AN APPROACH TO SEMI-SUPERVISED CLASSIFICATION USING THE HUNGARIAN ALGORITHM

Amparo Albalate, Aparna Suchindranath and Wolfgang Minker
Institute of Information Technology, University of Ulm, Ulm, Germany

Keywords: Semi-supervised classification, Clustering, Hungarian algorithm, Cluster pruning.

Abstract: In this paper we propose a novel semi-supervised classification algorithm from the cluster-and-label framework. A small amount of labeled examples is used to automatically label the extracted clusters, so that the initial labeled seed is implicitely ”augmented” to the whole clustered data. The optimum cluster labelling is achieved by means of the Hungarian algorithm, traditionally used to solve any optimisation assignment problem. Finally, the augmented labeled set is applied to train a SVM classifier. This semi-supervised approach has been compared to a fully supervised version. In our experiments we used an artificial dataset (mixture of Gaussians) as well as other five real data sets from the UCI repository. In general, the experimental results showed significant improvements in the classification performance under minimal labeled sets using the semi-supervised algorithm.

1 INTRODUCTION

Semi-supervised classification is a framework of algorithms proposed to improve the performance of supervised algorithms through the use of both labeled and unlabeled data (Design et al.,). One reported limitation of supervised techniques is their requisite of available training corpora of considerable dimensions order to achieve accurate predictions on the test data. Furthermore, the high effort and cost associated to labeling large amount of training samples by hand -a typical example is the manual compilation of labeled text documents- is a second limiting factor, which led to the development of semi-supervised techniques. It has been shown in numerous studies how the knowledge learned from unlabeled data can dramatically reduce the size of labeled data required to achieve appropriate classification performances (Nigam et al., 2000; Castelli and Cover, 1995).

Different approaches to semi-supervised classification have been proposed in the literature, including, among others, Co-training (Maeireizo et al., 2004), self-training (Yarowsky, 1995) or generative models (Nigam et al., 2000; Dempster et al., 1977). Two extensive surveys on semi-supervised learning are provided in (Zhu, 2006) and (Seeger, 2001). This paper focuses in a particular case of generative models, in which cluster algorithms are employed instead of probabilistic mixture models. This kind of approaches is commonly referred to as “cluster-and-label” framework (Zhu, 2006). The algorithm proposed in this paper differs from previous works in which both clustering and labeling stages are often integrated in one single process. Previously, the labeled seeds have been often used to initialise or guide the clustering algorithms, in such a way that the clusters’ patterns are implicitely tagged during the clustering process (Demiriz et al., 1999). In this work, however, the clustering and labeling tasks are separated into two independent processes. First, a cluster partition of the data set is obtained through a fully unsupervised clustering algorithm. Then, given a small set of labels (also referred to as prototype of labeled seed), a cost matrix is computed based on the distribution of labels through the clusters. The cluster labeling objective is then formulated as an assignment problem, which has been solved using the Hungarian algorithm (Kuhn, 1955). Thereby, an optimum cluster labeling *given the labeled seeds* is ensured. An extension of the proposed semi-supervised approach is also presented, using a cluster-pruning algorithm which is intended to improve the quality of the clusters by pruning such patterns with high probability of belonging to a overlapping region between classes.

The paper is organised as follows: Section 2 provides an overview of related work in the field semi-supervised classification. In Section 3, we outline the proposed algorithm. One important task in the new

Albalate A., Suchindranath A. and Minker W..
AN APPROACH TO SEMI-SUPERVISED CLASSIFICATION USING THE HUNGARIAN ALGORITHM.
DOI: 10.5220/0003187304240433
In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (ICAART-2011), pages 424-433
ISBN: 978-989-8425-40-9

algorithm is the optimum cluster labeling, which is explained in more detail in Section 4. In Section 5 we propose an extension to the semi-supervised algorithm described in Section 3. The data sets used in the experiments are introduced in Section 6. Finally, we draw conclusions and future directions algorithm in Section 7.

2 RELATED WORK

Different types of semi-supervised classifiers can be distinguished in the literature. Among them, in this section we briefly describe three of the main approaches: *self-training*, *co-training* and *generative models*.

2.1 Self-training

In self training, a single classifier is iteratively trained with a growing set of labeled data, starting from a small initial seed of labeled samples. Commonly, an iteration of the algorithm entails the following steps: 1) training on the labeled data available from previous iterations, 2) Applying the model learned from labeled data to predict the unlabeled data and 3) Sorting the predicted samples according to their confidence scores and adding the top most confident ones with their predicted labels to the labeled set.

One example of self training is the work by Yarowski (Yarowsky, 1995) on word sense disambiguation. A self training approach was applied to to classify a word and its context into the possible word senses in a polysemic corpus, starting by a tagged seed for each possible sense of the words.

2.2 Co-training

In a similar way as self-training, co-training approaches are based on an incremental augmentation of the labeled seeds by iteratively classifying the unlabeled sets and attaching the most confident predicted samples to the labeled set. However, in contrast to self training, two complementary classifiers are simultaneously applied, fed with two different “views” of the feature set. The prediction of the first classifier is used to augment the labeled set available to the second classifier and vice-versa. In (Maeireizo et al., 2004), a co-training strategy was applied to predict the emotional/non-emotional character of a corpus of student utterances collected within the ITSPOKE project (Intelligent Tutoring Spoken dialog system). The authors selected two *high-precision* classifiers. The first one was trained to recognise the emotional status of an utterance (e.g. ’1’ emotional vs ’0’ for non-emotional), while the second one predicted its non-emotional status (’1’ non-emotional vs. ’0’ emotional). The labeled set was iteratively increased by attaching the top most confident predicted samples to the labeled set from previous iterations.

2.3 Generative Models

Given a data set of observations $\mathcal{X}$, with the corresponding set of class labels, $\mathcal{Y}$, a generative model assumes that the observations and labels are drawn according to a model $p(x,y)$ whose parameters should be “identifiable” (Zhu, 2006). Typically, the Expectation Maximisation algorithm is applied to estimate the model parameters (Nigam et al., 2000).

Other strategies attempt to derive the underlying class distribution by means of clustering techniques. These approaches are commonly referred to as the *cluster -and- label* paradigm. For example, in (Demiriz et al., 1999) a *genetic* k-means clustering was implemented using a genetic algorithm. The goal of the algorithm was to find a set of k cluster centres that simultaneously optimised an internal quality objective (e.g minimum cluster dispersion) and an external criterion based on the available labels (e.g minimum cluster entropy). The simultaneous optimisation concerning internal and external criteria was attained through the formulation of a new objective function as a linear combination of both criteria.

3 NOVEL SEMI-SUPERVISED ALGORITHM USING THE CLUSTER AND LABEL STRATEGY

In this paper we propose a new semi-supervised algorithm, according to a cluster-and-label strategy. As explained in Section 1, in previous works, the labeling task has been often integrated into the clustering process as a simultaneous optimisation problem. In other words, the clusters’ patterns are simultaneously tagged during the clustering process.

Such simultaneous definition of the optimisation problem (clustering/labeling) produces a certain dependency of the extracted clusters with respect to the initial labels. Thus, potential labeling errors present in the labeled seeds may also induce a certain degradation of the clustering solution. In fact, training sets are not exempt from potential labeling errors. These may occur depending on the degree of expertise of the human annotators. Even for expert labelers, the confidence in annotating patterns with a certain degree of ambiguity may drop down significantly, as it happens, for example, with the annotation of non-acted emotions.

In other to avoid the aforementioned limitation, the approach proposed in this paper distinguishes the clustering and labeling processes as two independent optimisation problems. Essentially, the data set (both labeled and unlabeled patterns) is first clustered, *without any* a-priory information concerning labels. Thereby, a fully unsupervised, data-driven solution is enforced which optimises an *internal* quality objective. Then, the distribution of labels through the different clusters is taken into consideration in order to achieve the optimum labeling of the clusters' patterns. Thereby, higher robustness against possible errors in the labeled seeds is achieved in the proposed approach.

Data Set. First, the data is divided into a test set ($\sim$ 10%) and a training set ($\sim$ 90%). Let

$$\mathcal{X}_T = \{x_1, x_2, \cdots, x_p\}, \qquad \forall x_i \in \mathcal{R}^N.$$

denote the training data points. This set is in turn divided by two disjoint subsets:

$$\mathcal{X}_T = \mathcal{X}_T^{(l)} \cup \mathcal{X}_T^{(u)}$$

denoting $\mathcal{X}_T^{(l)}$ the labeled portion of $\mathcal{X}_T$ for which the corresponding set of labels $\mathcal{Y}_T^l$ is assumed to be known, and $\mathcal{X}_T^{(u)}$, the subset of unlabeled patterns in $\mathcal{X}_T$.

Clustering. The first step in the semi-supervised approach is to find a cluster partition $\mathcal{C}$ of the training data X_T in to a set of k disjoint clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$, where k is the number of classes (which is assumed to be known from the labeled set). In this work, the Partitioning around Medoids (Pam) algorithm has been selected using the Euclidean distance to compute dissimilarity matrices. The Pam clustering algorithm provides the cluster solution wich minimises the *sum of distances* to the cluster medoids.

Optimum Cluster Labeling. The labeling block performs a crucial task in the semi-supervised algorithm. Given the set of clusters $\mathcal{C}$ in which the training data is divided, the objective of this block is to find an optimum bijective mapping of labels to clusters:

$$\mathcal{L} : \mathcal{C} \rightarrow \mathcal{K}, \quad \mathcal{K} = \{1, 2, 3, \cdots, k\}$$

so that an optimum criterion is fulfilled. Each cluster is assigned exactly one class label in $\mathcal{K}$. This mapping of clusters to class labels is equivalent to a mapping function that assigns, to each clustered pattern, the class label of the cluster where it belongs. As a result of cluster labeling, the initial labeled seed $(\mathcal{X}_T^{(l)}, \mathcal{Y}_T^{(l)})$ is extended to the complete training set $(\mathcal{X}_T, \mathcal{Y}_T)$, denoting $\mathcal{Y}_T$, the set of augmented labels corresponding to the observations in $\mathcal{X}_T$

Classification. Finally, a Support Vector Machine (SVM) classifier (Burges, 1998; Joachims et al., 1997) is trained with the augmented labeled set $(\mathcal{X}_T, \mathcal{Y}_T)$ obtained after cluster labeling. The SVM learned model is then applied to predict the labels for the test set.

Simultaneously, a fully supervised classification scheme has been compared to the semi-supervised algorithm. In this case, the SVM is directly trained with the initial labeled seed ($\mathcal{X}^{(l)}$, $\mathcal{Y}^{(l)}$).

Both semi-supervised and supervised strategies have been evaluated in terms of accuracy, by comparing the predicted labels of the test patterns with their respective manual labels. The evaluation results are discussed in Section 6.

4 OPTIMUM CLUSTER LABELING

In this section, we described in more detail *optimum cluster labeling* task in the proposed semi-supervised algorithm.

Problem Definition. Given the training data, $\mathcal{X}_T = \mathcal{X}_T^{(l)} \cup \mathcal{X}_T^{(u)}$, the set $\mathcal{Y}_T^{(l)}$ of labels associated to the portion $\mathcal{X}_T^{(l)}$ of the training set, the set $\mathcal{K}$ of labels for the k existing classes [1], and a cluster partition $\mathcal{C}$ of $\mathcal{X}_T$ into disjoint clusters, the optimum cluster labeling problem is to find a bijective mapping function, L:

$$L : \mathcal{C} \rightarrow \mathcal{K}, \quad \mathcal{K} = \{1, 2, 3, \cdots, k\}$$

that assigns each cluster in $\mathcal{C}$ to a class label in $\mathcal{K}$, while minimising the total labeling cost. This cost is defined in terms of the labeled seed $(\mathcal{X}_T^{(l)}, \mathcal{Y}_T^{(l)})$ and the set of clusters $\mathcal{C}$. Consider the following matrix of overlapping products N:

[1] Although class labels can take any arbitrary value, either numeric or nominal, for simplicity in the formulation and implementation of the cluster labeling problem the k class labels are transformed to integer values ($[1 \ldots k]$).

$$N = \begin{pmatrix} n_{i1} & n_{i2} & \cdots & n_{ik} \\ n_{21} & n_{22} & \cdots & n_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ n_{k1} & n_{k2} & \cdots & n_{kk} \end{pmatrix}$$

with constituents n_{ij}, denoting the number of labeled patterns from $\mathcal{X}_T^{(l)}$ with class label $y = i$ that fall into cluster C_j. The labeling objective is to minimise the global cost of the cluster labeling denoted by L:

$$Total\ Cost(L) = \sum_{C_i \in C} w_i \cdot Cost\big(L(C_i)\big) \qquad (1)$$

where $W = (w_1, \cdots, w_k)$ is a vector of weights for the different clusters. For example, it may be used if clusters sizes show significant differences among the clusters. In this paper, the weights are assumed to be equal for all clusters, so that $w_i = 1, \forall i \in 1 \cdots k$.
The individual of labeling a cluster C_i with class j is defined as the number of samples from class j (in the labeled seed) that fall outside the cluster C_i, i.e.:

$$Cost\big(L(C_i)\big) = \sum_{C_k \neq C_i} n_{L(C_i),k} \qquad (2)$$

by applying Equation 2 into the total cost definition of Equation 1, yields:

$$Total\ Cost(L) = \sum_{C_i \in C} \sum_{C_k \neq C_i} n_{L(C_i),k} \qquad (3)$$

Using a greedy search algorithm, the cost minimisation of Equation 1 requires $k!$ operations (where k denotes the number of clusters/classes). Such a complexity becomes computationally intractable for $k \geq 10$. However, larger number of classes are often involved in real classification problems. In this paper, the Hungarian algorithm have been used, which can achieve the optimum cluster labeling with substantially lower complexities. It requires the definition of a cost matrix $\mathbf{C}_{[kxk]}$, whose rows denote the clusters and the columns are referred to class labels in $\mathcal{K}$. The elements $\mathbf{C}_{ij}$ denote the individual costs of assigning the cluster C_i to class label j, i.e. $\mathbf{C}_{ij} = Cost(L(C_i) = j)$.

4.1 The Hungarian Algorithm

The hungarian algorithm was devised by Harold Huhn in 1955 to solve the optimum assignment problem in polynomial time. The name "Hungarian" was given after two hungarian scientists who had previously established large part of the algorithm's mathematical background. It finds the optimum assignment on a matrix of costs where each element $\mathbf{C}_{i,j}$ denotes the penalty paid for the corresponding individual assignment (i, j). A typical example is the *worker-job* assignment where the rows represent different workers and the columns are the jobs to which the workers can be designated to. The original algorithm proposed by Huhn solved the assignment task in $O(k^4)$ operations, although some extensions of the algorithm have been proposed, leading to a complexity of $O(k^3)$.

The Hungarian algorithm has been described in terms of bipartite graphs, or equivalently, as a number of steps involving certain manipulations of the input cost matrix, which can be summarised as follows.

1. Substract from each row of the cost matrix, the values of the smallest element in the row.
2. Proceed as in step 1. columnwise.
3. Cross out the necessary rows and/or columns to cover all zeros in the modified cost matrix from step 2. by drawing the minimum number of lines.
4. If a number of k lines have been drawn, proceed to perform the assignments in step 5. Otherwise select the smallest number not covered by any line drawn in step 3. Substract this value to the non-covered elements, adding the value to the elements that are covered by two lines.
5. Starting from the first row, if the row contains a unique zero element in a column j, assign the worker in the row to the j^{th} job. Prune the row and column from the cost matrix and continue scanning the rest of rows. If some of the assignments are still left at the end of this process, repeat the procedure columnwise. If still some assignments are left, it means that a unique assignment is not possible. In such case, the remaining assignments can be performed at random.

5 DATA SETS

Mixture of Gaussians. This data set comprises a mixture seven Gaussians in two dimensions and 1750 instances (250 in each Gaussian), where a certain amount of overlapping patterns (potential ambiguities) can be observed.

Iris Data Set (Iris). The Iris set is one of the most popular datasets from the UCI repository (uci,). It comprises 150 instances iris of 3 different classes of iris flowers (Setosa, Versicolor, virginica). Two of these classes are linearly separable while the third one is not linearly separable from the second one.

Wine Data Set (Wine). The wine set is one of the popular data sets from the UCI repository. It consists of 178 instances with 13 attributes, representing three different types of wines.

Wisconsin Breast Cancer Data Set (Breast). This data set constains 569 instances in 10 dimensions, denoting 10 different features extracted from digitised images of breast masses. The two existing classes are referred to the possible breast cancer diagnosis (malignant, benign).

Handwritten Digits Data Set (Pendig). The fourth real data set is for pen-based recognition of handwritten digits. In our experiments, we used the test partition[2], composed of 3498 samples with 16 attributes. Ten classes can be distinguished for the digits 0-9.

Pima Indians Diabetes (Diabetes). This data set comprises 768 instances with 8 numeric attributes. Two classes denote the possible diagnostics (the patients show or not signs of diabetes.).

6 EXTENSION THROUGH CLUSTER PRUNING

In this section, an alternative to the cluster-and-label strategy is introduced. Even though the underlying class structure can be appropriately captured by a cluster algorithm, the augmented data set derived by the optimum cluster labeling may contain a number of "misclassification"[3] errors with respect to the real class labels. This happens specially when two or more of the underlying classes show a certain overlapping of patterns. In this case, the errors may be accumulated in the regions close to the cluster boundaries of adjacent clusters.

The general idea behind the proposed optimisation method is to improve the (external) cluster quality by identifying and removing such regions with high probability of missclassification errors from the clusters. To this aim, the concept of *pattern silhouettes* has been applied to prune the clusters in $\mathcal{C}$.

The silhouette width of an observation x_i is an internal measure of quality, typically used as the first step for the computation of average silhouette width of a cluster partition (Rousseeuw, 1987). It is formulated as:

$$s(x_i) = \frac{b(x_i) - a(x_i)}{max(a(x_i), b(x_i))} \tag{4}$$

where a is the average distance between x_i and the elements in its own cluster, while b is the smallest average distance between x_i and other clusters in the partition. Intuitively, the silhouette of an object $s(x_i)$, can be thought of as the "confidence" to which the pattern x_i has been assigned to the cluster $C(x_i)$ by the clustering algorithm. Higher silhouette scores are observed for patterns clustered with a higher "confidence", while low values indicate patterns which lie between clusters or are probably allocated in the wrong cluster.

The cluster pruning approach can be described as follows:

Input A cluster partition $\mathcal{C}$ of the data set; the distance matrix D
Output A set of pruned clusters $\mathcal{C}'$.

1. Given a cluster partition $\mathcal{C}$ and the matrix of dissimilarities between the patterns in the data set, D, calculate the silhouette of each object in the data set.

2. Sort the elements in each cluster according to their silhouette scores, in increasing order.

3. In each cluster, the elements with high silhouettes can be considered as objects with high "clustering confidence". In contrast, such elements with low silhouette values are clustered with lower confidence. This latter kind of objects may thus belong to a class-overlapping region with higher probability. Using the histograms of silhouette scores within the clusters, select a minimum silhouette threshold for each cluster. Further details about the selection of silhouette thresholds by the cluster pruning algorithm are provided in Section 6.1.

4. Prune each cluster C_i in $\mathcal{C}$ by removing patterns which do not exceed the minimum silhouette threshold for the cluster, chosen in the previous step.

6.1 Determination of Silhouette Thresholds

In the proposed cluster pruning method, different silhouette thresholds are applied according to the distribution of silhouette values within each cluster, es-

[2] due to memory limitations of the R software used in the experiments

[3] The term missclassification is not here used to indicate the predicted errors of the end classifiers but the errors after the cluster labeling block. Note that, after cluster labeling, each clustered data pattern is assigned a class label (the label of its cluster), which can be compared to the real label if the complete labeled set is available.

timated through histograms. Assuming that the underlying class distribution is appropriately captured in the cluster partition, if a significant distortion of the original clusters is introduced through cluster pruning, the learned SVM models may also deviate from the expected models to a certain extent. The objective is to remove potential clustering errors while preserving to the highest possible extent the shape and size of the original clusters. In practice, pruning an amount of patterns from 20% to 30% of the cluster size has been considered appropriate for the current purpose. In addition, the selected thresholds also depend on the pattern silhouette values: patterns with a silhouette score larger than 0.5 are deemed to be clustered with a sufficiently high "confidence". Thus, the maximum silhouette threshold applied in the cluster pruning algorithm is $sil_{th} = 0.5$. In consequence, if the minimum observed silhouette score in a cluster is larger than 0.5, the cluster remains unaltered in the pruned partitions.

The specific criteria to select the silhouette thresholds can illustrated by considering the clusters extracted from the Breast data set (all 569 data instances). The distribution of silhouette scores has been estimated by using the histogram function in the R-software, which also provides the vectors of silhouette values found as the histogram bin limits and the counts of occurrences in each bin [4]. The silhouette thresholds have been selected to coincide to histogram bin limits'. In the Breast data set (2 classes/clusters), the vector of silhouette thresholds for the first and second clusters is [0.5, 0.2]. The value $sil_{th} = 0.5$ for the first cluster corresponds to the upper bound for the silhouette thresholds, as explained in the previous paragraph. It results in the removal of 5.2% of the cluster's patterns. For the second cluster, the threshold $sil_{th} = 0.2$ is selected. The pruned section associated to sil_{th} corresponds to the first five histogram bins, comprising 25% of the patterns in the cluster. By including the sixth histogram bin in the pruned section, the next possible silhouette threshold level is $sil_{th} = 0.3$, However, such threshold level would lead to the removal of a considerable amount (46.28%) of the cluster patterns, which is considered unacceptable for preserving the cluster size/shape.

To summarise, the number of histogram bins corresponding to rejected patterns is determined according to one of these two conditions: (1) the upper limit of the last rejected bin should not be greater than $sil_{th} = 0.5$, and (2) The amount of rejected patterns (total number of occurrences in the rejected bins) should not exceed a ratio of 30% of the total number of patterns in the cluster.

6.2 Evaluation of the Cluster Pruning Approach

In this section, the efficiency of the cluster pruning method for rejecting missclassification errors from the clustered data is evaluated through an analysis of the algorithm outcomes on the Iris, Wine, Breast Cancer, Diabetes, Pendig and Seven Gaussians data set[5].

For the purpose of evaluating the cluster pruning algorithm, the cluster labeling task has been performed using the complete set of labels for each data set. The resulting misclassification error rates as well as the NMI results observed in Table 1 confirm the adequate behaviour of the proposed cluster pruning algorithm for removing such sections from the clusters with high probability of resulting in misclassification errors after cluster labeling. For instance, while the pruned sections comprise around $10-20\%$ of the patterns in the data sets, the percentage of remaining misclassification errors has been substantially reduced. As an example, the error rate has dropped from 10.66% to 4.03% after pruning on the Iris data set, while error rates have been reduced from 4.09% to 0.99% for the Breast data set, and from 22.40% to 8.98% in the Wine dataset. An exception to the previous observations is the Diabetes data set, in which the error rate after cluster pruning (38.16%) remains very similar to the original missclassification rate (40.10%) - note that, for 2 clusters as in the case of the Diabetes data, the worst possible error rate that can be observed is of 50%. Any error rate larger than 50% is not observed as it just produces an inversion of the cluster labels. In other words, the original error in the diabetes data set implies almost a roughly uniform distribution of patterns from any of the two underlying classes in the extracted clusters. This fact is also evidenced by the NMI score 0.012. In consequence, the error rate is roughly the same after cluster pruning, and the removal of patterns by means of cluster pruning algorithm is just as efficient as removing the same amount of patterns at random.

7 SIMULATIONS AND RESULTS

In the experimental setting, SVMs have been used as the baseline classifier. First, each data set has been

[4] The bin sizes provided by the R-software histogram function are estimated according to the Sturges formula (Freedman and Diaconis, 1981)

[5] note that the cluster partitions obtained in this experiments comprise all instances of the data sets (without prior partitions into test/training).

Table 1: Some details about the cluster pruning approach in the Iris, Wine, Breast cancer, Diabetes, pendig and Seven Gaussians data set.

Data Set	Silhouette thresholds	% Removed patterns	Error 1 (%)	Error 2 (%)	NMI 1	NMI 2
Iris	[0.5 0.3 0.4]	17.33%	10.66 %	4.03%	0.758	0.888
Wine	[0.2 0.14 0.24]	22.40 %	8.98 %	0.72%	0.783	0.967
Breast	[0.5 0.2]	11.56 %	4.09 %	0.99%	0.741	0.910
Diabetes	[0.5 0.1]	16.35 %	40.10%	38.16%	0.012	0.022
Pendig	[0.2 0.3 0.2 0.2 0.25 0.2 0.15 0.15 0.25 0.2]	20.10 %	31.93%	21.22%	0.701	0.796
Seven Gaussians	[0.4 0.5 0.4 0.4 0.4 0.4 0.4]	11.77 %	0.27% %	0.02%	0.944	0.993

divided into two training ($\sim$ 90%) and test ($\sim$ 10%) subsets. In order to avoid possible biases of a single test set or labeled seed, such partition of the data set into a training and test portions has been randomly repeated to generate 20 different partitions. Also, for each one of these partitions, 20 different random seeds of labeled prototypes (n labels /category) have been selected. In total, 400 different prototype seeds (20x20) have been obtained. In the experiments, only prototype labels are assumed to be known a-priory. No other class label knowledge has been applied to any of the algorithm stages. Each prototype seed has been used as the availabe training set for the supervised SVM. In the semi-supervised approach, these labeled prototype seeds have been used to trigger the automatic cluster labeling.

Both supervised and semi-supervised SVM classifiers have been evaluated on an accuracy basis, considering different number of labeled prototypes (samples) per category, from $n = 1$ to $n_{max} = 30$. The accuracy results obtained on the different data sets are shown in Figures 1 and 2. In particular, left plots are referred to the supervised and the semi-supervised approach without cluster pruning, while right plots are referred to the semi-supervised approaches by incorporating the cluster pruning approach.

Note that the right and left plots are obtained from different experiments (in each experiment a different labeled seed is involved) so that the mean accuracy values of the supervised approach in left and right plots can slightly differ. In all cases, horizontal axes are referred to the sizes of the initial prototype seeds, whereas vertical axes indicate the mean accuracy scores, averaged over the 400 prototype initialisations.

As it can be observed in Figures 1 and 2, the mean accuracy curves of the semi-supervised algorithm are roughly constant or slowly increasing with the labeled set size. Certain random variations can be observed, since the experiment outcomes for different seed sizes are referred to different random prototype seeds (note, however, that for each labeled set size, both supervised and semi-supervised outcomes have been simultaneously obtained with identical sets of prototypes, so that their respective accuracy curves can be compared). In contrast, accuracy curves of the supervised approach show stronger increasing trends with the labeled set sizes. In the Seven Gaussians, Iris, Pendig, Wine and Breast Cancer data sets, the mean accuracy curves for the supervised and semi-supervised algorithms intersect at certain labeled set sizes, n'. For smaller labeled seed sizes ($n < n'$), the training "information" available in the augmented labeled sets (after cluster labeling) is clearly superior than the the small labeled seeds. Therefore, although the augmented labels are not exempt from misclassifications due to clustering errors, higher prediction accuracies are achieved by the semi-supervised approach with respect to the supervised classifier. For ($n \geq n'$), the information in the increasing labeled seeds compensates for the missclassification errors present in the augmented sets and thus the supervised classifier outperforms the semi-supervised approach. As shown in the previous section, these errors present in the augmented data sets can be notably reduced by means of cluster pruning. In consequence, an improvement in the prediction accuracies achieved by the semi-supervised algorithm is generally observed by incorporating the cluster-pruning algorithm. Note that the values of n shown in the plots range from $n = 0$ to values slightly larger than the respective intersection points n'.

Unlike the accuracy results observed in the Seven Gaussians, Iris, Pendig, Wine and Breast Cancer data sets, a degradation in the semi-supervised classification performance with respect to the supervised classifier is observed in the Diabetes data set, regardless of the initial labeled seed sizes. This observa-

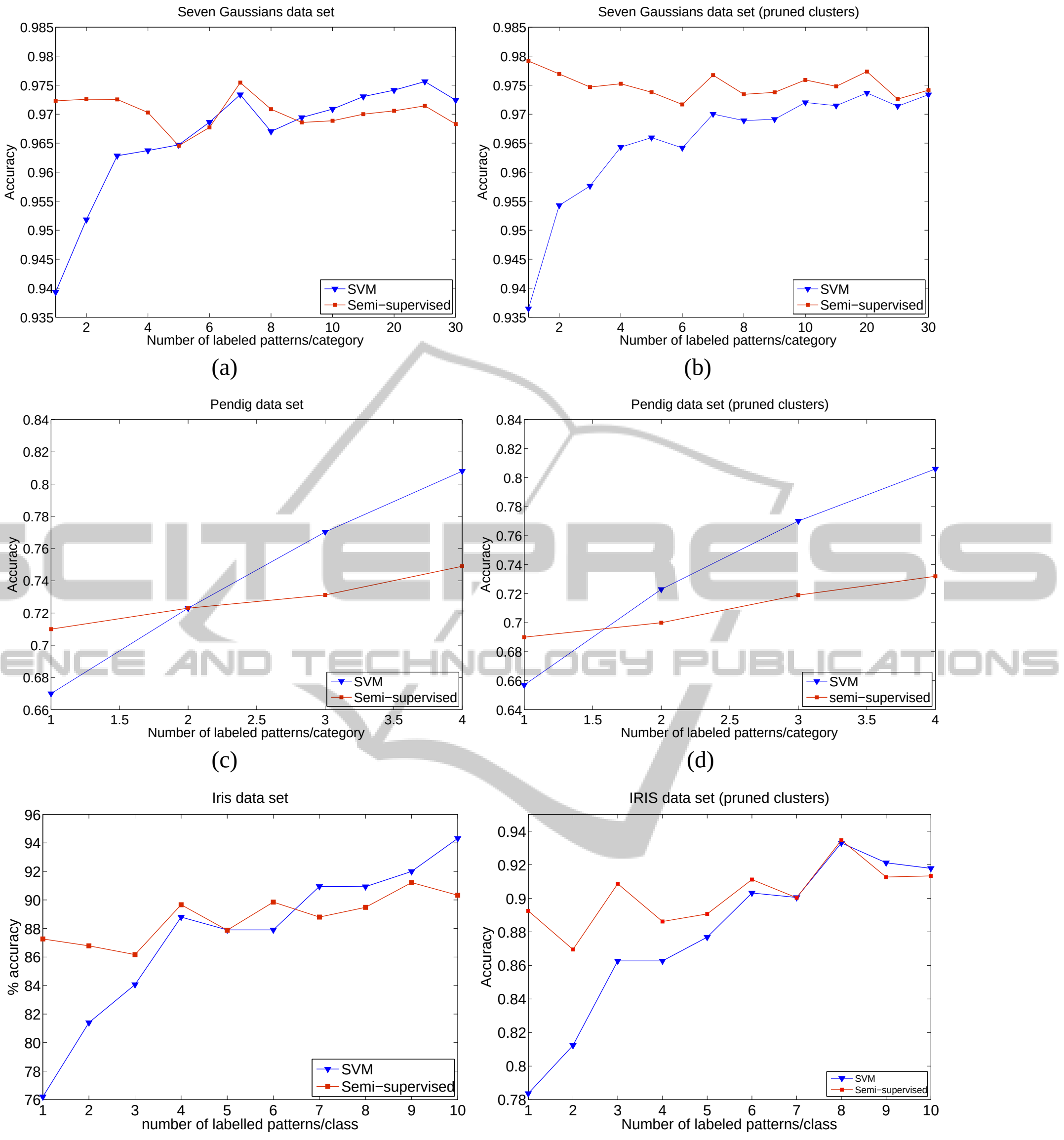
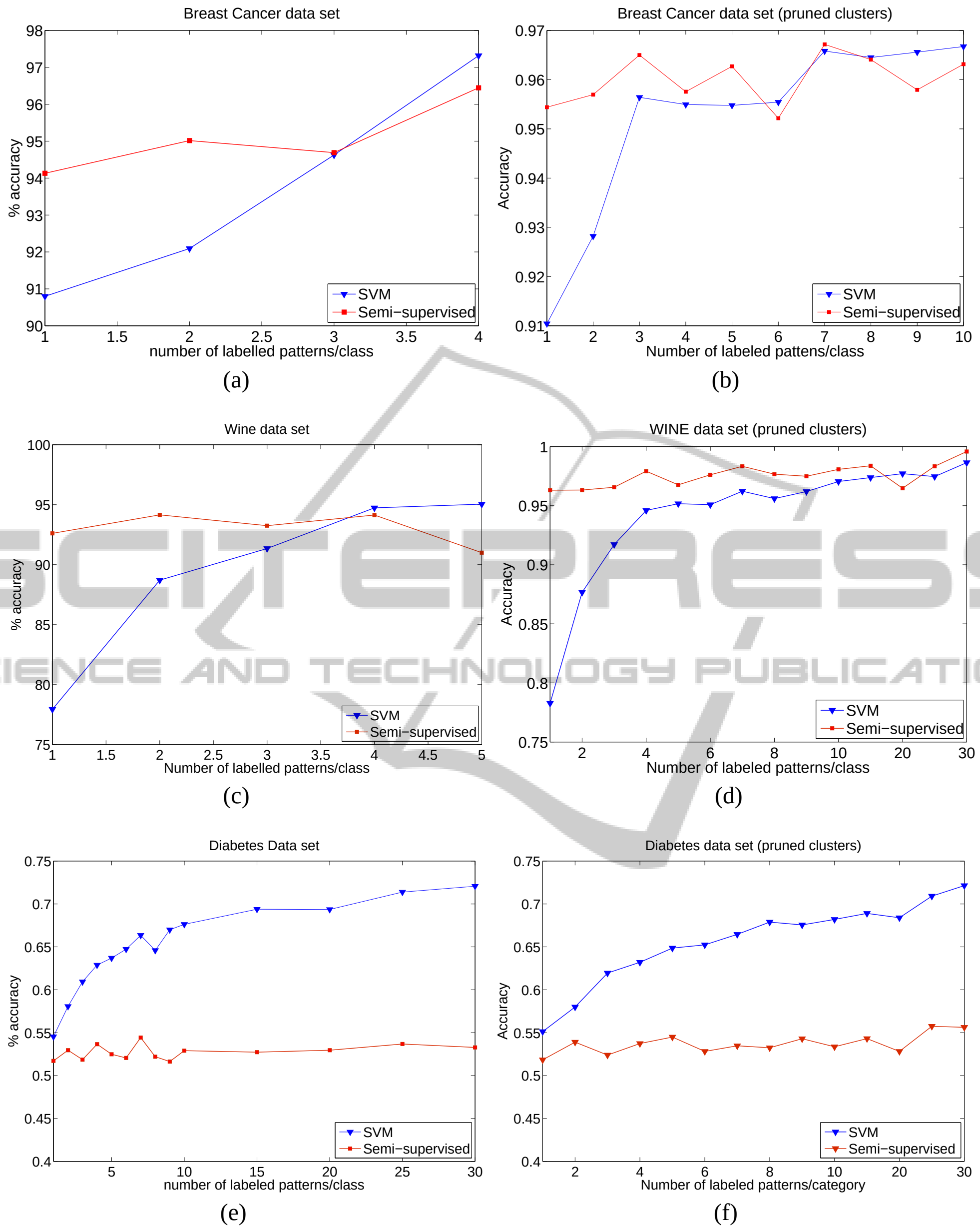


Figure 1: Mean accuracy curves obtain by the supervised (blue curves) and semi-supervised (red curves)classifiers. Left plots are referred to the basic semi-supervised approach, while right plots are obtained with the extension of the semi-supervised approach by means of cluster pruning.

tion is strictly associated to the NMI scores of the extracted clusters presented in the previous section (NMI=0.012), which corresponds to a missclassification rate of 40.10%. This means that almost no information concerning class labels is present in the augmented data sets used to train the SVM models. As a consequence, the semi-supervised performance on the diabetes data set is thus comparable to the a classifier which just performs random predictions, as it corresponds to the use of "unlabeled data alone" (Castelli and Cover, 1995).

8 CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, a semi-supervised approach has been presented based on the cluster-and-label paradigm. In contrast to previous works in the semi-supervised classification literature, in which labels are commonly integrated in the clustering process, in this work, the cluster and labeling processes are independent from each other. First, a conventional unsupervised clustering algorithm, the partitioning around medoids



(a) (b)

(c) (d)

(e) (f)

Figure 2: Mean accuracy curves obtain by the supervised (blue curves) and semi-supervised (red curves)classifiers.

.

(PAM) (Kaufmann and Rousseeuw, 1990) is used to obtain a cluster partition. Then, the output cluster partition, as well a small set of labeled prototypes (also referred to as labeled seeds) are used to decide the optimum cluster labeling given the labeled seed. The cluster labelling problem has been formulated as a typical assingment optimisation problem, whose solution is obtained by means of the Hungarian algorithm. Experimental results have shown significant improvements in the classification accuracy for minimum labeled sets, in such data sets where the underlying classes can be appropriately captured by means of unsupervised clustering.

In addition, an optimisation of the semi-supervised algorithm has been also developed by discarding the patterns clustered with small silhouette scores. Thereby, it has been shown that the quality of the pruned clusters can be improved, as significant reductions of the missclassification errors present in the clustered data are achieved through the removal of relatively small amounts of patterns from the clusters.

Future work is to investigate other possible alternatives for the definition of the cost matrix used by the

Hungarian algorithm. For example, probabilistic definition of the cost matrix by estimating class-cluster probabilities given the labeled seeds.

A further issue to be analysed is the choice of the number of clusters *k*, to be larger than the number of predefined categories. We believe such an strategy may provide better classification performances - specially for larger numbers of categories - as clusters can be more “specified” (lower Entropy values) with members of one category. In such case, the cluster sizes should be also taken into account for the definition of labeling costs. Moreover, this strategy would result in rectangular (non-square) cost matrices for which the Hungarian algorithm does not apply. A suitable alternative would be to solve the labeling problem given the cost matrices by means of genetic algorithms.

REFERENCES

http://archive.ics.uci.edu.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167.

Castelli, V. and Cover, T. M. (1995). On the exponential value of labeled samples. *Pattern Recogn. Lett.*, 16(1):105–111.

Demiriz, A., Bennett, K., and Embrechts, M. J. (1999). Semi-supervised clustering using genetic algorithms. In *In Artificial Neural Networks in Engineering (ANNIE-99*, pages 809–814.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Design, I. T., Gabrys, B., and Petrakieva, L. Combining labelled and unlabelled data.

Freedman, D. and Diaconis, P. (1981). On the histogram as a density estimator:l2 theory. *Probability Theory and Related Fields*, 57(4):453–476.

Joachims, T., Informatik, F., Informatik, F., Informatik, F., Informatik, F., and Viii, L. (1997). Text categorization with support vector machines: Learning with many relevant features.

Kaufmann, L. and Rousseeuw, P. (1990). *Finding Groups in Data. An Introduction to Cluster Analysis*. Wiley, New York, USA.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

Maeireizo, B., Litman, D., and Hwa, R. (2004). Co-training for predicting emotions with spoken dialogue data. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*.

Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *Mach. Learn.*, 39(2-3):103–134.

Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Jornal of Computational and Applied Mathematics*, 20:53–65.

Seeger, M. (2001). Learning with labeled and unlabeled data. Technical report.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*.

Zhu, X. (2006). Semi-supervised learning literature survey.