t_1, t_2, t_3, ..., t_m\}$ represents m terms in the documents. Then the objective function for clustering the n documents into k clusters can be defined as:

$$\sum_{i=1}^{k}\left(\sum_{j=1}^{n}\sum_{t=1}^{m}\delta_{ij}\phi_{it}(\mu_{it}-d_{jt})^2\right) \tag{5}$$

where

- δ is a $k \times n$ binary matrix representing the assignment of documents to clusters. $\delta_{ij} = 1$ iff document j is in cluster i, $\delta_{ij} = 0$ otherwise.
- ϕ is $k \times m$ topic-term matrix generated from LDA model. It represents k subspaces in rows and m terms in columns. The value in a cell is a weight of the term to its corresponding subspace and the value ranges from 0-1. The sum of the weights of all terms in a subspace is 1. i.e. $\sum_{t=1}^{m}\phi_{it} = 1, 1 \leq i \leq k, 0 < \phi_{it} < 1$
- μ is a $k \times m$ matrix representing the mean value of a term in a cluster. It is calculated as:
$$\mu_{it} = \frac{\sum_{j=1}^{n}\delta_{ij}d_{jt}}{\sum_{j=1}^{n}\delta_{ij}} \tag{6}$$
- d_{jt} represents a term t (a feature) of the j^{th} document, which is the term-frequency of the term in the document.

We iteratively assign documents to their nearest cluster centers until the algorithm converges. We minimize the objective function by updating δ using the following:

$$\delta = \begin{cases} \delta_{ij} = 1, & \text{if } i = \text{argmin}_x\, dist(\mu_x, d_j) \\ \delta_{ij} = 0, & \text{otherwise} \end{cases} \tag{7}$$

where $dist(\mu_x, d_j)$ is defined as

$$dist(\mu_x, d_j) = \sum_{t=1}^{m}\phi_{xt}(\mu_{xt}-d_{jt})^2 \tag{8}$$

Equation 8 defines our distance measure. Unlike k-means, our distance measure computes the distance of a document from the cluster centers by using a LDA parameter ϕ, which provides a semantic based feature weighting to different subspaces. Higher value of the probability that a term is assigned to a topic indicates that the term has a higher degree of presence in a subspace. Therefore the difference between a term in the document and the mean value of the term in the cluster for that particular term is more important. The use of LDA differentiates our method from other soft subspace clustering methods.

3.4 Our Algorithm: DWKM

Our Dirichlet Weighted K-mean algorithm is a modified version of k-means algorithm. The details are shown in Algorithm 1.

Algorithm 1 DWKM

Input: document set D and number of clusters k
Output: Clustering solution $\mathcal{C}$

```
1: Preprocess document set $D$
2: Initialize the LDA model and assign all term tokens to $Z$ Topics according to equation 2 and 3
3: Perform Gibbs sampling and generate $\theta$ and $\phi$ from LDA model using equation 4
4: Initialize $\delta$ using $\theta$. $\delta_{ij} = 1, \text{ if } i = \text{argmax}_x \, \theta_x$
5: repeat
6:     Update clusters means according to equation 6
7:     Assign documents to $\delta$ according to equation 7
8: until Convergence
```

Algorithm 1 takes two arguments: a document set and the number of clusters and outputs the clustering solution. The algorithm performs preprocessing step on the documents, which includes stop word removal, lemmatization and tokenization of words. Then the algorithm randomly assigns all term tokens to Z topics and performs Gibbs sampling. Once ϕ and θ matrices are generated, line 4 of the algorithm groups documents to different clusters according to their highest probability using θ. The algorithm then, fine tunes the clusters by repeating the update and assignment steps according to equation 6 and 7 until convergence criteria is met. The convergence criterion terminates the loop if there are no more documents to relocate to any clusters or the total number of specified iterations exceeds the predefined limit.

4 Experimental Setup

Our experiments are designed based on two recent papers [9] and [19]. Our method DWKM was evaluated on four synthetic and six real world datasets, and compared with five clustering methods using different cluster quality measures. Four synthetic datasets were generated by following the same process described in [9] and six real-world datasets were generated as described in [19].

4.1 Datasets

The synthetic datasets SD1, SD2, SD3, SD4 were generated according to [9]. Each consists of 6000 objects, 200 features, three subspaces and three clusters. The noise level in SD1, SD2, SD3 and SD4 are 0, 0.2, 0 and 0.2 respectively (as described in [9]). The percentage of missing values in DS1, DS2, DS3 and DS4 are 0, 0, 0.12, 0.12 respectively. Detailed information about how to reproduce the synthetic datasets can be found in [9].

The six real-word datasets with two or more clusters from 20-Newsgroup[1] are the same as [19]. Table 1 shows the details of these six datasets. The dataset D1,

[1] http://qwone.com/ jason/20Newsgroups/

D2 and D3 are easier than datasets D4, D5 and D6. D1 and D2 have semantically different clusters whereas D4 and D5 have semantically related clusters. D3 and D6 have unbalanced clusters (as shown in table 1).

Table 1: Six real world datasets created from 20-Newsgroup dataset

dataset	clusters	# of docs	dataset	clusters	# of docs
D1	alt.atheism	100	D4	talk.politics.mideast	100
	comp.graphics	100		talk.politics.misc	100
D2	comp.graphics	100	D5	comp.graphics	100
	rec.sport.baseball	100		comp.os.ms-windows	100
	sci.space	100		rec.autos	100
	talk.politics.mideast	100		sci.electronics	100
D3	comp.graphics	120	D6	comp.graphics	120
	rec.sport.baseball	100		comp.os.ms-windows	100
	sci.space	59		rec.autos	59
	talk.politics.mideast	20		sci.electronics	20

4.2 Evaluation Measures

In order to compare our method with other methods, we used two evaluation measures: Cluster Accuracy [23] and F-measure [19, 25–27] for synthetic dataset and three evaluation measures: F-measure, Normal Mutual Information(NMI) [32] and Entropy [31] for the real-world datasets. These measures are chosen based on [19] and [9] The lower entropy value of a clustering solution indicates the clustering solution has a better quality, whereas higher values of all other evaluation measures indicate a better cluster quality.

The evaluation measures can be computed as follows:

$$ClusterAccuracy = \frac{\sum_{i=1}^{k} d_i}{n} \tag{9}$$

$$\text{F-measure} = \sum_{i=1}^{k} \frac{n_i}{n} \cdot \max_{1 \le j \le k} \left\{ \frac{2 \cdot \frac{n_{ij}}{n_i} \cdot \frac{n_{ij}}{n_j}}{\frac{n_{ij}}{n_i} + \frac{n_{ij}}{n_j}} \right\} \tag{10}$$

$$\text{NMI} = \frac{\sum_{i=1,j=1}^{k} n_{ij} \log\left(\frac{n \cdot n_{ij}}{n_i \cdot n_j}\right)}{\sqrt{(\sum_{i=1}^{k} n_i \log \frac{n_i}{n})(\sum_{j=1}^{k} n_j \log \frac{n_j}{n})}} \tag{11}$$

$$\text{Entropy} = \sum_{j=1}^{k} \frac{n_j}{n} \left(-\frac{1}{\log k} \sum_{i=1}^{k} \frac{n_{ij}}{n_j} \cdot \log \frac{n_{ij}}{n_j} \right) \tag{12}$$

where d_i is correctly identified documents in cluster i, k is total number of clusters and n is the total number of documents in a dataset. n_i and n_j represent the number of documents in class i of the original dataset and cluster j in our computed clustering solution respectively, n_{ij} represents the number of documents that are common in both class i and cluster j.

Table 2: Comparison of clustering methods on synthetic dataset using Accuracy (AC) and F-measure (FM). The values on left are the mean values of 100 runs and the values in parenthesis are standard deviation of 100 runs.

datasets	Metric	k-means	LDA	FWKM	EWKM	FGKM	DWKM
SD1	AC	0.65 (0.09)	0.66 (0.11)	0.77 (0.14)	0.69 (0.10)	0.82 (0.16)	**0.87 (0.15)**
	FM	0.63 (0.13)	0.65 (0.09)	0.73 (0.19)	0.59 (0.13)	0.75 (0.22)	**0.81 (0.20)**
SD2	AC	0.63 (0.04)	0.68 (0.06)	0.76 (0.10)	0.72 (0.13)	0.87 (0.16)	**0.92 (0.15)**
	FM	0.64 (0.05)	0.69 (0.09)	0.75 (0.12)	0.63 (0.17)	0.82 (0.22)	**0.88 (0.21)**
SD3	AC	0.62 (0.04)	0.64 (0.07)	0.67 (0.07)	0.70 (0.09)	**0.94 (0.13)**	**0.94 (0.12)**
	FM	0.62 (0.06)	0.63 (0.13)	0.64 (0.11)	0.59 (0.11)	0.91 (0.18)	**0.92 (0.17)**
SD4	AC	0.60 (0.04)	0.61 (0.15)	0.61 (0.06)	0.69 (0.08)	0.91 (0.13)	**0.93 (0.13)**
	FM	0.59 (0.05)	0.60 (0.16)	0.60 (0.07)	0.58 (0.11)	0.88 (0.18)	**0.90 (0.19)**

5 Results

We compared our method DWKM with k-means, LDA based simple clustering, FWKM [20], EWKM [19] and FGKM [9].

5.1 Comparison

K-means and LDA based simple clustering algorithm were implemented in lingpipe. We provided predefined number of clusters as a parameter for both algorithms. The simple LDA clustering algorithm uses the same initial steps described in our method without the cluster refinement step. We treated initial clusters as final clusters and skipped the loop which refines the cluster using feature weights. The parameters for LDA are *number of topics = number of clusters in ground truth*, *number of clusters = number of clusters in ground truth*, $\alpha = 0.1$ and $\beta = 0.01$. We tuned the parameter α and β for the best performance. FWKM, EWKM and FGKM clustering algorithm were implemented in Weka[1] and we used standard parameters as described by the authors.

The performance of all six clustering algorithms for synthetic dataset is shown in Table 2 and for real-world dataset is shown in Table 3.

Table 2 shows the comparison of clustering methods in terms of Accuracy and F-measure on four synthetic datasets. The values in bold represent the best

[1] The code for FWKM, EWKM and FGKM was provided by the authors

Table 3: A comparison of clustering methods in terms of F-measure, NMI and Entropy on six real-world datasets created from 20-Newsgroup dataset. The values listed in the table are the mean values of 100 runs of five clustering methods on six real-world datasets

Datasets	Metric	k-means	LDA	FWKM	EWKM	FGKM	DWKM
D1	F-measure	**0.96**	**0.96**	0.95	**0.96**	**0.96**	**0.96**
	NMI	0.78	0.78	0.79	0.83	0.85	**0.86**
	Entropy	0.21	0.21	0.20	0.16	0.15	**0.13**
D2	F-measure	0.93	0.92	0.90	0.91	0.94	**0.96**
	NMI	0.80	0.78	0.75	0.76	0.78	**0.80**
	Entropy	0.19	0.24	0.25	0.23	0.17	**0.15**
D3	F-measure	0.89	0.90	0.95	0.95	0.95	**0.96**
	NMI	0.71	0.72	0.84	0.86	0.87	**0.88**
	Entropy	0.28	0.20	0.15	0.11	0.10	**0.08**
D4	F-measure	0.88	0.90	0.90	0.94	0.95	**0.96**
	NMI	0.47	0.55	0.60	0.72	0.75	**0.78**
	Entropy	0.52	0.30	0.40	0.28	0.27	**0.20**
D5	F-measure	0.70	0.75	0.86	0.89	0.90	**0.92**
	NMI	0.38	0.48	0.64	0.68	0.70	**0.73**
	Entropy	0.61	0.41	0.35	0.31	0.30	**0.29**
D6	F-measure	0.65	0.81	0.92	0.92	0.93	**0.94**
	NMI	0.37	0.68	0.73	0.75	0.76	**0.78**
	Entropy	0.53	0.28	0.23	0.23	0.22	**0.19**

results. In general, DWKM performs better than other clustering methods in terms of both Accuracy and F-measure on the synthetic datasets. The Accuracy and F-measure values on datasets SD1 and SD2 for DWKM and FGKM have large gaps, whereas the differences of the values on datasets SD3 and SD4 are relatively smaller. The LDA based simple clustering performed better than standard k-means, but performed worse than soft subspace clustering algorithms.

Table 3 shows the mean values of F-measure, NMI and Entropy for k-means, FWKM, FGKM and DWKM clustering methods on six real-world datasets. In general, on the six real-world data set DWKM performed better than other clustering methods in terms of F-measure, NMI and Entropy values. The D1 dataset is the easiest dataset. K-means, EWKM, FGKM and DWKM have the same F-measure value **0.96** on D1 dataset, which means these clustering methods produced equally good clustering solutions. However, if we consider the NMI and Entropy values along with F-measure value of the D1 dataset, we can see that DWKM performed slightly better than other clustering methods. The LDA based simple clustering followed the same trend as in synthetic datasets and performed better than standard k-means, but worse than soft subspace clustering algorithms.

It was also observed from the results that DWKM performed well on data with different level of difficulties (data without noise, with noise, with balanced clusters and with unbalanced clusters). This shows that our semantic weight-

Table 4: Percentage improvement of DWKM over FGKM in terms of Accuracy(AC) and F-measure (FM) on synthetic datasets

	AC % (IMP)	FM % (IMP)
SD1	5.75	7.41
SD2	5.43	6.82
SD3	0.00	1.09
SD4	2.15	2.22

Table 5: Percentage improvement of DWKM over FGKM in terms of F-measure (FM), NMI and Entropy (EN) on real datasets

	FM % (IMP)	NMI % (IMP)	EN % (IMP)
D1	0.000	1.163	2.299
D2	2.083	2.500	2.353
D3	1.042	1.136	2.174
D4	1.042	3.846	8.750
D5	4.255	4.110	1.408
D6	2.105	2.564	3.704

ing of subspaces derived from LDA is reasonably effective for finding clusters in different types of data. Moreover the LDA based simple clustering algorithm performed much better than k-means algorithm when datasets had semantically related clusters (results of D4 and D5). It was also noted that the use cluster refinement step based on feature weighting of LDA model boosted the performance of clustering solution. The DWKM algorithm without the cluster refinement step, performed better than k-means algorithm and slightly worse than other clustering methods.

Table 4 and table 5 provide percentage improvement of DWKM over FGKM on synthetic datasets and real datasets respectively. The results in all tables suggest that DWKM is a better clustering method. We further investigate the performance of all clustering methods by conducting a statistical analysis.

5.2 Statistical Analysis

We performed two types of statistical tests: 1) unpaired t-test and 2) paired Wilcoxon statistical significance test [28] by considering DWKM as the control group. The unpaired ttest was performed using the standard deviation and mean values of evaluation measures listed in table 2. In general the results from unpaired ttest showed that DWKM achieved statistically significant improvement over three methods k-means, FWKM and EWKM on all synthetic datasets with p-value less then **0.05**. The p-values of unpaired ttest computed for FGKM on SD1 and SD2 synthetic datasets are less than **0.05**, which indicates that our method DWKM has statistical significant improvement on SD1 and SD2 over FGKM. The performance of our method on other SD3 and SD4 synthetic dataset was found to be comparable over FGKM.

For the six real-world dataset we used paired Wilcoxon statistical significance test. The p-values of F-measure, NMI and Entropy values for FGKM were **0.0305**, **0.0028** and **0.0228** respectively. In general the p-values for all five clustering methods were found to be less than **0.05**, which suggested that our method DWKM shows a better performance and significant improvement over five clustering methods.

Table 6: P-values of unpaired ttest of DWKM and FGKM on synthetic datasets

SD1		SD2		SD3		SD4	
Accuracy	F-measure	Accuracy	F-measure	Accuracy	F-measure	Accuracy	F-measure
0.0237	**0.0449**	**0.0237**	**0.0449**	1	0.6867	0.278	0.4457

6 Conclusion

In this paper, we introduced a new soft subspace clustering method which uses LDA model to weight the features in the subspaces for clustering documents. The LDA model was implemented using a standard Gibbs sampling algorithm, and it generated two matrices: topic-term and topic-documents. We used the topic-term matrix to develop a new weighted distance measure, where topics are used as subspaces. We developed a k-mean based soft subspace clustering method based on our new weighted distance measure. The algorithm is initialized using the topic-document matrix, where topics are considered as initial clusters.

Our new method DWKM, was found to achieve a statistically significant improvement over recently developed soft subspace clustering methods on synthetic and real-world datasets.

Currently the method requires users to input the number of topics to initialize the LDA model. In future we will remedy this by investigating non-parametric LDA models and will try to reduce the computational complexity of the overall method. Another direction for the future work is to investigate the use of LDA to generate different candidate clustering solutions for clustering ensemble methods.

References

1. C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. In *ACM SIGMOD Record*, volume 28, pages 61–72. ACM, 1999.
2. C. C. Aggarwal and P. S. Yu. *Finding generalized projected clusters in high dimensional spaces*, volume 29. ACM, 2000.
3. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. *Automatic subspace clustering of high dimensional data for data mining applications*, volume 27. ACM, 1998.
4. I. Bhattacharya and L. Getoor. A latent dirichlet model for unsupervised entity resolution. In *SDM*, volume 5, page 59. SIAM, 2006.
5. D. M. Blei and J. D. Lafferty. Topic models. *Text mining: classification, clustering, and applications*, 10:71, 2009.
6. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
7. E. Y. Chan, W. K. Ching, M. K. Ng, and J. Z. Huang. An optimization algorithm for clustering using weighted dissimilarity measures. *Pattern recognition*, 37(5):943–952, 2004.

8. X. Chen, X. Xu, J. Z. Huang, and Y. Ye. Tw-(k)-means: Automated two-level variable weighting clustering algorithm for multiview data. *Knowledge and Data Engineering, IEEE Transactions on*, 25(4):932–944, 2013.
9. X. Chen, Y. Ye, X. Xu, and J. Z. Huang. A feature group weighting method for subspace clustering of high-dimensional data. *Pattern Recognition*, 45(1):434–446, 2012.
10. C.-H. Cheng, A. W. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 84–93. ACM, 1999.
11. G. De Soete. Optimal variable weighting for ultrametric and additive tree clustering. *Quality and Quantity*, 20(2-3):169–180, 1986.
12. G. De Soete. Ovwtre: A program for optimal variable weighting for ultrametric and additive tree fitting. *Journal of Classification*, 5(1):101–104, 1988.
13. W. S. DeSarbo, J. D. Carroll, L. A. Clark, and P. E. Green. Synthesized clustering: A method for amalgamating alternative clustering bases with differential weighting of variables. *Psychometrika*, 49(1):57–78, 1984.
14. C. Domeniconi, D. Papadopoulos, D. Gunopulos, and S. Ma. Subspace clustering of high dimensional data. In *SDM*, volume 73, page 93. SIAM, 2004.
15. J. H. Friedman and J. J. Meulman. Clustering objects on subsets of attributes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(4):815–849, 2004.
16. H. Frigui and O. Nasraoui. Simultaneous clustering and dynamic keyword weighting for text documents. In *Survey of text mining*, pages 45–72. Springer, 2004.
17. H. Frigui and O. Nasraoui. Unsupervised learning of prototypes and attribute weights. *Pattern recognition*, 37(3):567–581, 2004.
18. S. Goil, H. Nagesh, and A. Choudhary. Mafia: Efficient and scalable subspace clustering for very large data sets. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 443–452, 1999.
19. L. Jing, M. K. Ng, and J. Z. Huang. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *Knowledge and Data Engineering, IEEE Transactions on*, 19(8):1026–1041, 2007.
20. L. Jing, M. K. Ng, J. Xu, and J. Z. Huang. Subspace clustering of text documents with feature weighting k-means algorithm. In *Advances in Knowledge Discovery and Data Mining*, pages 802–812. Springer, 2005.
21. V. Makarenkov and P. Legendre. Optimal variable weighting for ultrametric and additive trees and k-means partitioning: Methods and software. *Journal of Classification*, 18(2):245–271, 2001.
22. D. S. Modha and W. S. Spangler. Feature weighting in k-means clustering. *Machine learning*, 52(3):217–237, 2003.
23. N. Nguyen and R. Caruana. Consensus clusterings. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 607–612. IEEE, 2007.
24. M. Steyvers and T. Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.
25. A. Wahid, X. Gao, and P. Andreae. Exploiting user queries for search result clustering. In *Web Information Systems Engineering–WISE 2013*, pages 111–120. Springer, 2013.
26. A. Wahid, X. Gao, and P. Andreae. Multi-view clustering of web documents using multi-objective genetic algorithm. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 2625–2632. IEEE, 2014.

27. A. Wahid, X. Gao, and P. Andreae. Multi-objective multi-view clustering ensemble based on evolutionary approach. In *To appear in Evolutionary Computation, 2015. CEC'15. IEEE Congress on*. IEEE, 2015.

28. F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, pages 80–83, 1945.

29. K.-G. Woo, J.-H. Lee, M.-H. Kim, and Y.-J. Lee. Findit: a fast and intelligent subspace clustering algorithm using dimension voting. *Information and Software Technology*, 46(4):255–271, 2004.

30. J. Yang, W. Wang, H. Wang, and P. Yu. δ-clusters: Capturing subspace correlation in a large data set. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 517–528. IEEE, 2002.

31. Y. Zhao and G. Karypis. Comparison of agglomerative and partitional document clustering algorithms. Technical report, DTIC Document, 2002.

32. S. Zhong and J. Ghosh. A comparative study of generative models for document clustering. In *Proceedings of the workshop on Clustering High Dimensional Data and Its Applications in SIAM Data Mining Conference*, 2003.