A semi-supervised cluster-and-label approach for utterance classification

Amparo Albalate[1], *Aparna Suchindranath*[1], *David Suendermann*[2], *Wolfgang Minker*[1]

[1]Institute of Information Technology, University of Ulm, Ulm, Germany
amparo.albalate@uni-ulm.de,aparna.suchindranath@uni-ulm.de, wolfgang.minker@uni-ulm.de

[2]SpeechCycle Labs, New York, USA
david@speechcycle.com

Abstract

In this paper, we propose a semi-supervised cluster-and-label algorithm for utterance classification. The approach assumes that the underlying class distribution is roughly captured through–fully unsupervised–clustering. Then, a minimum number of labeled examples is used to automatically label the extracted clusters so that the initial label set is "augmented" to the whole clustered data. The optimum cluster labeling is achieved by means of the Hungarian algorithm, traditionally used to solve optimization assignment problems. Finally, the augmented labeled set is applied to train an SVM classifier. We compare this semi-supervised approach to a fully supervised version in which the initial labeled sets are directly used to train the SVM model.

1. Introduction

In this paper, we propose a semi-supervised algorithm applied to the classification of transcribed utterances. We apply this algorithm to the natural language problem capture of troubleshooting dialog system. These systems automatically resolve customer care issues over the phone in a similar way as human agents do [1]. One important characteristic of these systems is the natural language modality of interaction with the user. The users are allowed to describe the experienced problem with their own words, and it is the system's task to analyze the utterance and classify it into the most probable symptom category. In commercial state-of-the-art implementations, the symptom classification task is performed by supervised classifiers. However, a significant limitation of supervised techniques is the requirement of labeled corpora of considerable dimensions in order to achieve accurate predictions. Numerous studies have shown how knowledge learned from unlabeled data can dramatically reduce the size of labeled data required to achieve appropriate classification performances [2]. 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.

In literature, several approaches to semi-supervised classification have been proposed, including, co-training [3], self-training [4] or generative models [2]. This paper focuses on 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" [5]. The algorithm proposed in this paper differs from previous work that mostly includes both clustering and labeling in a single optimisation problem. Commonly, the labeled seeds have been often used to initialize or guide the clustering algorithms in such a way that the clusters' patterns are implicitely tagged during the clustering process. In this work, however, the clustering and labeling tasks are separated into two independent processes. First, a cluster partition of the data set is produced by 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 throughout the clusters. The cluster labeling objective is then formulated as an assignment problem that is solved using the Hungarian algorithm [6]. 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 an overlapping region between classes.

The paper is organized as follows: In Section 2, we outline the proposed semi-supervised algorithm. One important task of the new algorithm is the optimum cluster labeling which is explained in more detail in Section 3. In Section 4, we propose an extension to the semi-supervised algorithm described in Section 2. Experimental results are discussed in Section 5. Finally, we draw conclusions and propose future directions in Section 6.

2. The new semi-supervised algorithm

As outlined in the introduction, in previous work, the labeled seeds have been often used to initialize or guide the clustering algorithms integrating the labeling task into the clustering process. In other words, the clusters' patterns are simultaneously tagged during the clustering process.

In consequence, the initial labeled sets may influence, to a certain degree, the quality of the discovered clusters, especially if the labeled sets are not exempt from labeling errors.

In the proposed method, clustering and labeling is separated into two independent tasks. Essentially, the data set (both labeled and unlabeled patterns) is first clustered, *without any* a priori information concerning labels. Thereby, a fully unsupervised, data-driven solution is enforced. Then, the distribution of labels through the different clusters is taken into consideration, in order to achieve an optimum labeling of the clusters' patterns.

Data set: First, the data set is divided into test and training subsets. 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 into 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 of the semi-supervised approach is to find a cluster partition $\mathcal{C}$ of the training data X_T into 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, we use the Partitioning around Medoids (Pam) algorithm in conjunction with two different distance functions to compute the matrix of dissimilarities between utterances: the cosine distance and the overlap distance. The overlap similarity between two utterances is the number of words that both utterances have in common. If utterances are represented as binary vectors of term occurrences (see Section 5), the overlap similarity corresponds to the dot product between utterance vectors. The overlap distance is then defined as $M - overlap\ similarity$, where M is the maximum of the similarity matrix.

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:

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

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 the class label of the cluster where it belongs to each cluster member. As a result of the 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 [7] is trained with the augmented labeled set $(\mathcal{X}_T, \mathcal{Y}_T)$ obtained after cluster labeling. The SVM model is then applied to predict the labels for the test set.

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

3. Optimum cluster labeling

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 with 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

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

that assigns each cluster in $\mathcal{C}$ to a class label in $\mathcal{K}$, while minimizing 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:

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

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 minimize the global cost of the cluster labeling denoted by L:

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

where $W = (w_1, \cdots, w_k)$ is a vector of weights for the different clusters. For example, it may be used if cluster 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 cost of labeling a cluster C_i with class $L(C_i)$ is defined as the number of samples from class $L(C_i)$ (in the labeled seed) that fall outside the cluster C_i, i.e.:

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

Applying Equation 2 to the total cost definition of Equation 1 yields:

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

In this paper, we applied the Hungarian algorithm to achieve the optimum cluster labeling in Equation 3. It requires the definition of a cost matrix $\mathbf{C}_{[kxk]}$ whose rows denote the clusters and the columns refer 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. $\mathbf{C}_{ij} = Cost(L(C_i) = j)$. The reader is referred to [6] for further details about the assignment problem and the Hungarian algorithm.

4. Extension through cluster pruning

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"[2] errors with respect to the real class labels. This happens especially when two or more of the underlying classes show a certain overlap 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 optimization method is to improve the (external) cluster quality by identifying and removing such regions with high probability of misclassification errors from the clusters. To this aim, we apply the concept of *pattern silhouettes* to prune the clusters in $\mathcal{C}$.

[1] Although class labels can take an arbitrary value, numeric or nominal, for the sake of simplicity we transformed the k class labels to integer values ($[1 \ldots k]$).

[2] Here, the term missclassification is not 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.

The silhouette width of an observation x_i is an internal measure of quality, typically used as the first step of the computation of the average silhouette width of a cluster partition [8]. It is formulated as:

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

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" with which the clustering algorithm has assigned pattern x_i to cluster $C(x_i)$. 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 mainly described with the following steps:

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. The observations in each cluster with lower silhouette scores may belong to a class-overlapping region with higher probabilities. Using the histograms of silhouette scores within the clusters, select a minimum silhouette threshold for each cluster.

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.

As aforementioned, the selection of silhouette thresholds are determined according to the histograms of silhouette values in each cluster. In this work, we estimate the distribution of silhouette values by using the histogram function in the software R which also provides the vectors of silhouette values found as the histogram bin limits and the counts of occurrences in each bin [3]. In practice, silhouette thresholds are selected to coincide with histogram bin limits. We set the selected number of histogram bins corresponding to rejected patterns to the largest possible satisfying the following 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 $1/3$ of the total number of patterns in the cluster.

5. Evaluation and results

The supervised and semi-supervised methods described in the previous section have been applied to a data set of transcribed utterances collected from user calls to commercial troubleshooting dialog systems.

Utterance preprocessing First, we preprocessed the utterance corpus using morphological analysis and stop word removal. The morphological analyzer [10] was applied to reduce the surface forms of words into their word lemmas. The lemmatized words were filtered using the SMART stop word list with small modifications. In particular, confirmation words (yes, no) were deleted from the stop word list, while some terms typical for spontaneous speech (eh, ehm, uh, ...) were added. Finally, we retained the lemmas with two or more occurrences in the preprocessed corpus, resulting in a vocabulary dimension of 554 word lemmas, also referred to as *index terms* in the following. After removing duplicate vectors, the final data set consisted of 2940 unique utterance vectors. From a total number of 79 symptoms, the following preliminary experiments used only the two most frequent symptoms in the training set (comprising 288 unique instances of utterance vectors). That means, we are speaking of a binary classification task.

[3] The bin sizes provided by the R's histogram function are estimated according to the Sturges formula [9]

Distance	Removed patterns	NMI 1	NMI 2	Error 1	Error 2
Overlap	31.94%	0.269	0.64	20.48%	6.63 %
Cosine	32.29%	0.100	0.297	36.11 %	21.02 %

Table 1: Normalized Mutual Information (NMI) before and after cluster pruning (referred to as NMI 1 and NMI 2). Misclassification errors before and after cluster pruning are denoted Error 1 and Error 2.

In addition, we prepared an independent test set comprising a total number of 10000 transcribed utterances using the same steps as described above. From this set, we randomly selected a number of utterances ($\sim 10\%$ of the training set size) as the test set applied to the classifiers. In order to avoid possible biases of a single test set, we generated 20 different test partitions. From the the training set, we also selected 20 different random seeds of labeled prototypes (n labels /category).

First, we evaluated the performance of the cluster pruning approach by computing the external cluster quality before and after cluster pruning (in terms of misclassification error rates and Normalized Mutual Information (NMI) [11] between the cluster partition and the reference labels. These scores can be observed in Table 1. While the pruned sections comprise around 30% of the total number of patterns in the data sets, the percentage of remaining misclassification errors has been substantially reduced from 20.48% to 6.63% in the clusters obtained with overlap distances and from 36.11% to 21.02% with cosine dissimilarities.

5.1. Results

For both supervised and semi-supervised SVM classifiers, we measured classification accuracy. The results are shown in Figures 1 and 2. Horizontal axes represent the sizes of the initial prototype seeds (from 1 to 5 labeled samples/class), vertical axes represent the mean accuracy scores, averaged over 400 experiments (20 test partitions x 20 prototype seeds).

The accuracy curves of the semi-supervised algorithm are roughly constant or slowly increasing with the labeled set size. In contrast, accuracy curves of the supervised approach show stronger increments with the training set sizes. By using the Pam algorithm with the cosine distance, the cluster quality is not sufficient to recover the underlying class structure, and thus, the supervised approach outperforms the semi-supervised method regardless of the labeled seed size. By applying the overlap distance (Figure 2), some improvements can be observed if cluster pruning is used to enhance the cluster quality (in terms of misclassification errors). In this case, the semi-supervised algorithm achieves higher performance than the supervised classifier under minimal labeled seeds ($n = 1$ or 2 samples/category). For larger values of n, the information in the increasing labeled seeds compensates for misclassification errors in the augmented sets, and thus, the supervised classifier outperforms again the

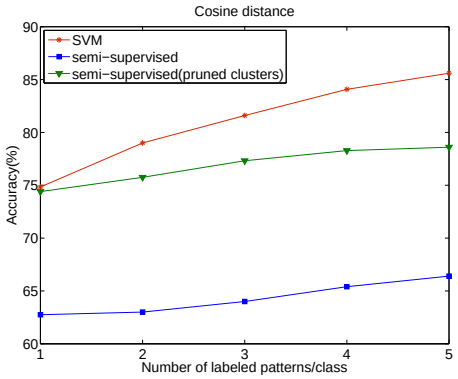

Figure 1: Accuracy with the supervised and semi-supervised methods using the cosine distance.

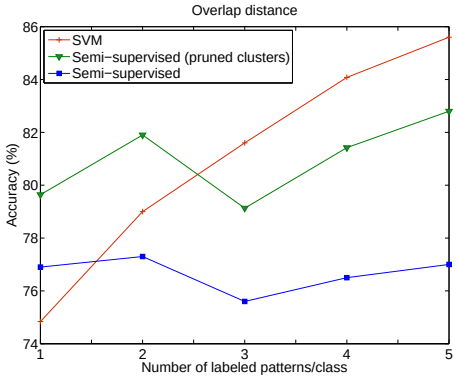

Figure 2: Accuracy with the supervised and semi-supervised methods using the overlap distance.

semi-supervised approach.

6. Conclusions and future directions

In this paper, we presented a semi-supervised cluster-and-label approach to classification of utterances has been presented. In contrast to previous works in semi-supervised classification literature where labels are commonly integrated in the clustering process, in this work, the cluster and labeling stages are independent of each other. First, an unsupervised clustering algorithm is used to obtain a cluster partition of the utterance training set. Then, the output cluster partition as well as a small set of labeled prototypes (also referred to as labeled seeds) are used to determine the optimum cluster labeling related to the labeled seed. We formulated the cluster labeling problem as an assignment optimization problem whose solution is obtained by means of the Hungarian algorithm. In addition, we improved the semi-supervised algorithm by discarding the patterns clustered with small silhouette scores. Thereby, we were able to demonstrate that the quality of the pruned clusters can be improved. This is because the removal of small numbers of cluster members can lead to significant reductions of missclassification errors.

Our experimental results showed improvements of the semi-supervised algorithm after cluster pruning for small labeled data sets ($n = 1$ and 2 labels/class), by applying the Pam clustering algorithm in conjunction with the overlap distance.

Future work is to analyze further alternatives for the definition of the cost matrix used by the Hungarian algorithm. For example, a probabilistic definition of the cost matrix by estimating class-cluster probabilities given the labeled seeds may help to extend the proposed semi-supervised approach to a larger number of categories.

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" with members of one category (lower cluster entropies).

7. References

[1] et al., K. A., "Technical support dialog systems: Issues, problems, and solutions," in [*Proc. of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*], (2007).

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

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

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

[5] Zhu, X., "Semi-supervised learning literature survey," (2006).

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

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

[8] Rousseeuw, P., "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Jornal Comp. Appl. Math.* **20**, 53–65 (1987).

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

[10] G. Minnen, J. C. and Pearce, D., "Applied morphological processing of english," *Natural Language Engineering* **7(3)** (2001).

[11] Strehl, A. and Ghosh, J., "Cluster ensembles — a knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.* **3**, 583–617 (2003).