Hierarchically SVM classification based on support vector clustering method and its application to document categorization

Pei-Yi Hao a,*, Jung-Hsien Chiang b, Yi-Kun Tu b

a Department of Information Management, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan, ROC
b Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan

Abstract

Automatic categorization of documents into pre-defined topic hierarchies or taxonomies is a crucial step in knowledge and content management. Standard machine learning techniques like support vector machines and related large margin methods have been successfully applied for this task, albeit the fact is that they ignore the inter-class relationships. Unfortunately, in the context of document categorization, we face a large number of classes and a huge number of relevant features needed to distinguish between them. The computational cost of training a classifier for a problem of this size is prohibitive. It has also been observed that obtaining a classifier that discriminates between two groups of classes is much easier than distinguishing simultaneously among all classes. This has prompted substantial research in using hierarchical classifiers to address single multi-class problems. In this paper, we propose a novel hierarchical classification method that generalizes support vector machine learning that is based on the results of support vector clustering method, and are structured in a way that mirrors the class hierarchy. Compared to previous non-hierarchical SVM classifier and famous documents categorization systems, the proposed hierarchical SVM classification has a better improvement in classification accuracy in the standard Reuters corpus.

Keywords: Information retrieval; Document categorization; Hierarchical classification; Support vector machines; Support vector clustering method; Machine learning

1. Introduction

Due to the rapid growth in textual data, automatic methods for organizing the data are needed. Automatic document categorization is one of these methods. It automatically assigns the documents to a set of pre-defined classes based on its textual content. Document categorization is a crucial and well-proven instrument for organizing large volumes of textual information. In most cases, the use of statistical or machine learning techniques has been proven to be successful as per this context, since it is typically more feasible to induce categorization rules based on example documents, than to elicit such rules from domain experts. The wide range of methods applied to this problem include multivariate regression models (Schütze, Hull, & Pedersen, 1995), probabilistic Bayesian models (Koller & Sahami, 1997; Lewis & Ringuette, 1994), decision trees (Lewis & Ringuette, 1994; Weiss et al., 1999), neural networks (Schütze et al., 1995; Weigend, Wiener, & Pedersen, 1999), symbolic rule learning (Apte, Damerau, & Weiss, 1994), nearest neighbor classifiers (Xie & Beni, 1991), and – more recently – boosting (Schapire, Singer, & Singhal, 1998) and support vector machines (SVMs) (Joachims, 1998). Extensive experimental comparisons (e.g. Joachims, 1998; Sebastiani, 2002; Yang & Liu, 1999) have evidenced that among the methods available today, SVMs are highly competitive in their classification accuracy and can therefore be considered as the state-of-the art in document categorization.
A potential drawback of all of the above mentioned classification methods is that they treat the category structure as ‘flat’ where the pre-defined categories are treated in isolation and there is no structure defining the relationships among them (D’Alessio, Murray, Schiaffino, & Kershbaum, 2000; Yang, 1999). Unfortunately, in the context of document categorization, we came across a large number of classes and a huge number of relevant features that are needed to be distinguished. The computational cost of training a classifier for a problem of this size is prohibitive. It has also been observed that obtaining a classifier that discriminates between two groups of classes is much easier than that distinguishes simultaneously among all classes. This has prompted substantial research in using hierarchical classifiers to address single multi-class problems.

The idea of hierarchical classification is that solving a set of small problems with fewer classes can be achieved faster and more effectively than solving one large-scale classification problem distinguishing a large amount of classes. Hierarchical classification allows us to address a large classification problem using a divide-and-conquer approach. It decomposes the classification task into a set of simpler problems, one at each node in the classification tree. As we show, each of these smaller problems can be solved accurately and efficiently. At the root level in the category hierarchy, a document can be first classified into one or more sub-categories using some flat classification methods. The classification can be repeated on the document in each of the sub-categories until the document reaches some leaf categories or cannot be further classified into any sub-categories. To do so, several approaches have been introduced (Cai & Hofmann, 2004; D’Alessio et al., 2000; Dumais & Chen, 2000; Koller & Sahami, 1997; Larkey, 1998; McCallum, Rosenfeld, Mitchell, & Ng, 1998; Sun & Lim, 2001; Vaithyanathan, Mao, & Dom, 2000). Most of them achieved a big performance improvement and some gained classification accuracy.

The hierarchical classification offers a lot of flexibility in designing the classifier system. For instance, one can replace the classifier at the internal nodes of the generated hierarchical structure with stronger classifiers such as SVM (Cortes & Vapnik, 1995; Vapnik, 1995). Moreover, different feature selection methods can be used at each node that is specific to the domain of the input data. Each sub-problem is smaller than the original problem, and it is sometimes possible to use a much smaller set of features (Dumais & Chen, 2000; Koller & Sahami, 1997).

In this paper, we propose a novel hierarchical classification method that generalizes Support Vector Machine learning and is based on the results of support vector clustering method that are structured in a way that mirrors the class hierarchy. The grouping of individual classes into the meta-class is determined by the class distributions described by the support vector clustering method. The rest of this paper is structured as follows. In Sections 2 and 3 we give a brief review of the hierarchical classification and support vector clustering method, respectively. In Section 4, we illustrate the proposed hierarchical SVM classification where the hierarchical structure is constructed by the support vector clustering method. We provide our experimental methodology with a variety of results supporting our approach in Section 5, and the concluding remarks are given in Section 6.

### 2. The hierarchical classification approach

In the hierarchical classification approach, one or more classifiers are constructed at each level of the category tree and each classifier works as a flat classifier at that level. A document will first be classified by the classifier at the root level into one or more lower level categories. It will then be further classified by the classifier(s) of the lower level category(ies) until it reaches a final category, which could be a leaf category or an internal category. Given the above general definition of a hierarchy, two basic cases can be distinguished: (i) A tree structure, where each class (except the root class) has exactly one parent classes, and (ii) a directed acyclic graph structure where a class can have more than one parent classes.

#### 2.1. (Virtual) category tree

In the virtual category tree structure, categories are organized as a tree. Each category can belong to at most one parent category and the documents can only be assigned to the leaf categories (Dumais & Chen, 2000). The category tree structure is an extension of the virtual category tree that allows documents to be assigned into both internal and leaf categories (Wang, Zhou, & He, 2001).

A famous example of the virtual category tree classification is the Binary Hierarchical Classifier (BHC) architecture Kumar, Ghosh, and Crawford, 2002 that addresses multi-class classification problems using a set of binary classifiers. The Binary Hierarchical Classifier recursively decomposes a multi-class (N-classes) problem into N-1 two meta-class problems, resulting in N-1 classifiers arranged as a binary tree, as shown in Fig. 1. The given set of classes is first partitioned into two disjoint meta-classes and each meta-class
thus obtained is partitioned recursively until it contains only one of the original classes. The tree thus has a number of leaf nodes equal to the number of classes in the output space. As we know, a binary tree of $n$ nodes has \( \frac{1}{n+1} \binom{2n}{n} \) different structures. In BHC, the structure of the binary tree is constructed through a deterministic annealing process that encourages similar classes to remain in the same partition. As a direct consequence of the BHC algorithm, classes that are similar to each other in the input feature space are thus lumped into the same meta-class higher up in the tree. Interested readers are referred to Kumar et al. (2002) for details of the algorithm.

2.2. (Virtual) directed acyclic category graph

In the virtual directed acyclic category graph structure, categories are organized as a Directed Acyclic Graph (DAG) where a class can have more than one parent classes. Similar to the virtual category tree, documents can only be assigned to leaf categories. The directed acyclic category graph structure is an extension of the virtual directed acyclic category graph structure. It is perhaps the most commonly used structure in the popular web directory services such as Open Directory Project and Yahoo. Documents can be assigned to both internal and leaf categories in the directed acyclic category graph structure.

Recently, a novel multi-class SVM approach, called DAGSVM, has been proposed (Platt, Cristianini, & Shawe-Taylor, 2000). Assuming the number of classes as $n$, its training phase is by solving \( n(n - 1)/2 \) binary SVMs. In the testing phase, it uses a rooted binary directed acyclic graph which has \( n(n - 1)/2 \) internal nodes and \( n \) leaves, as shown in Fig. 2. Each node is a binary SVM of \( i \)th and \( j \)th classes. Given a test sample $x$, starting at the root node, the binary decision function is evaluated. Then it moves to either left or right depending on the output value. Therefore, we go through a path before reaching a left node, which indicates the predicted class. Although the hierarchical structure of DAGSVM is fixed, the assignment of the binary SVM corresponding to each internal node is not unique. Fig. 3 shows different assignments of the Binary SVM to each internal node. It is easy to see that when the number of classes is $n$, there are $n(n - 2)!$ different assignments in the hierarchical structure. To our knowledge, not many studies address the issue of how to obtain a better assignment when the number of classes is huge.

3. Support vector machines for clustering method

Support vector (SV) clustering has been recently derived from the single-class support vector machine (Chiang & Hao, 2003; Schölkopf, Platt, Shawe-Taylor, Smola, & Williamson, 2001; Tax & Duin, 1999) for estimating the underlying probability distribution. Ben-Hur et al. generalize support vectors as the boundary of clusters (Ben-Hur, Horn, Siegelmann, & Vapnik, 2000; Ben-Hur, Horn, Siegelmann, & Vapnik, 2001). We now illustrate the support vector clustering method as shown in Fig. 4.

To begin, let $\Phi$ denotes a nonlinear transformation, which maps the original input space onto a high-dimensional feature space. Clustering may be viewed as finding
References


