Global Journal of Computer Science and Technology: C
Software & Data Engineering
Volume 14 Issue 1 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

A Framework for Context-Aware Semi Supervised Learning

By Vijaya Geeta Dharmavaram & Shashi Mogalla

GITAM University, India

Abstract- Supervised learning techniques require large number of labeled examples to build a classifier which is often difficult and expensive to collect. Unsupervised learning techniques, even though do not require labeled examples often form clusters regardless of the intended purpose or context. The authors proposes a semi supervised learning framework that leverages the large number of unlabeled examples in addition to limited number of labeled examples to form clusters as per the context. This framework also supports the development of semi supervised classifier based on the proximity of unknown example to the clusters so formed. The authors proposes a new algorithm namely "Semi Supervised Relevance Feature Estimation", (SFRE), to identify the relevant features along with their significance weightages which is integrated with the proposed framework.

Keywords: context – aware, semi supervised learning, feature relevance, subspace clustering, discriminant analysis.

GJCST-C Classification : I.2.6

A Framework for Context-Aware Semi Supervised Learning

Vijaya Geeta Dharmavaram [α] & Shashi Mogalla [σ]

Abstract- Supervised learning techniques require large number of labeled examples to build a classifier which is often difficult and expensive to collect. Unsupervised learning techniques, even though do not require labeled examples often form clusters regardless of the intended purpose or context. The authors proposes a semi supervised learning framework that leverages the large number of unlabeled examples in addition to limited number of labeled examples to form clusters as per the context. This framework also supports the development of semi supervised classifier based on the proximity of unknown example to the clusters so formed. The authors proposes a new algorithm namely "Semi Supervised Relevance Feature Estimation", (SFRE), to identify the relevant features along with their significance weightages which is integrated with the proposed framework. Experiments conducted on the benchmark datasets from UCI gave results which are very promising and consistent even with lesser number of labeled examples.

Keywords: context – aware, semi supervised learning, feature relevance, subspace clustering, discriminant analysis.

I. Introduction

Machine learning techniques are being adopted by various applications from different domains to build predictive models. These techniques are broadly classified as supervised learning and unsupervised learning based on the availability of class labels to build the model. Supervised learning methods require labeled data to build a classifier model that predicts the class labels of unknown examples based on the information available in the form of class labels. However, it is usually very expensive and time-consuming process to collect the labeled data (Han et al., 2011). Even in domains with abundance of unlabeled data, labeled data are usually scarce and would require some effort to collect such data. However, to build classifier with better generalized accuracy, large number of labeled data is required, more so for datasets with high dimensionality - one of the problems associated with curse of dimensionality (Ramona et. Al.,2012).

Accordingly, it is believed that with fixed number of labeled examples, the predictive power of the classifier decreases with the increase in number of dimensions thus requiring larger number of labeled examples for building classifier (Advani, 2011).

In unsupervised learning methods such as clustering, unlabeled data, if available in abundance, suffice to extract hidden patterns of knowledge from a given dataset. Traditional clustering algorithms take into account the entire feature space to partition the datasets into clusters such that there is homogeneity among the instances within a cluster. The proximity between the instances in the cluster is measured in terms of distance function. However, with the increase in dimensions, the distance measures employed in the clustering algorithm becomes insignificant and clusters so produced will be meaningless. Hence clustering will full feature space, especially when the number of dimensions are large, may not produce good clusters.

Finding the subset of feature space to produce meaningful clusters comes under the purview of subspace clustering. Subspace clustering focuses on finding a subset of features or a smaller set of transformed features with an aim to define cluster-able object spaces (Han et al., 2011; Sim et al., 2013). In high dimensional datasets due to exponentially large number of subsets of the feature set, subspace clustering techniques have to eliminate enormous possibilities before identifying the appropriate feature space that contain intrinsically significant clusters (Han et al., 2011). The basic research in subspace clustering falls into unsupervised learning as it tries to identify clusters based on the distribution of objects in various feature sub-spaces irrespective of the class labels of the objects. The clusters thus formed may be meaningful but may not be relevant to the intended purpose or context. For instance, the census data is described in terms of different features like social, economic, education, health, etc.,. However, it needs to be clustered in groups depending on the purpose of the data analysis. Features corresponding to social backwardness and eco-nomic status is used to identify the welfare schemes to be adopted, whereas features corr-espo-nding to place of living, commutability, etc., are used to decide the location of new amenities centers. In both the cases, features used and their relative significance will vary with the context or purpose thus requiring the clustering algorithm to give proper emphasis to appropriate features in accordance with the context for which the patterns are extracted. Such clustering is referred to as Context-Aware Subspace Clustering.

Author α: Associate Professor, Department of Operations, GITAM Institute of Management, GITAM University.
e-mail: vijayageeta@gitam.edu
Author σ: Professor, Department of Computer Science & Systems Engineering, AU College of Engineering, Andhra University.
e-mail: smogalla2000@yahoo.com

Context-aware-subspace clustering aims to find appropriate feature subspace for a given context represented in the form of class labels of a few labeled examples which are consistent with a large collection of unlabeled examples belonging to the same dataset. To the best of our knowledge, not much research was published in support of feature selection algorithms making use of combination of labeled as well as unlabeled examples. Hence semi supervised feature selection algorithms are needed to be developed for formation of context-aware clusters in domains having only limited examples labeled and the rest being left unlabeled.

Semi Supervised Learning which is an integration of supervised and unsupervised learning; makes use of both labeled and unlabeled examples to build a model (Zhu and Goldberg, 2009). Semi supervised learning has two forms namely semi supervised classification and semi supervised clustering. Semi supervised classification uses both labeled and unlabeled data to build the classifier. Using the limited number of labeled data, probable class labels for the unlabeled data is derived which in turn is added to the pool of labeled data thus increasing the number of labeled examples (Han et al., 2011). The basic assumption in this technique is that the similar data will have same class labels (cluster assumption) (Chapelle et al., 2006; Wang et al., 2012). Different methods like self training, co-training, generative probabilistic models, graph based and support vector machines are used for semi supervised classification (Zhu, 2008). In semi supervised clustering, a large set of unlabeled data is accompanied by a small amount of domain knowledge in the form of either class labels or pairwise constraints (must-link and cannot-link) (Grira et al., 2004; Ding et al., 2012). This domain knowledge is used to guide the clustering of unlabeled data so that the intra-cluster similarities are maximized and inter-cluster similarities are minimized and there exist consistency between the partition and the available knowledge (Gao et al., 2006).

Based on the above arguments, authors proposes context-aware semi supervised subspace clustering framework which leverages the domain knowledge in terms of class labels for at least some of the examples (if labeled examples are expensive) in order to estimate the suitability of the features to the intended cluster solution. Proper selection of features and their relative significance is essential in producing context-aware clusters which are probably uni-class clusters. Uni-class clusters contain all or majority of the elements belonging to same class label which is reflected in terms of cluster purity. The clustering framework is further extended to build a classifier which is referred to as semi supervised classifier that requires minimum information for prediction. The authors also proposes 'Semi Supervised Feature Relevance Estimation', (SFRE), algorithm to estimate the relevant features and their relative significance in terms of weights that define appropriate subspaces for different targets/context. The framework was tested on a few benchmark datasets from UCI repository which has given promising results.

II. Related Work

Researchers in the past came up with different methods for semi supervised learning. One popular approach is constrained based clustering. Constraint based methods uses pairwise constraints in the form of must-link and cannot-link that guides the clustering process to partition the data in a way that do not violate these constraints (Wagstaff et al., 2001; Basu et al., 2004; Lu and Leen, 2004). Recently Xiong et al., (2014) proposed an iterative based active learning approach to select pairwise constraints for semi supervised clustering. It uses the concept of neighbourhood that contains labeled examples of different clusters based on pairwise constraints. The uncertainty associated with each point's neighbor is resolved through queries. However, repeated clustering is required with growing list of constraints.

Another popular approach for semi supervised clustering is distance based techniques which is based on the cluster assumption. Yin and Hu (2011) proposed semi supervised clustering algorithm using adaptive distance metric learning where clustering and distance metric learning are performed simultaneously. The clustering results are used to learn the distance metric and the data is projected into a low dimensional space such that data seperability is maximized. Gao et al., (2006) focused on semi supervised clustering in terms of features rather than examples. It addresses the problem where labeled and unlabeled dataset have different feature set with few common features.

In terms of feature selection, Padmaja et al., (2010) proposed a dimensionality reduction approach that estimates the significance of features based on the fractal dimensions and accordingly selects a subset of features that are essential to capture the characteristics of the dataset. The algorithm detects all types of correlations among features to identify the essential features after eliminating the redundant and irrelevant features. Kernel based feature selection was also explored by a few researchers (Wang, 2008; Ramona et al., 2012). Clustering based feature selection for classification was proposed by Song et al., (2013) where features are clustered based on graph theoretic clustering method.

Research on feature weighting and ranking concentrated more on supervised learning (Eick et al., 2006; Al-Harbi and Rayward-Smith, 2006; Zhao and Qu, 2009). Most of these research studies initially weigh the features by using some random guess or equal weights. These initial weights are then adjusted accordingly. Such approach may take much time to arrive at the final optimum weights if the initial guess is not appropriate.

This paper deals with semi supervised learning methods with wrapper based feature selection method that uses discriminant analysis results to initialize the weights. These weights are adjusted accordingly in a stepwise refinement process using both labeled and unlabeled examples. The proposed framework is used to develop a classifier and a pertinent cluster solution.

III. Context-aware Semi Supervised Subspace Clustering Framework

Given a dataset with '*l*' number of labeled and '*u*' number of unlabeled examples such that *l*<<*u*, the proposed context-aware semi supervised subspace clustering framework has a four phase architecture as shown in Fig. 1. The phases are explained in the subsequent paragraphs.

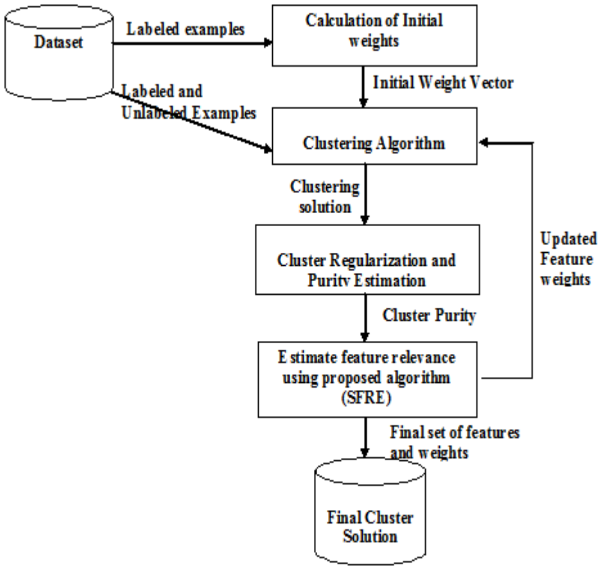
Dataset
Labeled examples
Calculation of Initial weights
Initial Weight Vector
Labeled and Unlabeled Examples
Clustering Algorithm
Clustering solution
Cluster Regularization and Puritv Estimation
Updated Feature weights
Cluster Purity
Estimate feature relevance using proposed algorithm (SFRE)
Final set of features and weights
Final Cluster Solution

Figure 1 : Architecture of the Proposed Context-Aware Semi Supervised Subspace Clustering Framework

a) Initial weights calculation

A dataset may be clustered in multiple ways by appropriately selecting a subset of features /attributes depending on the purpose. Hence to produce clusters conforming to a particular purpose or context, weights must be given to features that depict the importance of the feature. Researchers in the past initially start with a guess/random weights or equal weights to the feature and proceeds further to determine the more acceptable weights. Instead of starting with some arbitrary values, it is proposed to use the information from the available labeled data to initialize the weights which can be adjusted later. Authors thus propose usage of discriminant analysis that finds the relationship between the independent features (predictors) and the dependent feature (class label), to initialize the feature weights.

Discriminant analysis is a method that is used to predict categorical value from a given set of independent feature. It assumes the independent features to be normally distributed. The linear equation of Discriminant analysis is (Equation 1)

$$D=V_1X_1 + V_2X_2 + V_3X_3 + \ldots.. + V_iX_i + a \qquad 1$$

Where

D=Discriminant Score

V_i= the discriminant coefficient or weight of ith feature

X_i= Value of ith feature

a = a constant

Discriminant analysis thus identifies the relevant features and its coefficients reflect the relevancy of the feature. The outcome of the discriminant analysis in terms of coefficients is normalized and is used as initial weights for developing binary cluster solution where as development of multi-class cluster solution involves integration of results given through multiple discriminant functions. The proposed framework use potency index as per the approach given in Dharmavaram and Mogalla (2013) for determining the initial weights of various features based on the labeled examples in case of multi-class datasets.

b) Clustering Algorithm

The initial weight vector is used to form the initial cluster solution by using any partitional clustering algorithm. The authors have chosen K-means algorithm for its simplicity and computational efficiency to deal with numerical features. While dealing with datasets described in terms of numerical attributes, generally K-means algorithm employs Euclidean distance to compute the distance from each data point to the cluster centroid. Euclidean distance assumes that all the features are equally important while forming the clusters. However, as discussed previously, weights of the feature will determine the relevancy of the feature in forming the desired cluster solution and accordingly Weighted Euclidean Distance metric is used for distance calculation which has the following equation (Equation 2):

$$\text{dw } (x_i, x_j) = \sqrt{\sum_{m=1}^{p} w_m (x_{im} - x_{jm})^2} \qquad 2$$

$$\text{where } \sum_{m=1}^{p} w_m = 1$$

where w_m indicates the weight of the mth feature. If the significance of the feature is more, its weight will be more. The weight of an irrelevant feature can be set to zero.

For clustering, the number of clusters, K, is taken to be more than the number of classes. Larger values of K results in formation of large number of small uni-class clusters and hence, multiple clusters are associated with a single class. Each of these clusters represents a neighbourhood reflecting the natural grouping or profiling within the same class examples. However, clusters formed with a very few labeled examples may not be enough to define the purity of the clusters. The presence of an outlier labeled example will also alter/hamper the definition of the cluster. Clusters defined in terms of such labeled examples may lead to overfitting that could result in poor predictive performance for unseen examples. The cluster solution thus needs to be regularized to take care of the distribution of the labeled as well as unlabeled examples in the cluster space. The regularization procedure takes into account the presence of unlabeled examples in the vicinity of labeled examples in a cluster while estimating the purity of a cluster. The proximity measure between the labeled and unlabeled examples is used to define the probable class labels of the unlabeled examples. Cluster purity is thus defined in terms of true class labels for the labeled examples and probable class labels for the unlabeled examples.

c) Cluster Regularization and Estimation of Semi supervised Cluster Purity

Given a dataset D = L ⋃ U consisting set of '*l*' labeled examples, L and '*u*' unlabeled examples, U, where *l*<<*u*. Each labeled example, n, in L has a label y_n ∈ Q, where Q is a set of 'q' number of classes. The author devised a new metric called 'Cluster Concurrence' to measure the purity of the clusters based on the probabilistic class labels of unlabeled examples and true class labels of labeled examples.

In case of labeled examples, true class labels are taken from the dataset. In the case of unlabeled examples, probabilistic class labels $\hat{y}$, are generated in terms of the class labels of the labeled examples in their neighbourhood, based on the weighted Euclidean distance between the labeled and the unlabeled examples.

The cluster concurrence is estimated for each cluster based on the agreement of the members of the cluster towards a particular class label and hence reflects the uni-class property of a cluster. In order to estimate the cluster concurrence of k^{th} cluster, the support, S_{kj}, available for each class, *j*, in that cluster is aggregated as shown in Equation 3.

$$S_{kj} = \frac{1}{|C_k|} \sum_{n \in C_k} M \times P_j(n) \qquad 3$$

Where

$P_j(n)$ indicates the probability of the example *n* belonging to the class *j*

$|C_k|$ is the cardinality of the cluster *k,* i.e., the number of examples that are assigned to cluster C_k.

$$M = \begin{cases} 1 & if\ j = \underbrace{argmax}_{i} \{P_i(n)\} \\ 0 & otherwise \end{cases}$$

The binary term *M* acts as a deciding factor to indicate whether the example contributes to the support of class *j* or not. It may be noted that each example, whether labeled or unlabeled, contributes to the support of only one class: the unlabeled example support the class with the maximum probability, while the labeled example naturally support one and only one true class label.

$P_j(n)$ is calculated as per the equation given below (Equation 4) where *d(i,n)* is the weighted Euclidean distance between *i* and *n*.

$$P_j(n) = \begin{cases} 1 & if\ n \in L\ and\ y_n = j \\ \dfrac{\sum_{i \in L \cap C_k\ and\ y_i = j} \frac{1}{d(i,n)^2}}{\sum_{z \in L \cap C_k} \frac{1}{d(z,n)^2}} & if\ n\ \in U \cap C_k \\ 0 & otherwise \end{cases} \qquad 4$$

The predicted label of an unlabeled example, *t*, is the label for which the probability is maximum.

The cluster concurrence of K^{th} cluster is estimated as:

$$CC_k = max_j \{S_{kj}\}$$

Overall cluster purity of the cluster solution is taken as the weighted sum of individual cluster concurrences and is given below (Equation 5)

$$CP = \sum_{t=1}^{K} \frac{|C_t|}{|D|} CC_t \qquad 5$$

d) Estimating feature relevance (SFRE algorithm)

Though discriminant analysis is a proven technique for estimating the relevance of various features in a dataset, for the purpose of semi supervised subspace clustering, it can only make use of the limited labeled examples. In order to estimate the feature relevance more accurately leveraging abundantly available unlabeled examples in the dataset, the author proposes a new algorithm namely Semi supervised Feature Relevance Estimation (SFRE) algorithm, that improves the preliminary estimates made by discriminant analysis and thereby contributes to context-aware semi supervised subspace clustering.

The new algorithm, SFRE is guided by cluster purity estimated in terms of labeled as well as unlabeled examples belonging to various feature subspaces. The algorithm accepts the dataset D that includes L and U, initial cluster purity and the outcome of discriminant analysis as initial weights for formation of initial weight vector as input. The output of the algorithm is accurate relevance estimates of the feature set referred to as weight vector that defines the feature subspace for the given purpose indicated through class labels.

The cluster purity obtained by the initial weights is assigned to current cluster purity as initialization step, after which the algorithm executes the following three steps iteratively:

Step 1: Finding Relevant Features
Step 2: Updating Weights
Step 3: Check for convergence

In the first step, each feature in the feature set is checked for its relevance. Taking one feature at a time, clusters are formed without that feature and cluster purity is estimated. If there is a decrease in cluster purity when compared to the current cluster purity, it indicates that the absence of the feature has resulted in the loss in purity and hence it is marked as relevant feature and its relevance increment is calculated based on the proportionate difference in the cluster purity estimated with and without the feature. If there is increase in the cluster purity when compared to the current cluster purity, it indicates that the absence of the feature has resulted in the gain in purity and hence it is marked as irrelevant feature. The outcome of this step is to mark each feature either relevant or not and to estimate the relevance increment for those relevant features.

In the second step, based on the relevance marking, the weights are adjusted such that weights of the relevant features are incremented in accordance with the relevance increment calculated in step 1. The weights of those features marked irrelevant, are made zero and finally the weight vector is normalized to sum up to 1.

In the final step, clusters are formed with the adjusted weights to judge the final solution. The new cluster purity obtained from clusters formed with updated weights and features is compared with the current cluster purity. If there is improvement in the cluster purity, the new weights are accepted and the new cluster purity is taken as the current cluster purity for comparison in the next iteration. The steps are repeated till there is not much significant improvement in the cluster purity. To change the order in which the features are selected in the subsequent iterations; features are randomly selected without replacement. This supports in avoiding any overlap or correlation in the features and to avoid local maxima.

e) *Formal listing of Proposed Algorithm (SFRE)*

Let *CPcurr* be the cluster purity estimated for the initial cluster solution then stepwise refinement in weights proceeds as follows:

```
Step 1: For each feature $x$, randomly selected without replacement from the feature set $F$
Perform K-means without the feature $x$ by appropriately normalizing the weight vector
Estimate Cluster Purity $CP_{F-x}$
If $CP_{F-x} < CP_{curr}$ then
    $x$ is relevant
    calculate relevance increment, $Rel_x = \frac{CPcurr - CPF-x}{CP_{curr}}$
Else
    x is not relevant

Step 2: Increase the weight of each relevant feature $x$, $W_x = W_x (1 + Rel_x)$
For each irrelevant feature $x$, $W_x = 0$
Normalise the weight vector

Step 3: Perform K-means with adjusted weights
Estimate the cluster purity $CP_{new}$
    If $CP_{new} > CP_{curr}$
        Accept new weights
        $CP_{curr} = CP_{new}$
    Perform above steps till there is no improvement in the cluster purity.
```

The final cluster solution thus formed consists of context-aware clusters with final set of relevant features and weights.

IV. Semi Supervised Classification Framework

The above proposed clustering framework is further extended to support classification. To build the classifier model, the dataset is first divided into two sets: Training Set, *TR* and Test Set, *TS.* The test set contains only labeled examples as it is required to estimate the error based on the true class labels. However, the training set contains both labeled and unlabeled examples as the semi supervised learning makes use of unlabeled examples based on their vicinity to labeled examples. The model building process has three phases:

Phase 1: Training the Classifier: The training set, TR, is clustered as per the framework proposed in the previous section. The final clusters formed are checked for their decisiveness in predicting the class label. If a cluster is decisive, the cluster is labeled with the majority class label within the cluster otherwise the cluster is not labeled and the details of the cluster is stored in order to apply weighted nearest neighbour classification while classifying unknown examples.

Phase 2: Testing the Model: The trained classifier is tested for acceptance by estimating its accuracy in predicting the class labels of the test examples. The predicted class labels are compared with the true class labels to calculate the global error. If the global error is tolerable, the model is finalized otherwise the model is re-trained by changing the number of clusters and the test process is repeated.

Phase 3: Using the model for Classification: Once the final classifier is built, class labels of unknown examples are predicted as per the classification rules of the model.

a) *Estimating the decisiveness of a cluster*

A cluster which is considered as a uni-class cluster has most of its members agreeing on a particular class. Such cluster also referred to as decisive cluster, is labeled with the (majority) class label which has the maximum support of the examples in that cluster.

However, in the presence of overlapping examples or outliers, the examples in a cluster may not strongly agree on a particular class and such cluster is not considered as uni-class / decisive cluster and is not labeled as they are considered as indecisive cluster. The final cluster solution formed in the training phase contains K clusters with each cluster containing examples belonging to one or more classes. The support of a class in a cluster S_{kj}, is estimated in terms of true class labels of labeled examples and the predicted (probabilistic) class labels of unlabeled examples in the kth cluster. In a given cluster, the difference between the support available for majority class and its competing class reflects the decisiveness of the cluster in concurrence with the majority class. For this purpose, the authors propose a metric referred to as 'Purity Margin' which is measured for each cluster and is compared against purity threshold as detailed below.

b) Purity Threshold of the cluster

The 'Purity Threshold', *PT*, of a cluster, C_k, PT_k is set as the minimum difference or margin, to be imposed between two competing classes in a cluster, for it to be considered as the decisive cluster. The purity threshold is estimated as a pre-defined fraction (λ) of the product of cluster concurrence CC_k and the number of classes in the dataset. In a dataset with *q* classes, purity threshold PT_k, for a cluster C_k is calculated as (Equation 6)

$$PT_k = \lambda . CC_k . q \qquad 6$$

Various experiments conducted on the value of λ shows that 0.1 which indicates 10% of support value, is a good measure to get optimum purity threshold.

c) Purity Margin of the cluster

The purity margin measures the difference between the maximum support of a class in a cluster and the support of its immediate competitor class. Larger the margin, more pure the cluster is. Intuitively it is taken that it should be greater than or equal to the purity threshold.

For a cluster C_k, the purity margin $PM(C_k)$ is calculated as (Equation 7)

$$PM(C_k) = CC_k - S_{kp} \text{ where } p \text{ is the competing class.} \qquad 7$$

d) Decisive cluster

A cluster C_k, is considered to be a uni-class or a decisive cluster, if $PM(C_k) \geq PT_k$ else it is considered as indecisive cluster. The decisive cluster is labeled with the majority class label i.e., the class label that has maximum support of the examples in the cluster, over all classes in the cluster. The indecisive cluster is left unlabeled and the details of the cluster including the predicted labels of unlabeled examples are stored to apply the weighted nearest neighbour classification while classifying any unknown / test example.

e) Hybrid Model for Classification

The authors propose a hybridization of model-based classification and instance-based classification for classifying any unknown / test example based on whether it is compatible to decisive cluster or an indecisive cluster.

Let the cluster, C_k be the most compatible cluster for unknown example *x*:

- If the cluster, C_k, is decisive then
- Assign the cluster label, y_{C_k} to the example *x*.
- If the cluster is indecisive then
- Apply weighted nearest neighbor classification to predict the class label of *x*.

f) Finding the most compatible cluster for unknown / test example

Consider a set of clusters $C=\{C_1,C_2,\ldots,C_K\}$ with centroids as $c=\{c_1,c_2,\ldots c_K\}$. Weighted Euclidean distances are calculated between unknown / test example, *t*, and each centroid, c_i. The cluster C_k, which has the minimum distance among all the clusters is said to be the most compatible cluster for the example, t. Mathematically, it may be expressed as (Equation 8)

$$k = \underbrace{argmin}_{i} \{d(t,c_i)\} \qquad 8$$

Hybrid model for classification is applied on the value of *k* as discussed earlier.

g) Weighted Nearest Neighbour Classification

If the cluster C_k, is an indecisive cluster, the cluster contains examples from more than one class. In other words, the cluster contains multiple neighborhoods dominated by different class labels. Hence, the proximity of the unknown / test example, *'t'*, to each class must be measured. The closer the example, *'t'*, is to the neighborhood dominated by particular class label, it is more likely to share the same class label of its neighbors (Cluster Assumption). Accordingly, all the members of the most compatible cluster C_k, are considered as neighbors with weights assigned in the inverse proportion of their squared distance to the test example. The proximity of the example, *t*, to a class label, *p*, denoted by W_{tp}, is estimated by aggregating the weights of the members belonging to that particular class. Mathematically it may be expressed as (Equation 9)

$$W_{tp} = \sum\nolimits_{i \in C_k \, and \;\, \hat{y}_i = p} \frac{1}{d(t,i)^2} \qquad 9$$

where *d(t,i)* is the Euclidean distance between *t* and *i*.

This proximity estimate will ensure that the examples that are far (possibly an outlier) from the test example has less impact on prediction compared to the ones that are closer by.

The unknown / test example is assigned the class label for which the proximity is maximum (Equation 10).

$$\hat{y}_t = \underbrace{argmax}_{p} \left(W_{tp}\right) \forall\, p \in Q_k, \textit{the set of classes in the cluster } C_k. \qquad 10$$

V. Experiments and Results

a) Experimental Setup

The proposed model was implemented on Intel Pentium dual core processor with 3GB of DDR2 667 MHz memory and coded using .NET framework. SPSS statistic tool is used for performing discriminant analysis.

Experiments were conducted on benchmark datasets obtained from UCI repository and one dataset from SPSS Inc. to test the performance of the proposed framework. Five binary datasets and six multi-class datasets were used in the experimentation as shown in table 1.

Table 1 : Description of Datasets

S.No.	Dataset	#Instances	# Attributes	Class
1.	Breast Cancer	683	9	2
2.	Credit	690	15	2
3.	Ionosphere	351	34	2
4.	Pima	768	8	2
5.	Bankloan	700	8	2
6.	Ecoli	336	7	8
7.	Glass	214	9	10
8.	Iris	150	4	3
9.	Wine	178	13	3
10.	Yeast	1484	8	10
11.	Zoo	101	7	7

The labels from some of the examples were purposefully removed to consider them as unlabeled examples and the percentage of labeled examples were varied from 75% to 15% to observe the change in the model's performance with decreasing percentage of labeled examples.

In case of semi-supervised classification the dataset is split into training dataset and test dataset in the ratio of 75:25. Class labels were removed from appropriate number of examples in the training set to test the suitability of the model as a semi supervised classifier. The labeled and unlabeled examples in the training set are taken randomly in varied percentages, in the ratio of 75:25, 50:50, 25:75 and 15:85. The proposed model is trained on the training dataset and the accuracy of the model is estimated based on the test dataset by comparing the true class labels of test dataset with the predicted class labels of the model.

For binary class datasets, experiments were conducted with 100% labeled examples to assess the performance of the framework when all the examples in the datasets are labeled. However availability of labeled examples upto 100% does not call for semi supervised learning. The case with 100% labeled examples was demonstrated only to prove that the proposed method can handle datasets having less labeled examples in the similar way with datasets having 100% labeled maintaining consistently high performance. The complexity of cluster regularization and estimation of cluster concurrence and purity margin for development of hybrid classifier are not required for datasets having near 100% labeled examples and they may be better processed by an appropriate supervised learning algorithm. The performance of the model for multi-class datasets was analysed starting from 75%.

In both the cases of clustering and classification, discriminant analysis is performed using SPSS statistics tool on the labeled examples in the datasets to produce the discriminant function(s). For binary class datasets, discriminant coefficients, and for multi-class datasets, potency index values are used to get the initial weights of the features in the dataset, which are referred to as initial weight vector.

b) Results

In case of Semi Supervised Subspace Clustering, the cluster purity was estimated based on the cluster concurrence and the number of relevant features identified for the benchmark datasets are tabulated in table 2 and 3. The change in cluster purity with varied percentage of labeled examples is shown in figure 2 and figure 3.

Table 2 : Cluster Purity of Context-Aware Semi Supervised Subspace clustering – Binary Class Datasets

S.No	Dataset	100%	75 %	50 %	25 %	15 %
1	Bcancer	97.24	96.94	95.76	96.34	96.29
2	Credit	86.52	86.26	85.63	85.77	85.78
3	Ionosphere	90.56	88.23	90.21	88.24	88.56
4	Pima	77.65	76.14	75.86	76.79	77.90
5	Bankloan	80.0	77.92	76.91	77.39	73.94

Table 3 : Cluster Purity of Context-Aware Semi Supervised Subspace clustering – Binary Class Datasets

S.No	Dataset	75 %	50 %	25 %	15 %
1	Ecoli	86.24	82.90	83.82	82.81
2	Glass	72.31	73.72	72.76	69.01
3	Iris	96.64	96.64	95.30	95.92
4	Wine	96.61	97.74	96.61	95.44
5	Yeast	58.04	57.90	56.30	56.10
6	Zoo	84.81	97.0	92.0	65*

**The size of the zoo dataset is 101. As 15% of the examples could not cover all the seven classes, the error has increased unnaturally.*

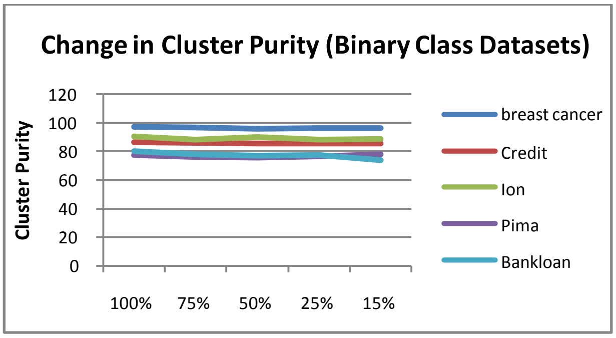

Figure 2 : Changes in cluster purity for Binary Class Dataset

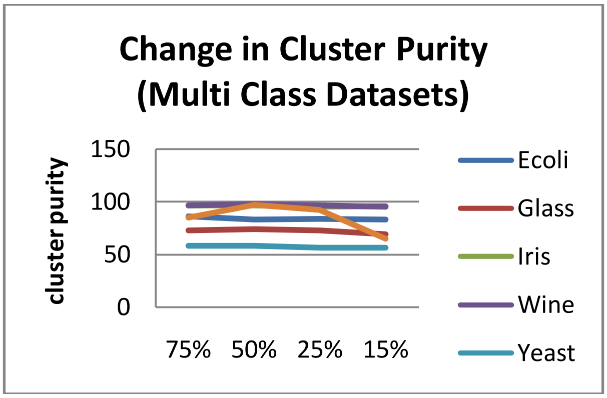

Figure 3 : Changes in cluster purity for Multi Class Dataset

From Fig. 2 and Fig. 3, it is observed that the proposed model has consistent performance in term of cluster purity and not much change is observed with variation in percentage of labeled example. Only in the case of Zoo dataset, there has been huge decline in the cluster purity when there are few labeled examples. This is attributed to the fact, that number of examples in zoo dataset are only 101 and 15% of labeled data is very less compared to number of class labels and may not capture representatives from all the 7 class examples.

In case of Semi Supervised Classification, the training sets of benchmark datasets are used to build the classifier and the accuracy of the classifier is tested on the test set where the predicted class labels are compared with true class labels of the test examples. These test results given in terms of accuracy is compared with the proven classifier models. The models considered for comparison are Weka implementation of C4.5 and an ensemble method, Bagging. Only one ensemble method is considered for comparison as all the other ensemble methods has similar performance on most of the datasets (Tan et al., 2006: Table 5.5). The results are tabulated in table 4 and 5 and a sample comparison graphs for a dataset in binary and multiple class is shown in Fig. 4 and Fig.5.

Table 4 : Comparison table for Semi Supervised Classification – Binary Class Datasets

	Ensemble – Bagging					C4.5					Proposed Model				
Dataset	100	75	50	25	15	100	75	50	25	15	100	75	50	25	15
Breast Cancer	97.56	95.21	95.20	95.09	86.82	94.84	95.24	94.03	91.70	91.61	**97.60**	**97.56**	**96.68**	**95.74**	**95.74**
Credit	**92.02**	81.08	80.41	79.02	77.20	86.37	83.78	**80.47**	80.0	79.0	86.52	**85.21**	80.28	**80.92**	**79.51**
Ionosphere	94.01	89.47	88.31	86.84	80.51	**99.0**	**92.20**	**90.78**	**88.31**	**84.21**	91.76	88.0	86.6	86.64	84.0
Pima	**88.93**	76.53	74.86	**75.69**	**71.80**	84.11	71.82	71.50	70.94	70.39	77.77	**76.83**	**76.27**	75.0	70.05
Bank loan	**85.23**	**76.40**	74.0	72.0	72.0	90.0	73.93	72.34	72.0	70.0	78.11	74.63	**74.62**	**74.09**	**72.27**

Table 5 : Comparison table for Semi Supervised Classification – Multi-class Datasets

	Ensemble – Bagging				C4.5				Proposed Model			
Dataset	75	50	25	15	75	50	25	15	75	50	25	15
Ecoli	74.66	74.02	70.66	56	76	**75.32**	70.66	54.66	**76.5**	75.3	**73.9**	**69.86**
Glass	61.66	**62.71**	49.15	49.15	**62.71**	61.66	49.15	45.76	60.3	57.72	**56.8**	**57.7**
Iris	**100**	94.11	94.11	58.82	97.11	94.11	91.17	76.47	96.96	**96.96**	**96.5**	**93.93**
Wine	91.66	91.66	88.88	86.11	91.66	91.66	88.88	83.33	**97.14**	**94.2**	**94.2**	**91.4**
Yeast	**58.71**	54.15	51.87	45.6	52.9	52.53	51.44	51.74	55.52	**54.71**	**52.5**	**51.21**
Zoo	82.14	77.77	75	53.57	82.14	78.57	77.77	64.28	88.88	**85.18**	**81.48**	**72.22**

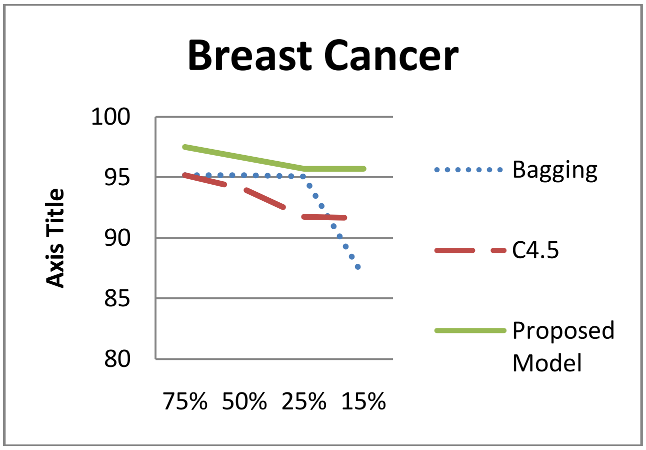

Figure 4 : Comparison Graphs for Breast Cancer (binary class)

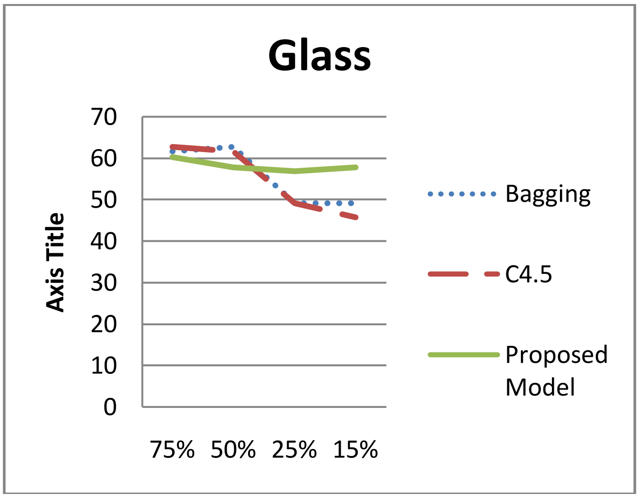

Figure 5 : Comparison Graphs for Glass (multi-class)

It is observed from the comparison tables and graphs that the proposed model has performed consistently. For fewer labeled data (15%), the model performed better than other classifier models. The performance of the model does not degrade much with decreasing number of labeled examples.

Experiments on the benchmark datasets shows that the proposed framework for both clustering and classification have performed consistently better for building models on the training set with varied range (75% to 15%) of labeled examples. When compared to other proven techniques, the proposed framework sustained its performance even when the number of labeled examples is reduced to 15% thus establishing its validity as a semi supervised learning model. The proposed framework was able to identify the relevant features along with their weightages thus reducing the information requirement for handling unknown situations may it be classification or clustering.

VI. Conclusion

In this paper, the authors proposed a framework for context-aware semi supervised learning in terms of both clustering and classification. The proposed framework is useful to work in the domains where availability of labeled data is either scarce or difficult/expensive to obtain. The framework with wrapper based feature selection is very much useful in handling high dimensional datasets. With dimensions reduced, a cluster and classification solution is defined with lesser number of features. This is very useful in cases where there are time and space constraints. The proposed framework not only identifies the relevant features but also estimates the importance of a feature in terms of weights such that cluster solutions are formed as per the intended purpose. Though the framework has used K-means for the formation of cluster solution, the proposed SFRE algorithm can be wrapped into any partitional clustering algorithm with equal ease for producing context-aware semi supervised subspace clusters leveraging a few labeled examples for defining the context.

Since the model uses discriminant analysis for identifying attributes, it is limited to the numerical data. However, in reality, many of the applications contains mixed data, a combination of numeric and categorical data. This opens an avenue for further research to extend the model to work with categorical data.

References Références Referencias

1. Advani, M. (2011). Learning from High Dimensional fMRI Data using Random Projections. Retrieved from http://cs229.stanford.edu/.
2. Al-Harbi, S. H., & Rayward-Smith, V. J. (2006). Adapting k-means for supervised clust-erin-g. Applied Intelligence, 24(3), 219-226.
3. Basu, S., Bilenko, M., & Mooney, R. J. (2004, August). A probabilistic framework for semi-supervised clustering. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 59-68). ACM.
4. Chapelle,O.,Schölkopf, B., and Zien, A. (20-6) Semi-supervised learning. Cambridge, MA: MIT press.
5. Dharmavaram, V. G., & Mogalla, S. (2013). Semi Supervised Weighted K-Means Clustering for Multi Class Data Classification. IJCAIT, 2(1), 36-44.
6. Ding, S., Qi, B., Jia, H., Zhu, H., & Zhang, L. (2012). Research of semi-supervised spectral clustering based on constraints expansion. Neural Computing and Applications, 1-6.
7. Eick, C. F., Rouhana, A., Bagherjeiran, A., & Vilalta, R. (2006). Using clustering to learn distance functions for supervised similarity asses-sm-ent. Engineering Applications of Artificial Inte-lligen-ce, 19(4), 395-401.

8. Gao, J., Tan, P. N., & Cheng, H. (2006, April). Semi-Supervised Clustering with Partial Background Information. In SDM.
9. Grira, N., Crucianu, M., & Boujemaa, N. (2004). Unsupervised and semi-supervised clustering: a brief survey. A review of machine learning tech-niques for processing multimedia content, Report of th-e MUSCLE European Network of Excellence (FP6).
10. Han, J., Kamber, M., & Pei, J. (2006). Data mining: concepts and techniques. New Delhi: Morgan kaufmann.
11. Lu, Z., & Leen, T. K. (2004). Semi-supervised lea-rning with penalized probabilistic clustering. I-n Ad-vances in neural information processing sy-ste-ms(pp. 849-856).
12. Padmaja P., Shashi M., Teja K.N.K. (2010). Intrinsic Dimensionality Based Conceptual Clustering, Inte-rnational Journal of Advanced Computer Eng-ineering, 3(1), 1-5.
13. Ramona, M., Richard, G., & David, B. (2012). Mul-ticlass feature selection with kernel Gram-matrix ba-sed criteria. Neural Networks and Learning Sy-stems, IEEE Transactions on, 23(10), 1611 – 1623.
14. Sim, K., Gopalkrishnan, V., Zimek, A., & Cong, G. (2013). A survey on enhanced subspace clu-ster-ing. Data mining and knowledge dis-covery, 26(2), 332-397.
15. Song, Q., Ni, J., & Wang, G. (2013). A fast clustering-based feature subset selection algorithm for high dimensional data. Knowledge and Data Engineering, IEEE Transactions on, 25(1), 1-14.
16. Tan Pang-Ning, Steinbach Michael and Kumar Vipin. (2006). Introduction to Data Mining. New Delhi: Pearson Education.
17. Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001, June). Constrained k-means clustering with background knowledge. In ICML (Vol. 1, pp. 577-584).
18. Wang, L. (2008). Feature selection with kernel class separability. Pattern Analysis and Machine Intell-igence, IEEE Transactions on, 30(9), 1534-1546.
19. Wang, Y., Chen, S., & Zhou, Z. H. (2012). New semi-supervised classification method based on modified cluster assumption. Neural Networks and Learning Systems, IEEE Transactions on, 23(5), 689-702.
20. Xiong, S., Azimi, J., & Fern, X. (2014). Active lear-ning of constraints for semi-supervised clustering. Kno-wledge and Data Engineering, IEEE Tran-saction on, 26(1), 43-54
21. Yin, X., & Hu, E. (2011). Distance metric learning guided adaptive subspace semi-supervised clus-tering. Frontiers of computer science in china, 5(1), 100-108.
22. Zhao, Q., & Qu, H. (2009, June). A Novel Supervised Clustering Based on the Feature Clas-sification Weight. In Computational Intelligence and Natural Computing, 2009. CINC'09. International Co-nference on (Vol. 1, pp. 117-120). IEEE.
23. Zhu, X. (2006). Semi-supervised learning literature survey. Computer Science, University of Wisconsin Madison, 2, 3.
24. Zhu, X., & Goldberg, A. B. (2009). Introduction to se-mi-supervised learning. Synthesis lectures on art-if-icial intelligence and machine learning, 3(1), 1-130.