14

Text Classification Aided by Clustering: a Literature Review

Antonia Kyriakopoulou
Athens University of Economics and Business
Greece

1. Introduction

Supervised and unsupervised learning have been the focus of critical research in the areas of machine learning and artificial intelligence. In the literature, these two streams flow independently of each other, despite their close conceptual and practical connections. In this work we exclusively deal with the text classification aided by clustering scenario. This chapter provides a review and interpretation of the role of clustering in different fields of text classification with an eye towards identifying the important areas of research. Drawing upon the literature review and analysis, we discuss several important research issues surrounding text classification tasks and the role of clustering in support of these tasks. We define the problem, postulate a number of baseline methods, examine the techniques used, and classify them into meaningful categories.

A standard research issue for text classification is the creation of compact representations of the feature space and the discovery of the complex relationships that exist between features, documents and classes. There are several approaches that try to quantify the notion of information for the basic components of a text classification problem. Given the variables of interest, sources of information about these variables can be compressed while preserving their information. Clustering is one of the approaches used in this context. In this vein, an important area of research where clustering is used to aid text classification is the area of dimensionality reduction. Clustering is used as a ***feature compression and/or extraction method***: features are clustered into groups based on selected clustering criteria. Feature clustering methods create new, reduced-size event spaces by joining similar features into groups. They define a similarity measure between features, and collapse similar features into single events that no longer distinguish among their constituent features. Typically, the parameters of the cluster become the weighted average of the parameters of its constituent features. Two types of clustering have been studied: i) one-way clustering, i.e. feature clustering based on the distributions of features in the documents or classes, and ii) co-clustering, i.e. clustering both features and documents.

A second research area of text classification where clustering has a lot to offer, is the area of ***semi-supervised learning***. Training data contain both labelled and unlabelled examples. Obtaining a fully labelled training set is a difficult task; labelling is usually done using human expertise, which is expensive, time consuming, and error prone. Obtaining unlabelled data is much easier since it involves collecting data that are known to belong to

one of the classes without having to label them. Clustering is used as a method to extract information from the unlabelled data in order to boost the classification task. In particularly clustering is used: i) to create a training set from the unlabelled data, ii) to augment the training set with new documents from the unlabelled data, iii) to augment the dataset with new features, and iv) to co-train a classifier.

Finally, ***clustering in large-scale classification problems*** is another major research area in text classification. A considerable amount of work is done on using clustering to reduce the training time of a classifier when dealing with large data sets. In particular, while SVM classifiers (see (Burges, 1998) for a tutorial) have proved to be a great success in many areas, their training time is at least $O(N^2)$ for training data of size N, which makes them non favourable for large data sets. The same problem applies to other classifiers as well. In this vein, clustering is used as a down-sampling pre-process to classification, in order to reduce the size of the training set resulting in a reduced dimensionality and a smaller, less complex classification problem, easier and quicker to solve. However, it should be noted that dimensionality reduction is not accomplished directly using clustering as a feature reduction technique as discussed earlier, but rather in an indirect way through the removal of training examples that are most probably not useful to the classification task and the selection of the most representative redundant training set. In most of the cases this involves the collaboration of both clustering and classification techniques.

The chapter is organized as follows: the next section presents a review of the literature on text classification aided by clustering. It provides a comprehensive summary of the alternative views and applications of clustering discussed above and their implications for text classification. A broader perspective on clustering and text classification research is then provided by discussing important research themes that emerge from the review of the literature and by classifying them into meaningful concept groups. We conclude by pointing out open issues and limitations of the techniques presented.

2. The literature review

2.1 Clustering as a feature compression and/or extraction method

Clustering as a feature compression and/or extraction method includes: i) one-way clustering, and ii) co-clustering.

2.1.1 One-way clustering (clustering features)

In text classification using one-way clustering, a clustering algorithm is applied prior to a classifier to reduce feature dimensionality by grouping together "similar" features into a much smaller number of feature clusters, i.e. clusters are used as features for the classification task replacing the original feature space. A crucial stage in this procedure is how to determine the similarity of features. Three main clustering methods have been applied in the literature: information bottleneck, distributional clustering, and divisive clustering.

An important feature clustering method that formulates a principle for the extraction and efficient representation of relevant information is the ***information bottleneck (IB)*** method (Tishby et al., 1999). The objective of the IB method is to extract the information from one variable X that is relevant for the prediction of another variable Y. In other words, the method finds an efficient compressed representation of the variable X, denoted X', such that

the predictions of Y from X through X' is as close as possible to the direct prediction of Y from X. The compactness of the representation is determined by the mutual information $I(X;X')$ while the quality of the clusters is measured by the fraction of information they capture about Y, that is $I(X';Y)/I(X;Y)$. Obviously there is a trade-off between compressing the representation and preserving meaningful information. The desirable is to keep a fixed amount of meaningful information about the relevant variable, Y, while minimizing the number of bits from the original variable X (maximizing the compression). In an alternative ***agglomerative implementation of the IB*** method, (Slonim & Tishby, 1999) attain maximum mutual information per cluster between feature data and given categories. This implementation can be considered as a bottom-up hard implementation of the original top-down soft hierarchical IB method. They demonstrate the algorithm on a subset of 20Newsgroups corpus, achieving compression by 3 orders of magnitude while maintaining about 90% of the original mutual information. The IB clustering method with its variations is used in the context of text classification by many authors. In this vein the classifier is applied in a reduced space where features represent clusters.

More specifically, in (Slonim & Tishby, 2001) the IB clustering method is used together with the Naive Bayes (NB) classifier. First, the feature clusters that preserve the information about the classes as much as possible are found using the agglomerative IB method. Then these clusters are used to represent the documents in a new, low dimensional feature space and the NB classifier is applied on this reduced space. Results from 20Newsgroups corpus show that when the size of the data sets is large, using feature clusters does not improve significantly the classification performance. However, when a small sample training set is used the method yields a significant improvement in classification accuracy, from 5% to 18%, compared to using the original feature space. (Verbeerk, 2000a, 2000b) applies the minimum description length (MDL) (Rissanen, 1989) principle to the agglomerative algorithm of (Slonim and Tishby, 2001) in order to define the number of clusters to be used for the classification task.

(Bekkerman et al., 2001, 2003) compare two classification schemes based on two representations: the simple, typical bag-of-words (BOW) representation (Salton & McGill, 1983) together with mutual information feature selection, and a representation that is based on feature clusters computed via the IB method. The comparison is performed over 20NG, Reuters-21578 and WebKB with SVMs used for the classification task. The results of the experiments are contradictory revealing a sensitivity of the algorithm to the datasets.

(Mubaid & Umair, 2006) use the IB clustering method with a least squares (Felici & Truemper, 2002) classifier. The method has been tested with the WebKB, 20NG and Reuters-21578 datasets and is compared against SVM. The experimental results show that the performance of the method is equally good and in some cases outperforms SVM, especially when there is limited training data.

(Baker & McCallum, 1998) apply ***distributional clustering*** as a feature clustering method for text classification. Distributional clustering (Pereira et al, 1993) is a special case of the general IB clustering algorithm as it is shown in (Slonim and Tishby, 2001). The similarity between two features, f_t and f_s , is measured as the similarity between the class variable C distribution they induce: $P(C|f_t)$ and $P(C|f_s)$. In the case of text classification, the similarity of two features is the similarity between their joint distributions with the category variable. For clustering this means that features with similar distributions over the classes (should) belong to the same cluster. Intuitively, if two different features have similar distributions

over the classes, they will play a similar role in the classification process, and thus might as well be clustered together. Using a Naïve Bayes classifier for the classification task, they compare their method with feature selection methods such as Latent Semantic Indexing, class-based clustering, mutual information, and Markov-blanket-based feature selection (Koller & Sahami, 1996). Their results show that distributional clustering outperforms the other methods by drastically reducing the number of features, achieving compression by 3 orders of magnitude, while loosing only 2% classification accuracy. An interesting outcome concerns the application of a feature selection method prior to the feature clustering method. It actually improves the feature clustering method, suggesting that there is place for combinations of the two methods.
(Dhillon et al, 2003a) propose an information-theoretic feature clustering algorithm, termed as ***divisive clustering***, and apply it to text classification. The method derives a global objective function that explicitly captures the optimality of feature clusters in terms of a generalized Jensen-Shannon divergence (Lin, 1991) between multiple probability distributions. Then a fast, divisive algorithm that monotonically decreases this objective function value is applied. The algorithm has many good qualities. Is optimises over all clusters simultaneously and it is much faster than the agglomerative strategies proposed by (Baker & McCallum, 1998) and (Slonim & Tishby, 2001) obtaining better feature clusters. Experiments using the Naive Bayes and SVM classifiers on the 20 Newsgroups and Dmoz data sets show that divisive clustering improves classification accuracy especially at lower number of features. When the training data is sparse, divisive clustering achieves higher classification accuracy than the maximum accuracy achieved by feature selection strategies such as information gain and mutual information.
(Lavelli et al., 2004) carry out experiments on feature classification tasks (i.e. grouping together features according to their meaning into prespecified classes) and feature clustering tasks in order to compare the two representations. Also, (Lewis, 1992) studies the properties of clustered feature representations on a text classification task. See (Jain et al., 1988) for a comprehensive survey on one-way clustering.

2.1.2 Co-clustering (clustering features and documents)

Using co-clustering in text classification, a two-stage procedure is usually followed: feature clustering and then document clustering. In this way a reduction for both dimensions is attained.
The ***double-clustering (DC)*** algorithm (Slonim & Tishby, 2000) is a co-clustering two-stage procedure based on the IB method. Intuitively, in the first stage of DC, feature clustering generates coarser pseudo features, which reduce noise and sparseness that might be exhibited in the original feature space. Then, in the second stage, documents are clustered as distributions over the "distilled" pseudo features, and therefore generate more accurate document clusters. An extension of the DC algorithm, the so called ***Iterative Double Clustering (IDC)*** (Yaniv & Souroujon, 2001) applies the DC algorithm in an iterative manner. Whenever the first DC iteration succeeds in extracting a meaningful structure of the data, a number of the next consecutive iterations can continually improve the clustering quality. This is achieved due to the generation of progressively less noisy data representations. Experiments conducted on text classification tasks indicate that IDC outperforms DC and competes even SVM when the training set is small. The works of (Slonim & Tishby, 2000), (Slonim et al., 2001), (Yaniv & Souroujon, 2001) use heuristic

procedures to cluster documents and features independently using an agglomerative algorithm.

(Dhillon et al, 2002, 2003b) on the other hand, propose an ***information-theoretic co-clustering*** algorithm that intertwines both row (feature) and column (document) clustering. The algorithm starts with a random partition of rows, *X*, and columns, *Y*, and computes an approximation *q(X,Y)* to the original distribution *P(X,Y)* and a corresponding compressed distribution by co-clustering rows and columns intertwined, i.e. the row-clustering incorporates column-clustering information and vice versa. The algorithm iterates until it almost accurately reconstructs the original distribution, discovers the natural row and column partitions and recovers the ideal compressed distribution. Experiments conducted demonstrate the efficiency of the algorithm especially in the presence of sparsity.

(Dai et al., 2007) extend the co-clustering algorithm of (Dhillon et al., 2002, 2003b) and present a ***co-clustering classification algorithm*** (CoCC) that focuses on classifying documents across different text domains. There is a labelled data set D_i from one domain, called *in-domain*, and an unlabelled data set D_o from a related but different domain, called *out-of-domain*, that is to be classified. The two datasets are drawn from different distributions, since they are from different domains. The algorithm is based on two assumptions. First, the set *C* of class labels in D_i prescribes the labels to be predicted in D_o. Second, even though the two domains have different distributions, they are similar in the sense that similar words describe similar categories, thus, the probability of a class label given a word is very close in the two domains. The algorithm applies co-clustering between all features and out-of-domain documents (new tasks) in D_o. Feature clustering is constrained by the labels of in-domain (old) documents *Di*. The feature clustering part in both domains serves as a bridge. For the classification task, each out-of-domain cluster is mapped to a corresponding class label based on the correlation with the document categories in D_i.

The idea of clustering features and documents to improve text classification is also pursued in (Takamura & Matsumoto, 2002; Takamura, 2003). They empirically show that the assumption that documents in the same category are generated from an independent identical distribution is inaccurate, and propose a new method called ***two-dimensional clustering*** to alleviate this problem. According to this method, training examples are first clustered so that the i.i.d. assumption is more likely to be true and features are also clustered in order to deal with the data-sparseness problem caused by the high dimensionality of the feature space. Two classifiers (NB and SVM) are trained on the training examples of each cluster and the testing examples are classified and assigned the label of the class of the cluster (all training examples in each cluster are supposed to have the same class label). The comparison of the method with distributional clustering (Baker & McCallum, 1998) and feature clustering on Reuters-21578 and 20NG shows promising results.

Table 1 summarizes the methods presented in this section.

2.2 Clustering in semi-supervised classification

Clustering in semi-supervised classification is used as a method to extract information from the unlabelled data in order to boost the classification task. In particularly clustering is used: i) to create a training set from the unlabelled data, ii) to augment the training set with new documents from the unlabelled data, iii) to augment the dataset with new features, and iv) to co-train a classifier.

Goal	Authors	Clustering method
One-way clustering: cluster feature space and replace it with a feature cluster representation	(Baker & McCallum, 1998)	Distributional clustering
	(Slonim & Tishby, 2001)	IB
	(Verbeerk, 2000a, 2000b)	Agglomerative IB
	(Bekkerman et al., 2001, 2003)	Agglomerative IB
	(Mubaid & Umair, 2006)	IB
	(Dhillon et al, 2003a)	Divisive clustering
Co-clustering: cluster both features and documents	(Yaniv & Souroujon, 2001)	Iterative double clustering
	(Dhillon et al, 2002, 2003b)	Information-theoretic co-clustering
	(Dai et al., 2007)	Co-clustering classification
	(Takamura & Matsumoto, 2002);(Takamura, 2003)	Two-dimensional clustering

Table 1. Clustering as a feature compression and/or extraction method

2.2.1 Create a training set from the unlabelled data

(Fung and Mangasarian, 2001) propose a model for classifying two-class unlabelled data, called ***clustered concave semi-supervised SVM (CVS^{3}VM)***. First, a *k*-median clustering algorithm finds *k* cluster centres for the unlabelled examples such that the sum of distances between each example and the closest cluster centre is minimized. Then, examples within a certain distance from these *k* cluster centres are treated as representative examples of the clusters, and hence of the overall dataset, and are given to an expert or oracle to label. Finally, a linear SVM is trained using this small sample of labelled data. The model is effectively compared to other methods.

(Li et al., 2004) follow a similar approach where a *k*-means clustering algorithm is used to cluster the unlabelled data into a certain number of subsets and to assign corresponding cluster labels. Then, an SVM classifier is trained on this labelled set.

2.2.2 Augment the training set with new documents from the unlabelled data

The ***clustering based text classification (CBC)*** approach (Zeng et al., 2003) improves classification performance by using unlabelled data, *U*, to augment the training, labelled data, *L*. According to this method a clustering algorithm is first applied to *L*. For each class, the centroids of the labelled data are computed and used as the initial centroids for *k*-means. The *k* value for *k*-means is set to the number of classes in the classification task. Accordingly, the label of each centroid is equal to the label of the corresponding examples of each class. Then, *k*-means runs for both *L* and *U* and *k* clusters are created. The most confident examples from each cluster (i.e. the ones nearest to the cluster's centroid) are added to *L*. This is considered to be a soft-constrained version of *k*-means because the constraints are not based on exact examples but on their centroid, thus reducing the bias in *L*. Finally, the augmented *L* and the rest of *U* are used to train and test a Transductive SVM (TSVM) classifier. Their experimental results demonstrate that CBC outperforms existing algorithms, such as TSVMs and co-training, especially when the size of the labelled dataset is very small.

(Chapelle et al., 2002) propose a framework to incorporate unlabelled data in a kernel classifier based on the "cluster assumption", i.e. nearby points are likely to have the same class label, and two points are likely to have the same class label if they belong to the same cluster. Using spectral methods (Spielman & Teng, 1996; Ng et al., 2002) they show how to design kernels such that the induced distance is small for points in the same cluster and large for points in different clusters. This representation with the points naturally clustered, is then used to train a discriminative learning algorithm. The testing set, if available during training, can be considered as unlabelled data; therefore spectral clustering is applied to training, unlabelled and testing data. Otherwise, an approximation of each testing example as a linear combination of the training and unlabelled data is computed. The experiments show encouraging results. The algorithm is applicable to a purely supervised learning task.
(Zhou et al., 2003) also base their method on the "cluster assumption" and apply spectral clustering to represent the labelled and unlabelled data. The keynote of the method is to let every labelled point in the representation iteratively spread its label information to its neighbours until a global stable state is achieved. Then, the label of each unlabelled point is set to be the class of which it has received most information during the iteration process. The algorithm demonstrates effective use of unlabelled data in experiments including digital recognition and text categorization.

2.2.3 Augment the dataset with new features

Unlike direct methods like CBC, which label the unlabelled data, the technique of (Raskutti et al., 2000a), augments the feature space with new features derived from clustering the labelled and unlabelled data. A non-hierarchical single-pass clustering algorithm is used to cluster labelled and unlabelled examples. In order to derive only the useful information from the clusters, the clusters are sorted by their sizes, and the largest N clusters are chosen as representatives of the major concepts. Each cluster contributes the following features to the feature space of the labelled and the testing examples: i) a binary feature indicating if this is the closest of the N clusters, ii) similarity of the example to the cluster's centroid, iii) similarity of the example to the cluster's unlabelled centroid, i.e. the average of the unlabelled examples that belong to the cluster, and iv) for each class in the labelled set, similarity of the example to the cluster's class l-centroid defined as the average of the examples in class l that belong to this cluster. The clusters are thought of as higher level "concepts" in the feature space, and the features derived from the clusters indicate the similarity of each document to these concepts. The unlabelled data are used to improve the representation of these concepts. They evaluate the method using SVM classifiers on well-known corpora, and find significant improvements in the classification performance.
In (Kyriakopoulou & Kalamboukis, 2007) the training and testing sets are augmented with new features derived from clustering without using unlabelled data. Consider a k-class categorization problem, (k>=2), with a labelled l-training sample $\{(x_1, y_1),\ldots,(x_l, y_l)\}$ of feature vectors $x \in R^n$ and corresponding labels $y_i \in \{1, \ldots, k\}$, and an unlabelled m-testing sample $\{(x_1^*,\ldots,x_m^*\}$ of feature vectors. The approach consists of three steps: clustering, expansion and classification step. In the clustering step, the number of clusters is chosen to be equal to k, i.e. the predefined number of classes. A divisive clustering algorithm with repeated bisections is selected to cluster both training and testing sets. In the expansion step, each cluster contributes one *meta*-feature to the feature space of the training and testing sets: given the total n features that are used in the representation of the l+m feature vectors, and

the k clusters derived from the clustering step, create *meta*-features $x_{n+1},\ldots,x_{n+k}$. A document x in the cluster C_j is characterized by the *meta*-feature x_{n+j}. Finally, in the classification step, linear SVM/transductive SVM classifiers are trained on the expanded training set and classify the expanded testing set. Evaluation of this approach using several widely used corpora indicates that it is extremely useful improving the classifier's performance especially when the number of the training examples is very small. The algorithm has also been successfully used in a spam-filtering setting (Kyriakopoulou & Kalamboukis, 2006). Also, it can be directly applied to a purely semi-supervised task using unlabelled data as an additional source of information.

In (Takamura, 2003) given the co-occurrences of features and documents of the training set, the features are first hard clustered. Let H be the reduced matrix resulting from clustering. The relation between a feature vector d and its reduced vector s is $Hd=s$. Next, the two vectors are concatenated into a vector d'. Then, the testing set is classified with SVM using d' as input. Takamura explains how the expansion of the feature space is equivalent to using a special kernel in the original feature space, where the form of the mapping to a higher dimensional space depends on the given data. Experiments conducted on Reuters-21578 and 20NG show that the method is effective especially when the training set is small.

2.2.4 Co-training

In general, a co-training algorithm produces an initial weak classifier from a few labelled examples and later uses unlabelled data to improve its performance. The idea was first introduced in (Blum & Mitchell, 1998). The key defining features of this problem class are that (i) the features can be factored into two (or more) components, i.e. there are two distinct views of an example x, which are redundantly sufficient to correctly classify the example, and (ii) the two components are independent and identically distributed, so that the features in one view of the example x do not always co-occur with the features in the second view. A different approach to co-training is given in (Goldman & Zhou, 2000). See (Abney, 2002; Seeger, 2000) for a comprehensive survey on co-training.

The use of "concepts" derived by clustering as in (Raskutti et al., 2000a) provides an alternate description of the data, similar to the redundant views used in co-training. In this vein, (Raskutti et al., 2002b) present a co-training strategy to make use of unlabelled data. Two predictors are trained in parallel, and each predictor labels the unlabelled data to train the other predictor in the next round. The process repeats for a number of iterations. The predictors are SVMs, one trained using the original word presence features view, and the other trained with solely the new cluster features that are derived by clustering both labelled and unlabelled data. The new features include membership information as well as similarity to clusters' centroids for the more populous clusters as described in their previous work (Raskutti et al., 2000a). This new feature space creates an alternative redundant view of the data as imposed by the co-training framework of (Blum & Mitchell, 1999). They evaluate the method using SVM classifiers on Reuters-21578, 20Newsgroups, and WebKB corpora. Their results are encouraging and confirm previous findings.

A different co-training approach is based on co-training between clustering and classification (Kyriakopoulou, 2007). Unlike the procedure in (Blum & Mitchell, 1999) it does not require a priori the existence of two distinct properties of the underlying data distribution in order to work. Also, it doesn't use two different supervised learning algorithms that complement each other as in (Goldman & Zhou, 2000). Instead, there is one

original feature space, which is used interchangeably by an unsupervised and a supervised learning algorithm, and each algorithm augments it by propagating its results in the form of corresponding *meta*-features. Specifically, following the procedure in (Kyriakopoulou & Kalamboukis, 2007), at every round of co-training a "hard" clustering algorithm groups the examples of the training and testing sets into k clusters. The examples that belong to the same cluster are augmented with a *meta*-feature that denotes membership information to this cluster. Then a separate SVM classifier for each class of the classification task is build from the augmented feature space. Each SVM classifier returns a prediction for each example, which is interpreted as the likelihood that the example belongs to a certain class. The predictions of the underlying classifiers for each example are compared and each example is assigned the label of the class with the highest prediction. The labels information is translated into *meta*-features that are used to augment the feature space and the algorithm iterates. According to experimental findings the combination of clustering with classification in a co-training setting, and the addition of corresponding *meta*-features, are successfully used as an additional source of information about margins. The experimental results on widely used datasets demonstrate the superiority of the approach over SVMs.
Table 2 summarizes the methods presented in this section.

2.3 Clustering in large-scale classification problems

Clustering in large-scale classification problems is used as a down-sampling pre-process to classification, in order to select the most representative training examples according to: i) clustering and information from the resulting hyperplane of a SVM initially trained on cluster representatives, ii) clustering and prior class label information, iii) a combination of cases i and ii, iv) solely clustering results, and v) problem decomposition.

2.3.1 Select most representative training data according to clustering and information from the resulting hyperplane of a SVM initially trained on cluster representatives

In this case, first, the training examples are clustered. Then, cluster representatives (clusters' centroids) are used to train an initial SVM classifier. Next, follows a process that selects the clusters that contain the most representative training examples according to a combination of the clustering and classification results. Usually, this process is called declustering and corresponds to an expansion of the training set according to clustering (i.e. the examples of a cluster are no longer represented by the cluster's centroid; instead all the examples are considered). Lastly, a SVM is trained on the new training set. The following algorithms differ in the selection of the cluster representatives, and the way the clustering and classification results are combined in order to select the clusters that contain the best candidates from the training examples. In concluding, they exploit the distributional properties of the training data, i.e. the natural clustering of the training data, and the overall layout of these clusters relative to the decision boundary of SVMs.

- The ***clustering-based SVM (CB-SVM)*** method (Yu et al., 2003) uses the hierarchical clustering technique named BIRCH (Zhang et al., 1996) to cluster the training examples. The key idea of CB-SVM is to use a hierarchical clustering algorithm to get a finer description of the training data closer to a SVM decision boundary and a coarser description away from it. Let T_p and T_n be the hierarchical trees built from the positive and the negative training examples respectively. Then, a SVM is trained from the centroids of the root nodes (i.e. clusters) of T_p and T_n. According to the solution of the

SVM, the clusters whose centroids are support vectors for the SVM and the clusters that are very close to the support vectors (satisfying a certain distance constraint) are declustered into the finer level using the tree structure. These clusters may introduce new support vectors for the SVM, and are thus accumulated into the training set. A new SVM is constructed from the augmented training set, and the declustering process is repeated until nothing is accumulated, i.e. this selective declustering procedure reaches leafs' level. Experiments show that CB-SVM is scalable for very large data sets while also generating high classification accuracy.

Goal	Authors	Clustering/ Classification method	Basic method
Create a training set from the unlabelled data	(Fung & Mangasarian, 2001)	k-means/linear SVM	Unlabelled data selected by k-means are labelled by an oracle or expert.
	(Li et al., 2004)	k-means/linear SVM	Unlabelled data selected by k-means are labelled by cluster labels.
Augment the training set with new documents from the unlabelled data	(Zeng et al., 2003)	k-means/TSVM	Training and unlabelled data are clustered. Unlabelled data nearest to clusters' centroids are added to the training set.
	(Chapelle et al., 2002)	Spectral analysis	Creation of diagonal matrix that contains clustering information.
	(Zhou et al., 2003)		
Augment the dataset with new features	(Raskutti et al., 2000a)	non-hierarchical single-pass clustering algorithm/SVM	Training and unlabelled data are clustered. Each cluster contributes new features to the feature space of the training and testing examples.
	(Takamura, 2003)	hard clustering	The features of the training set are clustered.
	(Kyriakopoulou & Kalamboukis, 2006; Kyriakopoulou & Kalamboukis, 2007)	divisive clustering algorithm /SVM	Training and testing data are clustered. Each cluster contributes a new feature to the feature space of the training and testing examples
Co-training	(Raskutti et al., 2000b)	non-hierarchical single-pass clustering algorithm/SVM	Clustering creates a redundant view in a co-training framework
	(Kyriakopoulou, 2007)	divisive clustering algorithm/SVM	Clustering is used as unsupervised classifier in a co-training framework.

Table 2. Clustering in semi-supervised classification

- (Awad et al., 2004) also apply a hierarchical clustering algorithm, called dynamically growing self-organizing tree (DGSOT) (Luo et al.), as a reduction method of the training set for SVM classification. The authors propose two alternatives to train a SVM for two classes based on the combination of DGSOT and SVM. The first approach generates two hierarchical trees, one for each class, up to a certain level, i.e. they are not fully grown. Then, a SVM is trained on the clusters' references of the trees top nodes (clusters). After computing the margin, the nodes that contain a support vector are declustered by adding their children nodes to the training set. The process of training and declustering is repeated until a stopping criterion holds. In the second approach, one more step is added to the previous procedure before declustering. Specifically, the distance between nodes in the training set is measured. Since the distance between nodes lying in the decision boundary area is the least, the nodes having distance more than the average are excluded. Unlike the approach of (Yu et al., 2003), that first builds the hierarchical tree and then starts to train the SVM, in this approach clustering goes in parallel with training the SVM. During the tree construction and declustering process, DGSOT re-distributes data among newly added children of a node and re-evaluates clustering results. The growth of the tree is controlled, because non-support vector nodes will be stopped from growing, and only support vector nodes will be allowed to grow. Experiments on several datasets against other relevant techniques give contradictory results. The second approach outperforms the rest but needs more time. Also, the algorithm is sensitive on the initial small training set, giving high error rates at the beginning of the training process, which is not fully recovered till the end.
- ***ClusterSVM*** (Boley & Cao, 2004) partitions the training data into pair-wise disjoint clusters using adaptive clustering. Then, a SVM is trained using the centroids of these clusters. Based on this initial SVM, it can be judged whether a cluster contains either only support vectors or only non-support vectors. The clusters that contain both support vectors and non-support vectors based on the decision boundary of the initial SVM are repeatedly divided into sub-clusters that approximately contain either only non-support vectors or only support vectors. Clusters having only non-support vectors are replaced by their representatives. Experiments on artificial and real world datasets prove the efficiency of clusterSVM over popular algorithms such as SMO.
- A similar approach named ***support cluster machines (SCMs)*** (Yuan et al., 2006) uses *k*-means to partition the negative training examples into disjoint clusters, and then trains an initial SVM using the positive examples and the representatives of the negative clusters. With the global picture of the initial SVM, it can approximately identify the support vectors and non-support vectors. A shrinking technique is then used to remove the examples, which are most probably not support vectors. This procedure of clustering and shrinking is performed iteratively until some stopping criteria are satisfied.
- The ***kernel based incremental clustering algorithm (KBIC)*** method uses a scalable kernel based clustering algorithm for the selective sampling based training of non-linear SVMs (Asharaf et al., 2007). This is a two-phase algorithm. In the first phase, KBIC is used to generate a high level description of the data (clusters) in an appropriate kernel induced feature space. The cluster prototypes obtained are used to train a SVM and the corresponding support vectors are identified. In the second phase, a declustering process that expands all the clusters near the boundary creates the training set for the subsequent training of a SVM.

2.3.2 Select most representative training data according to clustering and prior class label information

In this case, the selection of the representative training examples is determined by the composition of the clusters according to the available class label information.

- (Almedia et al., 2000) group the training data in k clusters using k-means. Clusters formed only by examples that belong to the same class label are disregard and only cluster centres are used. On the other hand, clusters with examples belonging to more than one class labels are unchanged and all training examples are considered. Clusters with mixed composition are likely to happen near the separation margins and they may hold some support vectors. Consequently, the number of training examples for the SVM training is smaller and the training time can be decreased without compromising the generalization capability of the SVM.
- (Fang et al., 2002) apply a clustering approach based on principal component analysis named principle direction divisive partitioning (PDDP) to cluster the training examples. The goal is to minimize noise effects in the training procedure by using those examples that are part of pure clusters, i.e. the ones that are dominated by one of the categories. The training examples that are clustered in pure nodes are used to seed a Naïve Bayes classifier. The authors evaluate the performance of the methods against several interesting variants and show improvements on classification performance.
- (Awad et al., 2004) apply the DGSOT hierarchical clustering algorithm to generate a hierarchical clustering tree from the training examples, and determine the most qualified nodes to decluster based on the heterogeneity of nodes. Heterogeneous nodes are those nodes that have data points assigned to them from different classes, thus, they are more likely to lie in the marginal area between two classes. Then, a SVM is trained on the training examples of the declustered nodes. Experiments on several datasets against other relevant techniques did not give satisfactory results.
- (Cervantes et al., 2006) apply SVM classification based on fuzzy partitioning clustering. The original training set is fuzzy clustered into k clusters with respect to a given criterion. The clusters obtained have elements of mixed category or uniform category. SVM is trained on the centroids of the clusters with mixed category elements, because these elements have bigger likelihood to be support vectors. Getting the clusters closer to the decision hyperplane and eliminating the clusters far away reduces the original data set. Then a de-clustering is applied to the reduced clusters and subsets from the original data set are obtained. Finally, SVM is used again and finishes classification. The experimental results show that the number of support vectors obtained using the SVM classification based on the fuzzy partitioning is similar to the normal SVM approach while training time is significantly smaller. However, the number of the clusters k is user-defined in order to avoid computational cost for determining the optimal number of clusters.
- (Li et al, 2007) propose the ***support cluster machine algorithm (SCM)*** to effectively deal with large-scale classification problems. It is a classification model built for clustering. Based on the learning framework of SVMs it defines clustering as a dual optimisation problem with a decision function formulised following the same steps as in SVMs. The goal is to maximize the margin between the positive and the negative clusters of a class, i.e. between clusters obtained only from the positive examples of a class and clusters obtained only from the negative examples accordingly. The examples are clustered using the threshold order-dependent (TOD) clustering algorithm (Friedman & Kandel, 1999). After clustering (or training phase), the training support clusters obtained can be directly used in the decision function to measure the similarity between a cluster and a testing example. The experimental results confirm that the SCM is very effective for

large-scale classification problems due to significantly reduced computational costs for both training and testing and comparable classification accuracies.

2.3.3 Select most representative training data according to clustering, information from the resulting hyperplane of a SVM initially trained on cluster representatives, and prior class label information

This case combines the two previous cases.

- ***Minimum enclosing ball clustering (MEB)*** (Cervantes et al., 2008) employs the concept of core-sets (Badoiu et al., 2002)(Kumar et al., 2003) over the training examples, L. The obtained clusters are of the following type: (i) clusters with only positive training examples, Ω^+, (ii) clusters with only negative training examples, Ω^-, and (iii) clusters with both positive and negative examples (or mix-labelled), Ω_m. MEB is used as a data selection method. To this end, only the centres of the Ω^+ and Ω^- clusters and all the examples of the mix-labelled Ω_m clusters are selected to form a reduced training set, L_r, used to train a SVM classifier with the sequential minimal optimisation (SMO) algorithm (Platt, 1998). Then, a de-clustering process augments L_r by including the examples in the clusters whose centres are support vectors of the classifier's solution. Taking the recovered data as new training data set, SVM classification with SMO algorithm is used again to get the final decision hyperplane. The experimental results show that the accuracy obtained by the approach is very close to the classic SVM method, while the training time is significantly shorter, enabling it to successfully handle huge data sets.

2.3.4 Select most representative training data according to solely clustering results

Various assumptions about the clustering results and the information they carry are adopted in order to build the redundant training set.

- (Sun et al., 2004) use k-means to cluster the input space. Because the data that decisively affect SVM classifiers are those at boundary of each class, it is assumed that the data residing on the boundaries of the clusters are critical data that together with the centroid of each cluster are used to train a SVM.
- (Wang et al., 2005) also combine the *k*-means clustering technique with SVM to build classifiers. *K*-means runs on the original training data and all cluster centres are regarded as the compressed data for building classifiers. Accordingly, SVM classifiers are built on the compressed data. The experiments show that it is possible for the algorithm to build classifiers with many fewer support vectors and higher response speed than SVM classifiers. Moreover, testing accuracy of the resulting classifiers can be guaranteed to some extent. This method also employs a parameter tuning method to achieve the required generalization performance at acceptable response time.

2.3.5 Problem decomposition

There are several decomposition methods that try to modify the SVM algorithm so that it can be applied to large datasets.

- The ***clustering support vector machines model (CSVMs model)*** (He at al., 2006) is different from the previous algorithms in this section in that all the training examples are kept during the training process. Using the theory of granularity computing the CSVMs model is able to divide a complex problem into a series of smaller and computationally simpler problems. To accomplish this a *k*-means clustering algorithm is used to cluster the training set into sub-clusters upon which SVMs are subsequently trained in parallel.

Table 3 summarizes the results from this section.

Goal	Authors	**Clustering method**	**Training sample selected or removed**
Select most representative training data according to clustering and information from the resulting hyperplane of a SVM initially trained on cluster representatives (1)	(Yu et al., 2003)	BIRCH	The clusters whose centroids are support vectors for the SVM and the clusters that are very close to the support vectors are declustered
	(Awad et al., 2004)	dynamically growing self-organizing tree	i) Clusters containing support vectors are declustered ii) Distant clusters are removed
	(Boley & Cao, 2004)	adaptive clustering	Clusters having only non-support vectors are replaced by their representatives
	(Yuan et al., 2006)	k-means	Clusters having only non-support vectors are removed
	(Asharaf et al., 2007)	kernel based incremental clustering	Clusters near the boundaries are declustered
Select most representative training data according to clustering and prior class label information (2)	(Almedia et al., 2000)	k-means	Clusters formed by examples that belong to the same class label are disregard and only cluster centres are used. All training examples from clusters of mixed composition are considered.
	(Fang et al., 2002)	principle direction divisive partitioning	Clusters formed by examples that belong to the same class label are considered.
	(Cervantes et al., 2006)	fuzzy partitioning clustering	Clusters of mixed class label composition are declustered and all training examples are considered.
	(Awad et al., 2004)	dynamically growing self-organizing tree	
	(Li et al, 2007)	TOD clustering algorithm	Support clusters obtained in the training phase are directly used in the decision function
Select most representative training data according to (1) and (2)	(Cervantes et al., 2008)	minimum enclosing ball clustering	The centroids of clusters with only positive or only negative training examples, all the examples of clusters with mixed composition, all the examples of the clusters whose centroids are support vectors are used.
Select most representative training data according to solely clustering results	(Sun et al., 2004)	k-means	The centroids and the training data residing at the boundaries of the clusters are selected.
	(Wang et al., 2005)	k-means	The centroids of the clusters are selected.
Problem decomposition	(He at al., 2006)	clustering support vector machines model	All training examples are used. The training set is clustered into subclusters upon which SVMs are subsequently trained in parallel.

Table 3. Clustering in large-scale classification problems

3. Conclusions and future directions

We presented several clustering methods for dimensionality reduction to improve text classification. Experiments show that one-way clustering is more effective than feature selection, especially at lower number of features. Also, when dimensionality is reduced by as much as two orders of magnitude the resulting classification accuracy is similar to a full-feature classifier. In some cases of small training sets and noisy features, feature clustering can actually increase classification accuracy. In the case of IB, various heuristics can be applied in order to obtain finer clusters, greedy agglomerative hard clustering (Slonim & Tishby, 1999), or a sequential K-means like algorithm (Slonim et al., 2002). Co-clustering methods are superior to one-way clustering methods as shown through corresponding experiments (Takamura, 2003). Benefits of using one-way clustering and co-clustering as a feature compression and/or extraction method include: useful semantic feature clusters, higher classification accuracy (via noise reduction), and smaller classification models. The second two reasons are shared with feature selection, and thus clustering can be seen as an alternative or a complement to feature selection, although it does not actually remove any features. Clustering is better at reducing the number of redundant features, whereas feature selection is better at removing detrimental, noisy features. The reduced dimensionality allows the use of more complex algorithms, and reduces computational burden. However, it is necessary to experimentally evaluate the trade-off between soft and hard clustering. While soft clustering increases the classification model size, it is not clear how it affects classification accuracy. Other directions for exploration include feature weighting and combination of feature selection and clustering strategies.

There are four cases of semi-supervised classification using clustering considered in the area. In the first case, in the absence of a labelled set, clustering is used to create one by selecting unlabelled data from a pool of available unlabelled data. In the second case, it is used to augment an existing labelled set with new documents from the unlabelled data. In the third case, the dataset is augmented with new features derived from clustering labelled and unlabelled data. In the last case, clustering is used under a co-training framework. The algorithms presented demonstrate effective use of unlabelled data and significant improvements in classification performance especially when the size of the labelled set is small. In most experiments, the unlabelled data come from the same information source as the training and testing sets. Since the feature distribution of the unlabelled data is crucial to the success of the method, an area of future research is the effect of the source and nature of information in the unlabelled dataset and clustering.

Lastly, clustering reduces the training time of the SVM i) by modifying the SVM algorithm so that it can be applied to large data sets, and ii) by finding and using for training only the most qualified training examples of a large data set and disqualifying unimportant ones. A clustering algorithm and a classifier cooperate and act interchangeably and complementary. In the first case, many algorithms have been proposed (sequential minimal optimisation, projected conjugate gradient, neural networks amongst others) in order to simplify the training process of SVM, usually by breaking down the problem into smaller sub-problems easier to solve. In the second case, the training set is clustered in order to select the most representative examples to train a classifier instead of using the whole training set. The clustering results are used differently by the various approaches, i.e. the selection of the representative training examples follows different methods. Some of the proposed algorithms manage to decrease the number of training examples without compromising the

generalization capability of the SVM. However, there were other approaches that gave contradictory results revealing the difficulty of the problem under examination.
Some methods are applied only on linear problems. Even though some of them can also be used to train non-linear SVMs, the iterative nature of their cluster generation/exploration strategy makes them very expensive to be used in large-scale datasets. There is a need for methods that perform a small number of data scans in order to work. Incremental clustering can also come in useful. Constructing effective indexing structures for non-linear kernels is an interesting direction of future work since it has high practical value especially for pattern recognition of large data sets. Developing an effective indexing structure for high dimensional problems is an interesting direction of future work.
Another important topic for exploration is the choice of the number of word and/or document clusters to be used for the classification task. This and various other parameters are usually defined using various heuristics or are tuned manually. An investigation of automatic approaches to tune the parameters is also desirable.
This review reveals that the area under research is vivid and that clustering is applied in many sub-domains of the problem of text classification. The clustering field can, and indeed must play an important role in enabling effective classification. It is important to invent new designs that are able to support new forms of collaboration but it is essential that this should be done only on the basis of a better understanding of what needs to be accomplished. In this paper, an attempt has been made to achieve such an understanding by abstracting patterns of current applications of clustering to aid classification. We believe that text classification aided by clustering is worthy area of focus for information retrieval, machine learning and artificial intelligence research; both for its direct application and for the insight it gives into other similar problems. Research should focus on model selection and theoretic analysis.

4. References

Abney, S., (2002). Bootstraping. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 360-367.

Almeida, M. B., Braga, A. P., Braga, J. P., (2000). SVM-KM: speeding SVMs learning with a pnori cluster Selection and k-means. *IEEE 6th Brazilian Symposium on Neural Networks*, SBRN 2WO.

Asharaf, S., Murty, M. N., Shevade, S. K., (2007). Cluster based training for scaling non-linear Support Vector Machines. *Proceedings of the International Conference on Computing: Theory and Applications* (ICCTA'07).

Awad, M., Khan, L., Bastani, F, Yen, I. L., (2004). An effective support vector machine SVMs performance using hierarchical clustering, *in Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'04)*, pp. 663-667.

Badoiu, M., Har-Peled, S., Indyk. P., (2002). Approximate clustering via core-sets, *in Proceedings of the 34th Symposium on Theory of Computing*.

Baker L. D., McCallum A. K., (1998). Distributional clustering of words for text classification, *Proceedings of SIGIR'98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, AU. ACM Press, New York, US.

Bekkerman R., El-Yaniv R., Tishby N., Winter Y., (2001). On Feature Distributional Clustering for Text Categorization. *Proceedings of SIGIR'01, 24th ACM International*

Conference on Research and Development in Information Retrieval, pages 146–153, New Orleans, US, ACM Press, New York, US.

Bekkerman R., El-Yaniv R., Tishby, N., Winter Y., (2003). Distributional Word Clusters vs. Words for Text Categorization, *Journal of Machine Learning Research*, 3, 1183-1208.

Blum, A., Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *COLT: Proceedings of the Workshop on Computational Learning Theory*.

Boley, D., Cao, D., (2004). Training support vector machine using adaptive clustering. *In Proceeding of 2004 SIAM International Conference on Data Mining*.

Burges, C. J. C., (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* 2:121 – 167.

Cervantes, J., Li, X., Yu, W., (2006). Support vector machine classification based on fuzzy clustering for large data sets, *in MICAI 2006 Advances in Artificial Intelligence, Lecture Notes in Computer Science (LNCS)*, vol. 4293, Springer, Berlin, pp. 572-582.

Cervantes, J., Li, X., Yu, W., Li, K., (2008). Support vector machine classification for large data sets via minimum enclosing ball clustering. *Neurocomputing*, Vol. 71, Issue 4-6, pp. 611-619.

Chapelle, O., Weston, J., Scholkopf, B., (2002). Cluster kernels for semi-supervised learning. In *NIPS*, volume 15.

Dai, W., Xue G.R., Yang, Q., Yu, Y., (2007). Co-clustering based classificaiton for out-of-domain documents. *In Proceednгs of KDD 2007*.

Dhillon I., Mallela S., Kumar R., (2002). Enhanced word clustering for hierarchical text classification, *in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, Edmonton, Alberta, Canada, pp. 191-200.

Dhillon I., Mallela S., Kumar R., (2003a). A Divisive Information-Theoretic Feature Clustering Algorithm for Text Classification, *Journal of Machine Learning Research 3*, 1265-1287.

Dhillon I., Mallela S., Modha, S., (2003b). Information theoretic co-clustering. *In Proceedings of the ACM SIGKDD Conference*.

Fang, Y., C., Parthasarathy, S., Schwartz, F., (2002). Using Clustering to Boost Text Classification. *In Proceedings of the IEEE ICDM Workshop on Text Mining*.

Felici, G., Truemper, K., (2002). A minsat approach for learning in logic domains. *Informs. J. Computing*, Vol. 14, No. 1.

Friedman, M., Kandel, A. (1999). Introduction to pattern recognition, *Chapter Distance Functions*, 70–73. London, UK: Imperial College Press.

Fung, G. and Mangasarian, O.L., (2001). Semi-supervised support vector machines for unlabeled data classification. Optim. Methods Software. v15 i1. 29-44.

Goldman, S., Zhou, Y. (2000). Enhancing supervised learning with unlabeled data. *Proc. 17th International Conf. on Machine Learning*, pp. 327–334, Morgan Kaufmann, San Francisco, CA.

He, J., Zhong, W., Harrison, R., Tai, P. C., Pan, Y., (2006). Clustering support vector machines and its application to local protein tertiary structure prediction. *ICCS 2006*, part II,LNCS 3992, pp. 710-717.

Jain, A. K., Dubes, R.C., (1988). Algorithms for Clustering Data. *PrenticeHall*, Englewood Clis, New Jersey.

Koller, D., Sahami, M., (1996). Toward optimal feature selection. *In proceedings of te 13th International Conference on Machine Learning (ICML-96).*

Kumar, P., Mitchell, J.S.B., Yildirim, A., (2003). Approximate minimum enclosing balls in high dimensions using core-sets, *ACM J. Exp. Algorithmics*, 8.

Kyriakopoulou, A., (2007). Using Clustering and Co-training to Boost Classification Performance. *In proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence*, volume 2, pp. 325-330.

Kyriakopoulou, A., Kalamboukis, T., (2006) Text classification using clustering. *In Proceedings of the ECML-PKDD Discovery Challenge Workshop*.

Kyriakopoulou, A., Kalamboukis, T., (2007). Using clustering to enhance text classification. *In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 805 – 806

Lavelli, A., Sebastiani, F., Zanoli, R., (2004). Distributional term representations: an experimental comparison. *CIKM 2004*: 615-624.

Lewis, D. D., (1992). An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*

Li, B., Chi, M., Fan, J., Xue, X., (2007). Support Cluster Machine. *In Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR.

Li, M., Cheng, Y., Zhao, H., (2004). Unlabeled data classification via support vector machine and k-means clustering. *In Proceedings of the International Conference on Computer Graphics, Imaging and Visualization*, CGIV'04, pp. 183-186.

Lin, J., (1991). Divergence measures based on shannon entropy. *IEEE Transactions on Information Theory*, 37 (14):145–51.

Luo, F., Khan, L., Bastani, F., Yen, I.L., Zhou, J., A Dynamical Growing Self-Organizing Tree (DGSOT) for Hierarchical Clustering Gene Expression Profiles, *The Bioinformatics Journal*, Oxford University Press, UK.

Mubaid, H.A., Umair, S.A., (2006). A new text categorization technique using distributional clustering and learning logic. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 9.

Ng, A. Y., Jordan, M. I., Weiss, Y, (2002). On spectral clustering: Analysis and an algorithm. In T. Dietterich, S. Becker and Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems (NIPS)*, 14.

Pereira F., Tishby N., Lee L., (1993). Distributional clustering of English words, *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, p. 183-190.

Platt, J., (1998). Fast training of support vector machine using sequential minimal optimization. *In Advances in Kernel Methods: Support Vector Machine*, MIT Press, Cambridge, MA.

Raskutti, B., Ferrá, H., Kowalczyk, A., (2002a). Using Unlabelled Data for Text Classification through Addition of Cluster Parameters. *Proceedings of the Nineteenth International Conference on Machine Learning*, Pages: 514 – 521.

Raskutti, B., Ferrá, H., Kowalczyk, A., (2002b). Combining clustering and co-training to enhance text classification using unlabelled data, *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*.

Rissanen, J, (1989) Stochastic Complexity in Statistical Enquiry. *World Scientific.*

Salton, G. and McGill, M. J. (1983). Introduction to Modern Information Retrieval. *McGraw-Hill*, New York, NY.

Seeger, M. (2000). Input-dependent regularization of conditional density models. *Technical Report*. http://www.kyb.tuebingen.mpg.de/bs/people/seeger/

Slonim, N., Tishby, N., (1999). Agglomerative Information Bottleneck. *Advances in Neural Information Processing Systems*, p. 617-623.

Slonim, N., Tishby, N., (2000). Document Clustering using Word Clusters via the Information Bottleneck Method. *In Proceedings of the ACM SIGIR*.

Slonim, N., Tishby, N., (2001). The power of word clustering for text classification. *In Proceedings of the European Colloquium on IR Research, ECIR*.

Slonim, N., Friedman, N., Tishby, N., (2001). Agglomerative Multivariate Information Bottleneck. *Neural Information Processing Systems (NIPS 01)*.

Slonim, N., Friedman, N., Tishby, N., (2002). Unsupervised document classification using sequential information maximization. In Proc. of SIGIR, pages 129-136.

Spielman, D. Teng, S. (1996). Spectral partitioning works: planar graphs and finite element meshes. *In 37th Annual Symposium on Foundations of Computer Science*. Burlington, VT, pp. 96–105. Los Alamitos, CA: IEEE Comput. Soc. Press.

Sun, S., Tseng, C. L., Chen, Y. H., Chuang, S. C., Fu, H. C. (2004). Cluster-based support vector machines in text-independent speaker identification. *In Proceedings of the International Joint Conference on Neural Network*.

Takamura, H., (2003). Clustering approaches to text categorization, *Doctor's thesis*, NAIST-IS-DT0061014.

Takamura, H., Matsumoto, Y., (2002). Two-dimensional Clustering for Text Categorization. *In Proceedings of Sixth Conference on Natural Language Learning (CoNLL-2002)*, Taipei, Taiwan, pages 29-35, August-September.

Tishby, N. Z., Pereira, F., Bialek, W., (1999). The Information Bottleneck Method. *In Proceedings of the 37th Allerton Conference on Communication, Control and Computing*.

Verbeek, J., (2000a). An information theoretic approach to finding word groups for text classification. *Master 's thesis, Institute for Logic, Language and Computation (ILLC-MoL-2000-03)*, Amsterdam, The Netherlands.

Verbeek, J. (2000b). Supervised Feature Extraction for Text Categorization. *Benelearn: Annual Machine Learning Conference of Belgium and the Netherlands*.

Wang, J., Wu, X., Zhang, C. (2005). Support vector machines based on K-means clustering for real-time business intelligence systems. *Int. J. Business Intelligence and Data Mining*, Vol. 1, No. 1, pp.54–64.

Yaniv R. E., Souroujon O., (2001). Iterative Double Clustering for Unsupervised and Semi-supervised Learning. *In proceedings of the 12th European Conference on Machine Learning, ECML*.

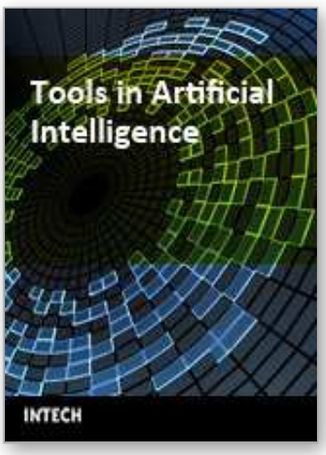
Yu, H., Yang, J., Han, J., (2003). Classifying large data sets using SVMs with hierarchical clusters, *in Proceedings of the 9th ACM SIGKDD 2003*, Washington, DC, USA.

Yuan, J., Li, J., & Zhang, B. (2006). Learning concepts from large scale imbalanced data sets using support cluster machines. *Proceedings of the ACM International Conference on Multimedia*, (pp. 441–450).

Zhang, T., Ramakrishnan, R., Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. *Proc. of the 1996 ACM SIGMOD Int'l Conf. on Management of Data*, pp. 103–114.

Zeng, H., J., Wang, X., H., Chen, Z., Lu, H., Ma, W., Y., (2003). CBC: Clustering Based Text Classification Requiring Minimal Labeled Data. *In Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03)*.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., Scholkopf, B.. (2003). Learning with local and global consistency. *In 18th Annual Conf. on Neural Information Processing Systems*.



Tools in Artificial Intelligence

Edited by Paula Fritzsche

ISBN 978-953-7619-03-9
Hard cover, 488 pages
Publisher InTech
Published online 01, August, 2008
Published in print edition August, 2008

This book offers in 27 chapters a collection of all the technical aspects of specifying, developing, and evaluating the theoretical underpinnings and applied mechanisms of AI tools. Topics covered include neural networks, fuzzy controls, decision trees, rule-based systems, data mining, genetic algorithm and agent systems, among many others. The goal of this book is to show some potential applications and give a partial picture of the current state-of-the-art of AI. Also, it is useful to inspire some future research ideas by identifying potential research directions. It is dedicated to students, researchers and practitioners in this area or in related fields.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen