PSEUDO-LABEL GENERATION FOR MULTI-LABEL TEXT CLASSIFICATION

MOHAMMAD SALIM AHMED[1], LATIFUR KHAN[1], AND NIKUNJ OZA[2]

Abstract. With the advent and expansion of social networking, the amount of generated text data has seen a sharp increase. In order to handle such a huge volume of text data, new and improved text mining techniques are a necessity. One of the characteristics of text data that makes text mining difficult, is multi-labelity. In order to build a robust and effective text classification method which is an integral part of text mining research, we must consider this property more closely. This kind of property is not unique to text data as it can be found in non-text (e.g., numeric) data as well. However, in text data, it is most prevalent. This property also puts the text classification problem in the domain of *multi-label classification (MLC)*, where each instance is associated with a subset of class-labels instead of a single class, as in conventional classification. In this paper, we explore how the generation of pseudo labels (i.e., combinations of existing class labels) can help us in performing better text classification and under what kind of circumstances. During the classification, the high and sparse dimensionality of text data has also been considered. Although, here we are proposing and evaluating a text classification technique, our main focus is on the handling of the multi-labelity of text data while utilizing the correlation among multiple labels existing in the data set. Our text classification technique is called *pseudo-LSC (pseudo-Label Based Subspace Clustering)*. It is a subspace clustering algorithm that considers the high and sparse dimensionality as well as the correlation among different class labels during the classification process to provide better performance than existing approaches. Results on three real world multi-label data sets provide us insight into how the multi-labelity is handled in our classification process and shows the effectiveness of our approach.

1. Introduction

Classification is an important part of text data analysis as has been pointed out in text research over a long period of time. With the increase of its volume, it has become necessary that we find automated means for text classification. However, text data is different from its non-text counterpart in a number of ways. The first difference that we look into and address in this paper is that text data tends to address multiple topics at the same time. As a result, they can be associated with multiple class labels giving rise to multi-labelity. And, these class labels are not independent of one another, indicating the existence of correlation or label dependence across the class labels. One of the main contributions of this paper is to take this correlation into account during the classification process.

We must also consider the high and sparse dimensionality of text data. All documents in text data sets are written in plain language. Since the vocabulary of any natural language is vast, the dimensionality is very high and compared to the whole vocabulary, only a few words appear in each document which gives rise to the sparseness.

[1] The University of Texas at Dallas, salimahmed@utdallas.edu, lkhan@utdallas.edu
[2] NASA Ames Research Center, nikunj.c.oza@nasa.gov.

Another important consideration during text classification is the availability of labeled data. Manual labeling of data is a time consuming task and as a result, in many cases, they are available in limited quantity. If we consider just the labeled data, then we are sometimes left with too little data to build a classification model that can perform well. On the other hand, if we ignore the class label information of the labeled data, then we are forsaking valuable information that could allow us to build a better classification model.

If we look into the literature, we see that usually, text classification approaches focus on a specific characteristic of text data such as its high dimensionality, multi-labelity or availability of limited labeled data. As a result, many of these methods can not be used universally. Sometimes, the underlying theory of these methods may become incorrect. For example, *Entropy Weighting K Means* approach [12] uses a subspace clustering approach that is based on the entropy of the features or dimensions. If the data is multi-label, then the entropy calculation of that method no longer holds ground. This happens in case of *SISC* [3], too, which is our previously formulated semi-supervised subspace clustering algorithm that considers both the high dimensionality and limited labeled data challenges. In *SISC* [3], however, the measure that becomes incorrect is the class impurity calculation. Also, it is only applicable for multi-class text data, not multi-label data, let alone considerations of label correlation.

In face of all these challenges, traditional as well as state-of-the-art text classification approaches perform poorly on multi-label data sets as we have found through our experiments. In order to address this multi-labelity scenario, we extended *SISC* to formulate *SISC-ML*[4] which is a multi-label variation of *SISC*. However, if we look closely into the data, we find that not all the classes co-occur with the same frequency. Which implies that the correlation among different class labels are not the same. In order to incorporate this correlation information during the clustering process, We, therefore, extended *SISC* [3] further based on this correlation information and formulated *pseudo-LSC* in this paper.

The reason behind choosing *SISC* as our classification method for extension is due to its notion of subspace clustering. Subspace clustering allows us to find clusters in a weighted hyperspace [10] and can aid us in finding documents that form clusters in only a subset of dimensions. In our proposed *pseudo-LSC (pseudo-Label Based Subspace Clustering)* approach, we augment the original class labels in the data set with *pseudo-labels* which are actually combinations of multiple class labels. Assigning such pseudo-labels allows us to use the correlation among different class labels during clustering and to achieve better classification performance.

In short, we have a number of contributions in this paper. First, *pseudo-LSC* is a semi-supervised subspace clustering algorithms that successfully finds clusters in the subspace of dimensions irrespective of the data being multi-class or multi-label. Second, our proposed algorithm performs well in practice even when a very limited amount of labeled training data is available. Third, at the same time, this algorithm minimizes the effect of high dimensionality and its sparse nature during training. Finally, we compare *pseudo-LSC* with other classification and clustering approaches to show the effectiveness of our algorithms over three benchmark multi-label text data sets.

The organization of the paper is as follows: Section 2 discusses related works. Section 3 presents the theoretical background of *pseudo-LSC*, the semi-supervised multi-class text classification approach. Section 4, then describes how *pseudo-LSC* handles multi-labeled data. Section 5 discusses the data sets, experimental setup and evaluation of our approach. Finally, Section 6 concludes with directions to future research.

2. Related Work

We can divide our related work based on the characteristic of our proposed *pseudo-LSC*. *pseudo-LSC* is a semi-supervised approach, it uses subspace clustering, and most important of all, it can handle multi-label data. Therefore, we have to look into the state-of-the-art methods that are already in the literature for each of these categories of research.

Approaches that have been proposed to address multi-label text classification, including margin-based methods, structural SVMs [19], parametric mixture models [21], κ-nearest neighbors (κ-NN) [25], and ensemble pruned methods [16]. One of the most recent works include *RAndom k-labELsets (RAKEL)* [20]. In a nutshell, it constructs an ensemble of *LP (Label Powerset)* classifiers and each *LP* is trained using a different small random subset of the multi-label set. Then, ensemble combination is achieved by thresholding the average zero-one decisions of each model per considered label. *MetaLabeler* [18] is another approach which tries to predict the number of labels using *SVM* as the underlying classifier. Most of these methods utilize the relationship between multiple labels for collective inference. One characteristic of these models is they are mostly supervised [16, 20, 18]. Aside from multi-label text classification, there are also work on regret analysis and loss function for such classification. In [9], Dembczynski et al. compare two loss functions namely subset 0/1 loss and Hamming loss for different multi-label classifiers. They focus mainly on the close connection between conditional label dependence and loss minimization. Unlike their approach, we are utilizing the unconditional label correlation that exists in the data as well as cluster impurity minimization.

Semi-supervised methods for classification is also present in the literature. This approach stems from the possibility of having both labeled and unlabeled data in the data set and in an effort to use both of them in training. In [5], Bilenko et al. propose a semi-supervised clustering algorithm derived from *K-Means*, *MPCK-MEANS*, that incorporates both metric learning and the use of pairwise constraints in a principled manner. There have also been attempts to find a low-dimensional subspace shared among multiple labels [12]. In [24], Yu et al. introduce a supervised *Latent Semantic Indexing (LSI)* method called *Multi-label informed Latent Semantic Indexing (MLSI)*. *MLSI* maps the input features into a new feature space that retains the information of original inputs and meanwhile captures the dependency of output dimensions. Our method is different from this algorithm as our approach tries to find clusters in the subspace. Due to the high dimensionality of feature space in text documents, considering a subset of weighted features for a class is more meaningful than combining the features to map them to lower dimensions [12]. In [7] a method called *LPI* is proposed. *LPI* is different from *LSI* which aims to discover the global Euclidean structure whereas *LPI* aims to discover the local geometrical structure. But *LPI* only handles multi-class data, not multi-label data. In [17] must-links and cannot-links, based on the labeled data, are incorporated in clustering. But, if the data is multi-label, then the calculation of must-link and cannot-link becomes infeasible as there are large number of class combinations and the number of documents in each of these combinations may be very low. As a result, this framework can not perform well when using multi-label text data.

There has been some subspace clustering approaches to minimize the impact of high dimensionality on classification. Subspace clustering can be divided into hard and soft subspace clustering. In case of hard subspace clustering, an exact subset of dimensions are discovered whereas soft subspace clustering determines the subsets of dimensions according to the contributions of the dimensions in discovering corresponding clusters. Examples of hard subspace clustering include *CLIQUE* [2], *PROCLUS* [1], *ENCLUS* [8] and *MAFIA* [11]. A hierarchical subspace clustering approach with automatic relevant dimension selection, called *HARP*, was presented by Yip et al. [23]. *HARP* is based on the assumption that two objects are likely to belong to the same cluster if they are very similar to each other along many dimensions. But, in multi-label and high dimensional text environment, the accuracy of *HARP* may drop as the basic assumption becomes less valid. In [14], a subspace clustering method called *nCluster* is proposed. But, it has similar problems when dealing with multi-label data. In [22], Wang et al. focuses on an ensemble approach and proposes a nonparametric Bayesian clustering ensemble method to discover the number of clusters for consensus clustering. In [13], SciForest has been proposed which uses clustering for finding group of anomalies/outliers. There, Liu et al. employ a split selection criterion to choose a split that separates clustered anomalies from normal points. They also makes use of randomly generated hyper-planes in order to provide suitable projections that separate anomalies from normal points. However, such a clustering is not applicable for multi-label text classification. Other soft clustering include [6] where spectral decomposition of the normalized affinity matrix is performed. The affinity matrix indicates the similarity measure between any two instances in the training set and therefore, depends too much on the quality of the similarity measure. Also, [6] focuses on using such clustering on graph data rather than text data.

pseudo-LSC uses subspace clustering in conjunction with *κ-NN* approach. In this light, it is closely related to the work of Jing et al. [12], Frigui et al. [10] and Ahmed et al. [3]. The closeness is due to the subspace clustering and fuzzy framework respectively. A significant difference with Frigui et al. [10] is that, unlike *pseudo-LSC*, it is unsupervised in nature. Another work that is closely related to ours is the work of Masud et al. [15]. In [15], a semi-supervised clustering approach called *SmSCluster* is used. They have used simple *K-Means Clustering* and it is specifically designed to handle evolving data streams. Finally, *SISC* [3] is another subspace clustering approach which has close resemblance to our approach. But, it is designed for only multi-class data. It has a multi-label variation called *SISC-ML* [4]. However, as mentioned previously, *SISC-ML* does not consider the class label correlation and assumes the class labels to be independent of each other. Our proposed *pseudo-LSC* is different in this respect as it does not make such class label independence assumption. It is also not specific for multi-class or multi-label data as is the case for *SISC* or *SISC-ML*. *pseudo-LSC* can work with a data set irrespective of it being multi-class or multi-label and therefore, addresses many of the challenges associated with text classification simultaneously.

Notation	Range	Explanation
x_i	$i = 1 : n$	i-th data point in the n document data set
d_j	$j = 1 : m$	j-th binary feature of m unigram features for data point x_i
t_i	$i = 1 : p$	i-th class of p classes in the data set where $p = \|\mathcal{T}\|$
c_l	$l = 1 : k$	l-th cluster of k subspace clusters
w_l	$l = 1 : k$	Membership weight of data point x_i of l-th subspace cluster
Lc_l	-	Total number of labeled points in cluster c_l

Table 1. SISC Notations

3. Multi-Label Classification

In this section, we describe the *MLC* problem in more detail and formalize it from a soft-clustering perspective. Along the way, we introduce the notations used throughout the paper.

3.1. Problem Statement.

Let $\mathcal{X}$ denote the training instance space $\hat{\mathcal{X}}$ denote the test set. Also, let $\mathcal{T} = t_1, t_2, \ldots, t_p$ be a finite set of class labels. We assume that any instance x across the training and test set is associated with a subset of labels $T \in 2^{\mathcal{T}}$; this subset is often called the set of relevant class labels while the complement of T is considered as irrelevant for x. Our goal is to predict the probability of a test instance $\hat{x}_i$ to belong to each class label $t_r, r = 1 : p$. In short, for each test instance $\hat{x}_i \in \hat{\mathcal{X}}$, we generate a class label vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_p)$, in which $y_i = [0, 1]$ and $\sum_{i=1}^{p} y_i = 1$.

In this paper, we define a multi-label classifier $\mathbf{h}$ as an $\mathcal{X} \to \mathcal{Y}$ mapping that assigns a class-label vector $\boldsymbol{y}_i \in \hat{\mathcal{Y}}$ to each test instance $\hat{x}_i \in \hat{\mathcal{X}}$. Therefore, the problem of *MLC* can be stated as follows:

Given training data in the form of a finite set of observations $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$, the goal is to learn a classifier $\mathbf{h} : \mathcal{X} \to \mathcal{Y}$ that generalizes well beyond the training observations. Table 1 specifies some of the notations that will be used throughout this paper.

It should be noted that since we are using a soft subspace clustering formulation, each training instance $x_i \in \mathcal{X}$ is a member of all the k subspace clusters (but with different membership weights). Apart from these notations, the following *two* measures are also used in *pseudo-LSC* as has been defined for *SISC* in [3].

3.2. Description of pseudo-LSC.

In *pseudo-LSC*, each data point may belong to multiple clusters. The weight with which a data point belongs to a particular cluster is referred to as *cluster membership weight*. For a data point, these membership weights across all the clusters sum up to 1. So, the membership weights can be regarded as probabilities with which a data point belongs to a cluster. Also, *pseudo-LSC* applies subspace clustering and the weight of a dimension in a cluster represents the probability of contribution of that dimension in forming that cluster. These dimension weights within a cluster are kept as a vector and the different dimension vectors of the clusters indicate how the clusters are different from one another. We, therefore, have to update three parameters during our clustering process - the *dimension weights* within each cluster, the

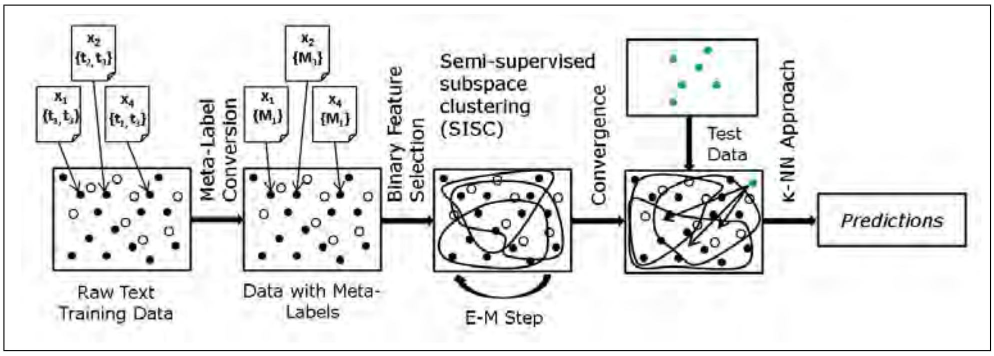


Figure 1. pseudo-LSC Top Level Diagram

cluster membership weights of each data point and the *cluster centroids*. *pseudo-LSC* utilizes the *Expectation-Maximization(E-M)* approach that locally minimizes the following objective function.

$$F(W, Z, \Lambda) = \sum_{l=1}^{k}\sum_{j=1}^{n}\sum_{i=1}^{m} w_{lj}^{f}\lambda_{li}^{q}D_{lij} * (1 + Imp_l) + \gamma\sum_{l=1}^{k}\sum_{i=1}^{m}\lambda_{li}^{q}\chi_{li}^{2} \tag{1}$$

where

$$D_{lij} = (z_{li} - x_{ji})^2$$

subject to

$$\sum_{l=1}^{k} w_{lj} = 1, 1 \leq j \leq n, 1 \leq l \leq k, 0 \leq w_{lj} \leq 1$$

$$\sum_{i=1}^{m} \lambda_{li} = 1, 1 \leq i \leq m, 1 \leq l \leq k, 0 \leq \lambda_{li} \leq 1$$

In this objective function, W, Z and Λ represent the *cluster membership*, *cluster centroid* and *dimension weight* matrices respectively. Also, the parameter f controls the fuzziness of the membership of each data point, q further modifies the weight of each dimension (λ_{li}) of each cluster c_l and finally, γ controls the strength of the incentive given to the *Chi Square* component.

Our algorithm is formulated using the E-M approach. In the E-Step, the *dimension weights* and the *cluster membership weights* are updated. Initially, every data point has equal membership weights across the clusters and the dimensions are given equal weights, too. During the *dimension weight* and *cluster membership weight* update, the cluster impurity is calculated using the pseudo-labels, not the original class labels. In the M-Step, the centroids of the clusters are updated and the summary statistics, i.e., the representation (percentage) of each class label present in the cluster, is updated. During the summary calculation, the membership weights are used. In the final step, the κ nearest neighbor clusters are identified for each test point where κ is a user defined parameter. The distance is calculated in the subspace where the cluster resides. If κ is greater than 1, then during the class probability calculation, we multiply the class representation with the inverse of the subspace distance and then sum them up for each class across all the κ nearest clusters.

3.3. **Impurity Measure.** Each cluster $c_l, l = 1 : k$, has an *Impurity Measure* [15] associated with it. This measure quantifies the amount of impurity within each cluster c_l. If the data points belonging to c_l all have the same class label, then the *Impurity Measure* of this cluster Imp_l is 0. On the other hand, if more and more data points belonging to different class labels become part of cluster c_l, the *Impurity Measure* of this cluster Imp_l also increases. This component has been used to modify the dispersion measure for each cluster. Its use helps in generating purer clusters in terms of cluster labels. However, it should be noted that Imp_l can be calculated using only labeled data points. If there are very few labeled data points, then this measure does not contribute significantly during the clustering process. Therefore, we use $1 + Imp_l$, so that unlabeled data points can play a role in the clustering process. Using Imp_l in such a way makes *pseudo-LSC* semi-supervised.

$$Imp_l = ADC_l * Ent_l$$

Here, ADC_l indicates the *Aggregated Dissimilarity Count* and Ent_l denotes the entropy of cluster c_l. In order to measure ADC_l, we first need to define *Dissimilarity Count* [15], $DC_l(x_i, y_i)$:

$$DC_l(x_i, y_i) = |L_{c_l}| - |L_{c_l}(t)|$$

if x_i is labeled and its label $y_i = t$, otherwise $DC_l(x_i, y_i)$ is 0. L_{c_l} indicates the set of labeled points in cluster c_l. In short, it counts the number of labeled points in cluster c_l that do not have label t. Then ADC_l becomes

$$ADC_l = \sum_{x_i \in L_{c_l}} DC_l(x_i, y_i)$$

Summing up the ADC_l for all the class labels provide us with the ADC_l for the entire cluster. The *Entropy* of a cluster c_l, Ent_l is computed as

$$Ent_l = \sum_{t=1}^{|\mathcal{T}|} (-p_t^l * log(p_t^l))$$

where p_t^l is the prior probability of class t, i.e., $p_t^l = \frac{|L_{c_l}(t)|}{|L_{c_l}|}$. It can also be shown that ADC_l is proportional to the *gini index* of cluster c_l, $Gini_l$ [15]. But, in *pseudo-LSC* method, the data points have fuzzy cluster memberships. So, the ADC_l calculation needs to be modified to incorporate this concept into *pseudo-LSC*. Rather than using counts, the membership weights are used for the calculation. This is reflected in the probability calculation.

$$p_t^l = \sum_{j=1}^{n} w_{lj} * j_t$$

where, j_t is 1 if data point x_j is a member of class t, and 0 otherwise. This *Impurity Measure* is normalized using the global impurity measure, i.e., the impurity measure of the whole data set, before using in the subspace clustering formulation.

3.4. **Chi Square Statistic.** During the clustering process, it may happen that the clusters are formed using only a few features. However, if only a few features (e.g., 2 or 3 features) are involved in the clustering with their dimension weights being greater than 0, they may fail to play any role during the label prediction step. This may happen as those few features may never appear in a test document rendering us unable to ascertain the κ-NN clusters of a test data point. To prevent such a scenario to happen, this *Chi Square* component has been included in the objective function so that more features or dimensions have nonzero weights and can participate in the clustering process. It works against the dispersion component of the objective function to create a balancing effect and ensures that clusters are not formed in just a few dimensions. From a clustering perspective, the conventional *Chi Square Statistic* can be defined as,

$$\chi^2_{li} = \frac{m(s_1 s_4 - s_2 s_3)^2}{(s_1+s_3)(s_2+s_4)(s_1+s_2)(s_3+s_4)}$$

where

$$\begin{aligned}
&s_1 = number\ of\ times\ feature\ d_i\ occurs\ in\ cluster\ c_l\\
&s_2 = number\ of\ times\ feature\ d_i\ occurs\ in\ all\\
&\quad clusters\ except\ c_l\\
&s_3 = number\ of\ times\ cluster\ c_l\ occurs\ without\ feature\ d_i\\
&s_4 = number\ of\ times\ all\ clusters\ except\ c_l\ occur\\
&\quad without\ feature\ d_i\\
&m = number\ of\ dimensions
\end{aligned}$$

This *Chi Square Statistic* χ^2_{li} indicates the measure for cluster c_l and dimension d_i. However, if the conventional approach is used for calculation of s_1, s_2, s_3, s_4 and m, then a threshold has to be specified to determine which point can be regarded as a member of a cluster. This not only brings forth another parameter, but also the membership values themselves are undermined in the calculation. So, *pseudo-LSC* modifies the calculation of these counts to consider the corresponding membership values of each point. The modification is provided below:

$$\begin{aligned}
s_1 &= \sum_{j=1}^{n}\sum_{d_i \in x_j} w_{lj}, \qquad s_2 = 1 - \sum_{j=1}^{n}\sum_{d_i \in x_j} w_{lj}\\
s_3 &= \sum_{j=1}^{n}\sum_{d_i \notin x_j} w_{lj}, \qquad s_4 = 1 - \sum_{j=1}^{n}\sum_{d_i \notin x_j} w_{lj}\\
m &= total\ number\ of\ labeled\ points
\end{aligned}$$

3.5. **Update Equations.** Minimization of F in Eqn. 5 with the constraints, forms a class of constrained nonlinear optimization problems. This optimization problem can be solved using partial optimization for Λ, Z and W. Detailed derivation of the update equations can be found in [3].

3.5.1. *Dimension Weight Update Equation.* Given matrices W and Z are fixed, F is minimized if

(2) $$\lambda_{li} = \frac{1}{M_{lij}\sum_{i=1}^{m}\frac{1}{M_{lij}}}$$

where

$$M_{lij} = \left\{ \sum_{j=1}^{n} w_{lj}^{f} D_{lij} * (1 + Imp_l) + \gamma \chi_{li}^{2} \right\}^{\frac{1}{q-1}}$$

3.5.2. *Cluster Membership Update Equation.* Similar to the dimension update equation, the update equations for cluster membership matrix W can be derived, given Z and Λ are fixed. The update equation is as follows:

$$w_{lj} = \frac{1}{N_{lij} \sum_{l=1}^{k} \frac{1}{N_{lij}}} \tag{3}$$

where

$$N_{lij} = \left\{ \sum_{i=1}^{m} \lambda_{li}^{q} D_{lij} \right\}^{\frac{1}{f-1}}$$

3.5.3. *Cluster Centroid Update Equation.* The cluster center update formulation is similar to the formulation of dimension and membership update equations. The update equations for cluster center matrix i.e., Z can be derived, given W and Λ are fixed. The update equation is as follows:

$$z_{li} = \frac{\sum_{j=1}^{n} w_{lj}^{f} x_{ij}}{\sum_{j=1}^{n} w_{lj}^{f}} \tag{4}$$

4. Handling Multi Labeled Data

The previously described *Impurity Measure* calculation is only applicable for multi-class data where each document may belong to only a single class label. This constraint ensures that the calculated probabilities always sum up to 1. However, if each data point may belong to more than one class label, the sum of probabilities may become greater than 1. One way would be to convert the classification problem into T binary class problems and modify the *Impurity Measure* to handle the multi-label data in such a way. But, doing so is only feasible if all the class labels are independent of each other. Our experience with multi-label data also indicate that there is a correlation among the different class labels. In order to handle the multi-labelity as well as the co-occurrence of the class labels, our proposed *pseudo-LSC* generates *pseudo-labels* which are combinations of one or more original class labels in the data set. In short, we transform the multi-label data set into a multi-class data set where each data point can belong to only a single *pseudo-label.* The following example illustrates how the *pseudo-labels* are generated.

As can be seen from the example in Table 2, the 5 data points belong to 3 *pseudo-labels.* And each of the *pseudo-labels* may constitute of one or more original class labels in the data set. After assigning such *pseudo-labels* to the data points, the new data set becomes multi-class. Therefore, the *pseudo-LSC* multi-class algorithm becomes applicable to such a data set. In Figure 3.1, we show how the text data is converted from its original label to *pseudo-labels.* In this example, data points x_1, x_2 and x_3 are assigned *pseudo-labels* p_1, p_2 and p_3 respectively. This pseudo-label generation is

Data	Labels
x_1	t_1, t_3
x_2	t_1, t_2, t_4
x_3	t_2
x_4	t_1, t_3
x_5	t_2

Pseudo Labels	Label Sets
p_1	t_1, t_3
p_2	t_1, t_2, t_4
p_3	t_2

Data	Pseudo Labels
x_1	p_1
x_2	p_2
x_3	p_3
x_4	p_1
x_5	p_3

Table 2. Construction of Pseudo Labels In pseudo-LSC

only applicable during the *Impurity Measure* calculation. The original class labels are used in all other calculations during the classification process.

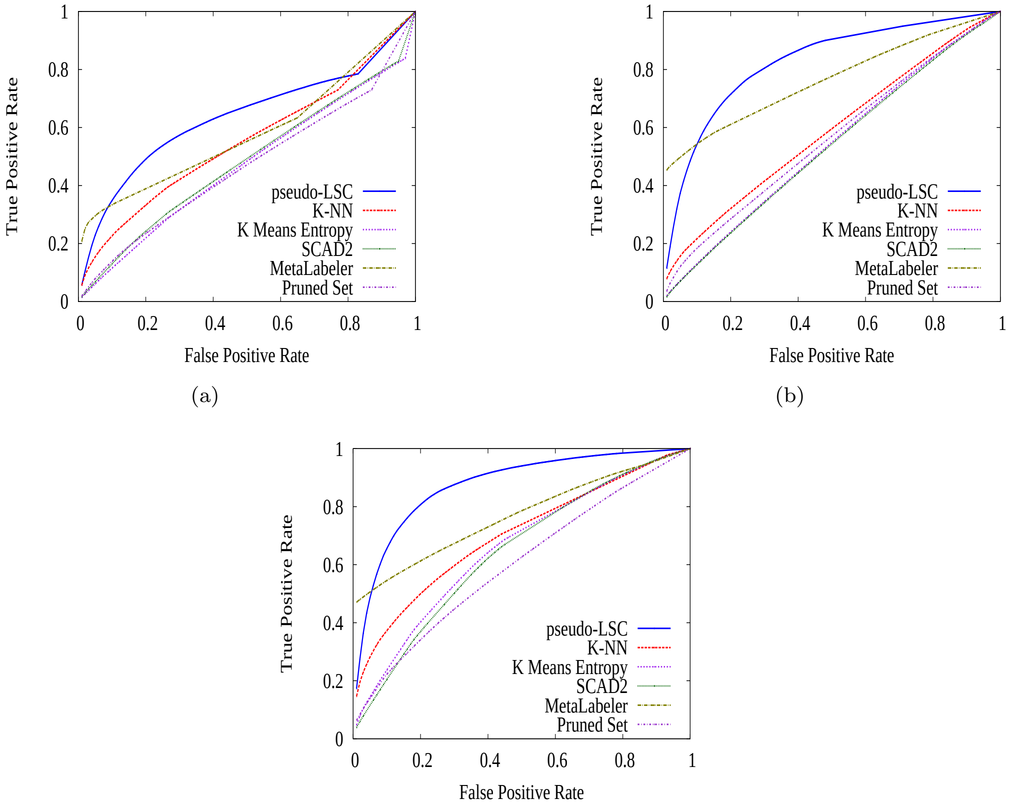
5. Experiments and Results

We have used a total of three multi-label data sets to verify the effectiveness of our algorithm on multi-label data. In all cases, we used fifty percent data as training and rest as test data in our experiments as part of 2-fold cross-validation. Similar to other text classification approaches, we performed preprocessing on the data and removed stop words from the data. We used binary features as dimensions, i.e. features can only have 0 or 1 values. The parameter γ is set to 0.5. For convenience, we selected 1000 features based on information gain and used them in our experiments. In all the experiments related to a data set, the same feature set was used. We performed multiple runs on our data sets. And in each case, the training set was chosen randomly from the data set.

5.1. **Data sets.** We describe here all the three data sets that we have used for our experiments.

(1) Reuters Data Set: This is part of the Reuters-21578, Distribution 1.0. We selected $10,000$ data points from the $21,578$ data points of this data set and henceforth, this part of the data set will be referred to as simply *Reuters* data set. We considered the most frequently occurring 20 classes in our experiments. Of the $10,000$ data points, $6,651$ are multi-labeled.
(2) 20 Newsgroups Data Set: This data set is also multi-label in nature. We selected $15,000$ documents randomly for our classification experiments. Of them $2,822$ are multi-label documents and the rest are single labeled. We have performed our classification on the top 20 classes of this data set.
(3) NASA ASRS Data Set: We randomly selected $10,000$ data points from the *ASRS* data set and henceforth, this part of the data set will be referred to as simply *ASRS* Data Set. We considered 21 class labels (i.e., anomalies) in our experiments.

5.2. **Base Line Approaches.** We have chosen 3 sets of baseline approaches. First, since we are using κ-nearest neighbor (κ-NN) approach along with clustering approach, we compare our method with the basic κ-NN approach. Second, we compare two subspace clustering approaches. They are *SCAD2* [10] and *K-means Entropy* [12] approaches. The reason behind using them as baseline approaches is that they have similarities in objective functions with our methods. So, a comparison with them will show the effectiveness of our algorithms from a subspace clustering perspective. Finally, we perform experiments using two multi-label methods and compare them to *pseudo-LSC.*



(a)

(b)

(c)

FIGURE 2. ROC Curves for (a) NASA ASRS Data Set (b) Reuters Data Set (c) 20 Newsgroups Data Set.

Methods	ASRS	Reuters	20 Newsgroups
pseudo-LSC	**0.637**	**0.821**	**0.874**
Pruned Set	0.469	0.56	0.60
MetaLabeler	0.58	0.762	0.766
κ-NN	0.552	0.585	0.698
SCAD2	0.482	0.533	0.643
K Means Entropy	0.47	0.538	0.657

TABLE 3. Area Under The ROC Curve Comparison Chart For Multi-Label Classification

They are *Pruned Set* [16] and *MetaLabeler* [18] approaches. Both these methods are state-of-the-art multi-label approaches and use *SVM* as their base classifiers. We, therefore, did not choose *SVM* as a baseline approach as it is already takes part in the comparison through these multi-label approaches. Also, it was not possible to used *SISC* [3] as it is applicable only for multi-class data, not multi-label data. Below we describe these 5 baseline approaches briefly.

5.2.1. *Basic κ-NN Approach.* In this approach, we find the nearest κ neighbors in the training set for each test point. Here κ is a user defined parameter. After finding the neighbors, we find how many of these neighbors belong to the t-th class. We perform this calculation for all the classes. We can then get the probability of the test point belonging to each of the classes by dividing the counts with κ. Finally, using these probabilities, for each class, we generate ROC curves and take their average to compare with our methods.

5.2.2. *K-Means Entropy.* This is another soft subspace clustering approach that we compare with *pseudo-LSC*. Its objective function has two components, the first one is based on dispersion and the second one is based on the negative entropy of cluster dimensions. Another difference between this approach and SCAD2 is that it is not fuzzy in nature. So, a training data point can belong to only a single cluster. The objective function that is minimized, as specified in [12] to generate the clusters, is as follows:

$$F(W,Z,\Lambda) = \sum_{l=1}^{k}\sum_{j=1}^{n}\sum_{i=1}^{m} w_{lj}\lambda_{li}D_{lij} + \gamma\sum_{l=1}^{k}\sum_{i=1}^{m}\lambda_{li}log(\lambda_{li}) \tag{5}$$

5.2.3. *SCAD2. SCAD2* [10] is a soft subspace clustering method with a different objective function than the *pseudo-LSC* method. This clustering method is also fuzzy in nature and can be considered the most basic form of fuzzy subspace clustering. As it does not consider any other factors during clustering except for dispersion. Its objective function has close resemblance to the first term of the *pseudo-LSC* objective function. As mentioned earlier, the reason we have used this method as benchmark is due to this similarity. The objective function of *SCAD2* is as follows:

$$F(W,Z,\Lambda) = \sum_{l=1}^{k}\sum_{j=1}^{n}\sum_{i=1}^{m} w_{lj}^{f}\lambda_{li}^{q}|x_{ij} - z_{li}|^2 \tag{6}$$

After performing this clustering using the same E-M formulation of *pseudo-LSC*, we use κ nearest clusters of each test point to calculate label probabilities.

5.2.4. *MetaLabeler.* This is a multi-label classification approach that learns a function from the data to the number of labels [18]. It involves two steps - i) constructing the meta data set and ii) learning a meta-model. Unlike our formulation of *pseudo-labels* in *pseudo-LSC*, the label of the meta data for this method is the number of labels for each instance in the raw data. There are three ways [18] that this learning can be done. We have applied the *Content-based MetaLabeler* to learn the mapping function from the features to the meta-labels (i.e., the number of class labels). As specified in [18], we consider the meta learning as a multi-class classification problem and use it in conjunction with *One-vs-Rest SVM.* We, therefore, train *T + 1 SVM* classifiers where *T* is the total number of class labels in the data set. Of them, one is a multi-class classifier and the rest are One-vs-Rest *SVM* classifiers for each of the classes. We then normalize the scores of the predicted labels and consider them as probabilities for generating *ROC curves.*

Methods	ASRS	Reuters	20 Newsgroups
pseudo-LSC	0.637	0.821	0.874
pseudo-LSC Without Chi Square	0.455	0.532	0.582

TABLE 4. Area Under The ROC Curve Comparison Chart For Chi Square Statistic

5.2.5. *Pruned Set.* The main goal of this algorithm is to transform the multi-label problem into a multi-class problem. In order to do so, *Pruned Set* [16] method finds frequently occurring sets of class labels. Each of these sets (or combinations) of class labels are considered as a distinct label. The benefit of using this approach is that, only those class label combinations that occur in the data set and the user does not need to consider an exponential amount of class label combinations. The user specifies parameters like what is the minimum count of a class label combination to consider it as frequent and the minimum size (i.e., class combinations having at least r class labels) of such sets or combinations.

At first, all data points with label combinations having sufficient count are added to an empty training set. This training set is then augmented with rejected data points having label combinations that are not sufficiently frequent. This is done by making multiple copies of the data points, only this time with subsets of the original label set. So, some data points may be duplicated during this training set generation process. This training set is then used to create an ensemble of *SVM* classifiers. We have also varied the number of retained label subsets to add to the training set and chose the best result to report.

5.3. **Evaluation Metric.** In all of our experiments, we use the *Area Under ROC Curve (AUC)* to measure the performance. For all the baseline approaches and our *pseudo-LSC* method, we generate each class label prediction as a probability. Then, for each class we generate an ROC curve based on these probabilities and the original class labels. After generating all the ROC curves, we take the average of them to generate a combined ROC curve. Finally, the area under this combined ROC curve is reported as output. This area can have a range from 0 to 1. The higher the AUC value, the better the performance of the algorithm.

5.4. **Results and Discussion.** As can be seen from Figure 2(a), *pseudo-LSC* performs much better than the baseline approaches. In Table 3, the *AUC* values for *pseudo-LSC* and all the baseline approaches are provided. With the *ASRS* data set, the *AUC* value for *pseudo-LSC* is 0.637. The closest performance is provided by the state-of-the-art *MetaLabeler* approach which is 0.58. Therefore, there is around 5%-8% increase in performance with our approaches.

Similar results can be found for *Reuters* and *20 Newsgroups* data sets. In Figure 2(b) and Figure 2(c), we provide these results. Just like the *ASRS* data set, *pseudo-LSC* provides much better results. For *Reuters* data set, our algorithm achieves AUC values of 0.821 and the nearest baseline approach value is 0.762. And, for *20 Newsgroups* data set, the AUC value achieved is 0.874 whereas, the nearest value is 0.766.

5.5. **Impact of Chi Square Statistic.** We have included the *Chi Square Statistic* in our objective function to achieve better performance by ensuring that more features have nonzero dimension weights as opposed to only a few features having nonzero weights and generating the clusters over a larger subset of dimensions. This increases the probability that a test point will have some of those nonzero features making the distance calculation meaningful. We have performed experiments to determine the impact of this component on the classification performance. To do so, we have removed this component from the objective function and performed the same experiments. We found that using it indeed increases the performance quite significantly. The objective function used in this case is given below.

$$F(W, Z, \Lambda) = \sum_{l=1}^{k} \sum_{j=1}^{n} \sum_{i=1}^{m} w_{lj}^{f} \lambda_{li}^{q} D_{lij} * (1 + Imp_l) \tag{7}$$

The results are provided in Table 4.

6. Conclusions

In this paper, we have presented *pseudo-LSC*, a semi-supervised text classification approaches based on fuzzy subspace clustering that considers the correlation among different class labels by generating pseudo labels during the clustering process. It provides a unified approach to perform classification on both multi-class and multi-label data. The experimental results on real world multi-labeled data sets like *ASRS*, *Reuters* and *20 Newsgroups*, have shown that *pseudo-LSC* outperforms *κ-NN*, *K-Means Entropy* based method, *SCAD2* and state-of-the-art multi-label text classification approaches like *Pruned Set* and *MetaLabeler* in classifying text data.

References

[1] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. *SIGMOD Rec.*, 28(2):61–72, 1999.

[2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec.*, 27(2):94–105, 1998.

[3] M. S. Ahmed and L. Khan. SISC: A text classification approach using semi supervised subspace clustering. *DDDM '09: The 3rd International Workshop on Domain Driven Data Mining in conjunction with ICDM 2009*, Dec. 2009.

[4] M. S. Ahmed, L. Khan, N. Oza, and M. Rajeswari. Multi-label ASRS dataset classification using semi-supervised subspace clustering. *In Conference on Intelligent Data Understanding (CIDU) 2010*, September 2010.

[5] M. B. Basu; and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *ICML*, pages 81–88, 2004.

[6] F. Bavaud. Euclidean distances, soft and spectral clustering on weighted graphs. In *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part I*, ECML PKDD'10, pages 103–118, Berlin, Heidelberg, 2010. Springer-Verlag.

[7] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637, Dec. 2005.

[8] C.-H. Cheng, A. W. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 84–93, New York, NY, USA, 1999. ACM.

[9] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier. Regret analysis for performance metrics in multi-label classification: the case of hamming and subset zero-one loss. In *Proceedings of the*

2010 European conference on Machine learning and knowledge discovery in databases: Part I, ECML PKDD'10, pages 280–295, Berlin, Heidelberg, 2010. Springer-Verlag.

[10] H. Frigui and O. Nasraoui. Unsupervised learning of prototypes and attribute weights. *Pattern Recognition*, 37(3):567 – 581, 2004.

[11] S. Goil, H. Nagesh, and A. Choudhary. Mafia: Efficient and scalable subspace clustering for very large data sets. *Technical Report CPDC-TR-9906-010, Northwest Univ.*, 1999.

[12] L. Jing, M. K. Ng, and J. Zhexue Huang. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Trans. Knowl. Data Eng.*, 19(8):1026–1041, 2007.

[13] F. T. Liu, K. M. Ting, and Z.-H. Zhou. On detecting clustered anomalies using sciforest. In *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part II*, ECML PKDD'10, pages 274–290, Berlin, Heidelberg, 2010. Springer-Verlag.

[14] G. Liu, J. Li, K. Sim, and L. Wong. Distance based subspace clustering with flexible dimension partitioning. In *IEEE 23rd International Conference on Data Engineering (ICDE)*, pages 1250–1254, April 2007.

[15] M. Masud, J. Gao, L. Khan, J. Han, and B. Thuraisingham. A practical approach to classify evolving data streams: Training with limited amount of labeled data. In *Eighth IEEE International Conference on Data Mining (ICDM)*, pages 929–934, Dec. 2008.

[16] J. Read, B. Pfahringer, and G. Holmes. Multi-label classification using ensembles of pruned sets. In *Eighth IEEE International Conference on Data Mining (ICDM)*, pages 995–1000, Dec. 2008.

[17] J. Struyf and S. Džeroski. Clustering trees with instance level constraints. In *ECML '07: Proceedings of the 18th European conference on Machine Learning*, pages 359–370, Berlin, Heidelberg, 2007. Springer-Verlag.

[18] L. Tang, S. Rajan, and V. K. Narayanan. Large scale multi-label classification via metalabeler. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 211–220, New York, NY, USA, 2009. ACM.

[19] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 104, New York, NY, USA, 2004. ACM.

[20] G. Tsoumakas and I. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *ECML '07: Proceedings of the 18th European conference on Machine Learning*, pages 406–417, Berlin, Heidelberg, 2007. Springer-Verlag.

[21] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. *In Advances in Neural Information Processing Systems 15. Cambridge: MIT Press.*, 2003.

[22] P. Wang, C. Domeniconi, and K. B. Laskey. Nonparametric bayesian clustering ensembles. In *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part III*, ECML PKDD'10, pages 435–450, Berlin, Heidelberg, 2010. Springer-Verlag.

[23] K. Yip, D. Cheung, and M. Ng. Harp: a practical projected clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1387–1397, Nov. 2004.

[24] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–265, New York, NY, USA, 2005. ACM.

[25] M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038 – 2048, 2007.