A Hierarchical Classification Model for Document Categorization

Jian-Wu Xu†, Vartika Singh‡, Venu Govindaraju‡ and Depankar Neogi†
†*Copanion Inc., Andover, MA 01810, USA*
{*jxu,dneogi*}*@copanion.com*
‡*Center for Unified Biometrics and Sensors, University at Buffalo, USA*
{*vsingh2,govind*}*@buffalo.edu*

Abstract

We propose a novel hierarchical classification method for documents categorization in this paper. The approach consists of multiple levels of classification for different hierarchies. Regularized Least Square (RLS) binary classifiers are applied in the middle levels of the hierarchy to classify documents into smaller set of categories and K-nearest-neighbor (KNN) multi-class classifiers are used at the bottom to classify documents into final classes. Experiments on large-scale real world tax documents show that the proposed hierarchical approach outperforms traditional flat classification method.

1. Introduction

With the exponential proliferation of text documents, it is increasingly necessary to find relevant information quickly. However, not all information is in clean digital form. Even in this digital age, a lot of the information is processed from paper documents. A large amount of existing paper documents are transformed into digital document images via scanners or cameras. Efficient storage, retrieval, and management of these document image archives are extremely important in many office automation and digital library applications. As a result, techniques for automatic document image analysis and classification are highly demanded.

A good example of high volume document processing is the US individual income tax preparation process. US IRS (Internal Revenue Services) received more than 135 million individual income tax returns in 2008. More than 80 million of these tax returns are prepared by professionals such as Certified Public Accountants (CPAs). Manual labeling of documents is not only a time consuming process, but also very error-prone given the large variation tax document classes and the size of individual tax returns. Therefore, an automatic tax document classification process is highly desirable to assist the preparers.

Many machine learning algorithms have been used in traditional document classification such as Support Vector Machines (SVMs), K-nearest-neighbor, latent conditional independence model, decision trees, neural networks and others [1]. These classifiers operate on one of image features, textual features and document layout structure features or some combination depending on the specific problems. Most classifications employ a flat (non-hierarchical) approach. All document classes are treated at the same level without considering the hierarchical structure of different document categories. A single classifier is trained to categorize documents into single or multiple classes.

However, most of real world text documents possess intrinsic hierarchical structures (e.g. US individual income tax related documents). As illustrated in Fig. 1, tax documents can be organized in a multi-layer hierarchical tree. Tax documents consist of organizers (forms containing client tax and personal information), IRS (forms related with individual income tax issued by Internal Revenue Service), tax return documents and other broad categories. IRS forms are composed of tens of individual documents such as W-2 (wage and tax statement), 1099-INT (interest income statement) and many others. W-2 itself has more than 500 various forms with distinct layout structures such as ADP-W2 and Bain-W2 which are created by different companies.

Recently there have been increasing interests in applying hierarchical classification for documents categorization to take the advantage of intrinsic hierarchical information of categories. Sako *et al.* used hierarchical template matching to classify forms [5]. A hierarchical multiple Bayesian classifier is proposed to decompose the classification task into a set of much sim-

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.187

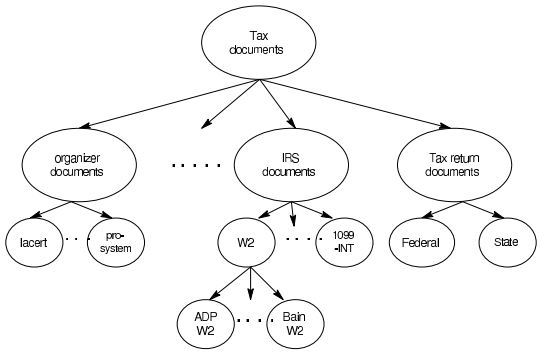


Figure 1. Hierarchy of individual income tax return related documents categorization

pler problems by Koller in [4]. Xue *et al.* presented a deep hierarchical classification framework consisting of search and classification stages for large-scale text hierarchies [8]. By exploiting the hierarchical structure, one is able to decompose the overall classification problem into a set of smaller ones corresponding to hierarchical partitions in the tree. Experiments using different hierarchical classification schemes on different data sets have been shown to outperform the traditional flat classification methods. However, most of the hierarchical approaches only apply one single classifier, be it naive Bayes classifier [4] [8], SVM [2] or others.

In this paper, we describe a hierarchical classification model consisting of multi-class and binary classifiers for text documents. We apply binary Regularized Least Square (RLS) classifiers to the hierarchical branches in the tree. In the same level of the hierarchical tree, multiple *one-against-rest* binary RLS classifiers are trained for each sub-categories. We rank the sub-categories from classifiers' outputs and pick the highest one as the category candidate so that a large-scale hierarchy is pruned into a single and much smaller set. This approach continues until it reaches the leaves of the hierarchical tree. Then K-nearest-neighbor (KNN) classifiers, as multi-class classifiers, are used at the bottom of the hierarchical tree to classify documents into final required classes. This flexible hierarchical categorization framework not only speeds up the classification but also improves the performance by tackling the unbalanced data issue because it balances positive and negative samples by merging the classes in the lower level into a higher category.

In order to evaluate the proposed method, we have collected one of the largest and most comprehensive sets of individual income tax related documents so far in the literature. We test the hierarchical classifiers on this data set and compare with a non-hierarchical classification. The results show a substantial improvement of classification performance of our method over the non-hierarchical classifier.

The rest of the paper is organized as follows. We present the proposed hierarchical classification method in Sec. 2. We compare the proposed method against a traditional flat classification approach on a large-scale set of tax documents in Sec. 3 . We conclude our work in Sec. 4.

2. Hierarchical Classification

We present the overall hierarchical classification framework and introduce the individual multi-class and binary classifiers in this section.

2.1. Overall Hierarchical Framework

The proposed hierarchical classification consists of RLS classifiers in the middle level of hierarchical tree and KNN classifiers for the bottom classes. In each level above the bottom, we classify the test document by multiple *one-against-rest* binary RLS classifiers. Each RLS classifier is constructed for one of the several categories. The l-th RLS classifier is trained on the whole training data set to classify the members of class l against the rest. Each level classification produces a single candidate category for the next level processing. Based on the output from the upper level, another set of RLS classifiers are invoked to further classify the document into smaller set of categories. When the process reaches the bottom of the hierarchical tree, the corresponding category KNN classifier is invoked to classify the document into the final required class. In the case there are multiple candidate categories from the middle levels, we pick the one with highest confidence value. If none of the RLS classifiers have positive labels, the test document is rejected. As shown in the experiments, the latter two cases are quite rare.

Compared to the flat classification approach, the proposed hierarchical classifiers tackles the unbalanced data issue by merging the lower level classes into upper level larger categories in the hierarchical tree. This eventually improves the classification performance as demonstrated in the experiments. It also offers flexibility in classifying test documents into multi-resolution categories depending on domain specific requirements.

2.2. Regularized Least Square

Regularized Least Square has been proven to be effective for classification problems in the text mining do-

main. Given the training data set $(\mathbf{x}_i, y_i)$, $i = 1, \cdots, N$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the i-th feature vector extracted from i-th training document and $y_i \in \{+1, -1\}$ is the corresponding label, denote $\mathbf{X} \in \mathbb{R}^{N \times d}$ the feature matrix whose i-th row contains the features for the i-th data point, and $\mathbf{y}$ the label vector of N labels. RLS constructs a hyperplane-based function $\mathbf{w}^\top \mathbf{x}$ to approximate the output $\mathbf{y}$ by minimizing the following loss function [3],

$$L(\mathbf{w}) = ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2 + \lambda ||\mathbf{w}||^2, \quad (1)$$

where $||\cdot||$ means the L_2-norm of a vector and $\lambda > 0$ is the *regularization parameter*. λ balances off the square loss of the classification error and the regularization term. Regularization is important because the covariance matrix of the feature matrix $\mathbf{X}^\top \mathbf{X}$ might be singular due to the sparsity of the OCR text features. By zeroing the derivative of L with respect to $\mathbf{w}$, the optimal solution of RLS is given by

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}.$$

In order to train RLS classifiers at the top level of the overall hierarchies, the training samples are merged from the lower levels and relabeled. Members of the l-th category are labeled to $+1$, and members of the other classes are labeled to -1. After the training phase, the inner product between a given test document $\mathbf{x}_t$ and the trained weight vector, $\mathbf{w}^\top \mathbf{x}_t$, is compared to a threshold to determine either positive or negative class label.

2.3. Probabilistic Perspective towards RLS

The regularization parameter λ is often tuned via a cross-validation procedure. Here we present a probabilistic interpretation for RLS and derive a principled way of updating the parameter.

Suppose the output y_i conditioned on $\mathbf{w}$ follows a Gaussian distribution with mean $\mathbf{w}^\top \mathbf{x}_i$ and variance σ^2, i.e., $y_i|\mathbf{w} \sim \mathcal{N}(\mathbf{w}^\top \mathbf{x}_i, \sigma^2)$, and the weight vector $\mathbf{w}$ is Gaussian distributed, $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then the negative log-posterior density of $\mathbf{w}$ is equivalent to the RLS cost function $L(\mathbf{w})$ defined in (1) with $\lambda = \sigma^2$ [3].

The probabilistic interpretation of RLS provides an alternative to optimize the regularization parameter $\lambda = \sigma^2$ by maximizing the marginal likelihood of the data,

$$\log P(\mathbf{y}|\sigma^2) = -\frac{N}{2} \log 2\pi - \frac{1}{2} \log |\mathbf{X}\mathbf{X}^\top + \sigma^2 \mathbf{I}| - \frac{1}{2} \mathbf{y}^\top (\mathbf{X}\mathbf{X}^\top + \sigma^2 \mathbf{I})^{-1} \mathbf{y}.$$

Alternatively, one can also derive an EM (Expectation-Maximization) algorithm. In this approach, we estimate the posterior distribution of $\mathbf{w}$ in the E-step, which is a Gaussian $\mathcal{N}(\boldsymbol{\mu}_\mathbf{w}, \mathbf{C}_\mathbf{w})$, with

$$\boldsymbol{\mu}_\mathbf{w} = (\mathbf{X}^\top \mathbf{X} + \sigma^2 \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y},$$
$$\mathbf{C}_\mathbf{w} = \sigma^2 (\mathbf{X}^\top \mathbf{X} + \sigma^2 \mathbf{I})^{-1}.$$

Then in the M-step we maximize the "complete" log-likelihood with respect to σ^2, assuming the posterior of $\mathbf{w}$ as given in the E-step. This leads to the following update rule for λ:

$$\lambda = \frac{1}{N} \left[||\mathbf{y} - \mathbf{X}\mathbf{w}||^2 + \mathrm{tr}(\mathbf{X}\mathbf{C}_\mathbf{w}\mathbf{X}^\top) \right].$$

The final algorithm iterates the E-step and M-step until convergence.

2.4. KNN

KNN classifiers are used at level 2 classification. It is an effective multi-class classifier. KNN is an instance-based learning algorithm which stores all the training data during training phase, finds the most similar training instances for a given test document and assigns the label. In this paper, we use cosine distance $\cos(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^\top \mathbf{x}_j}{||\mathbf{x}_i|| ||\mathbf{x}_j||}$ as similarity measure and only consider the closest neighbor.

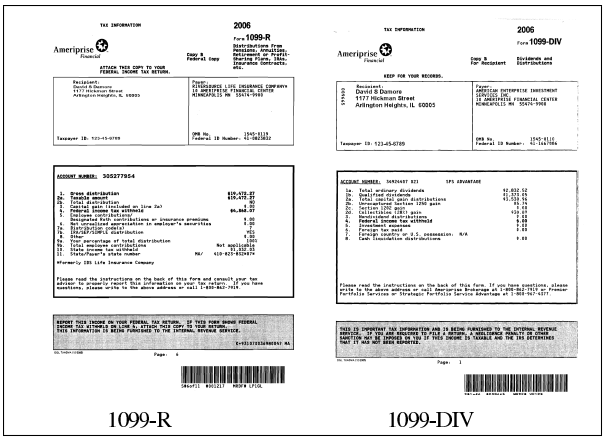
KNN is suitable for level 2 classification since there are around 50 classes inside each of the four categories and KNN is practical for multi-class classification though the computation complexity is higher than RLS. While for the first level, it is affordable to construct four binary RLS classifiers because only four merged classes exist at level 1. Therefore, the proposed hierarchical classification methodology novelly combines RLS and KNN classifiers for different classification stages. RLS classifiers at level 1 are used to narrow down the searching category quickly and KNN classifiers further refines the final output label.

3. Experiments

In this section, we apply the proposed hierarchical classification method to individual income tax documents categorization problem. We describe the data set collection, feature extraction and experiments. The results on hierarchical classification of tax documents show significant improvement over flat classification with minimum distance classifiers.

3.1. Tax Documents Classification

Recently, Sarkar proposed to use 5 dimensional thresholded Viola-Jonese rectangular features with a



1099-R 1099-DIV

Figure 2. forms with similar image features but of different classes

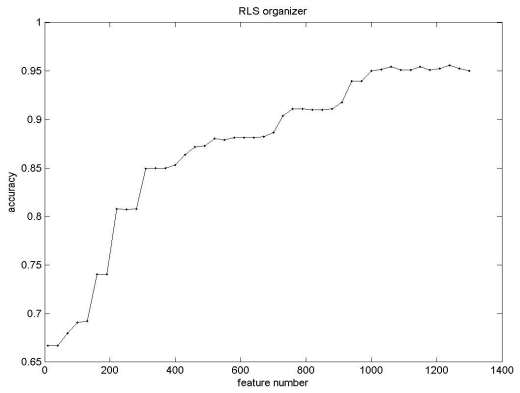


Figure 3. Accuracy versus feature numbers for RLS classifier for Organizer

latent conditional independence model to classify tax documents [6]. Shin *et al.* presented a decision tree classifier based on image features such as percentages of text and non-text content regions, column structures to classify documents [7]. Both papers applied the proposed methods on NIST tax forms data sets [1]. However NIST database only contains 20 classes of IRS tax documents, which account for a very small fraction of overall tax document classes as shown in Fig. 1. Moreover, the categorization of tax documents is based on textual content of forms, not image features. There are many forms sharing similar image features but belonging to different classes. For example, two forms in Fig. 2 have similar image features, but the left one is 1099-R (Distributions from Pensions, Annuities, Retirement Plans, IRAs, or Insurance Contracts) and the right one is 1099-DIV (Dividends and Distributions) which can only be determined by textual information.

3.2. Data Collection and Pre-processing

In the experiment,we have collected the most comprehensive tax documents so far in the literature. A large-scale individual income tax related documents are collected through various accounting firms across US for the 2008 tax season. There are 33 final classes required by CPAs. All these pages have been manually labeled by professionals. Our dataset is much more complete than the NIST tax forms dataset which contains 5,590 pages and only accounts for a small portion of IRS documents. This makes our experiments more realistic compared to previous work on NIST dataset.

[1] http://www.nist.gov/srd/nistsd2.htm

3.3. Feature Extraction

We use textual content from OCR as the features for classification partly because document categorization is mainly based on texts and partly due to the ambiguities of image features as illustrated in Fig. 2. Even though OCR might produce noisy features, we posit that the effect of OCR errors only increases the dimensionality of feature vectors while not deteriorating the classification performance because OCR will only produce multiple words for a given string in the image if the OCR error is consistent.

Unigram ("bag-of-words") is used as document feature. We apply χ^2-statistics for each word to select sufficient number of features based on the rank of χ^2 value [9]. Fig. 3 plots the variation of accuracy with respect to different feature numbers for RLS classifier for Organizer category. As more words are selected, the accuracy also increases. However, it saturates after sufficient number of features. We select that number as the final feature dimension for that classifier. Each document is represented as a vector in the feature space of distinct terms based on the vector-space model. We employ *tf-idf* (term frequency-inverse document frequency) as feature value which is calculated as

$$tfidf_{ij} = \log(1 + tf_j) \log \frac{N}{df_j}$$

for j-th term in i-th document where tf_j is the frequency of j-th term in i-th document, N is the total number of training documents, and df_j corresponds to the number of documents containing j-th term. Each vector is normalized into unit Euclidean norm.

We first clean up noise and de-skew for bad-quality pages. Then we apply OCR to obtain texts from doc-

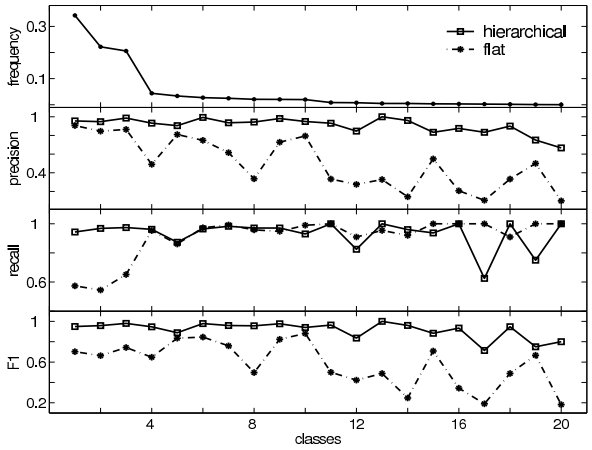


Figure 4. Data statistics and experiment results

uments. Most of the numerals are deleted since they do not provide useful information for classification purpose except for those associated with class labels. We also remove stop words and extreme low frequency terms. After all these pre-processings, we randomly select 70% of the overall documents for training and remaining 30% for testing.

3.4. Results

We use precision, recall and F1 measures to compare the proposed hierarchical classification approach with a flat minimum distance classifier based on the same textual feature. The precision, recall and F1 are standard measures in text classification. The minimum distance classifier regards all classes at the same level and assigns the label for the test document according to the Euclidean distance between the test sample and centroids of training data of different classes. We present classification performance on 20 classes with more than 20 pages per class. In our hierarchical classification, the outputs from top level hierarchy RLS classifiers consists of 98.1% single positive, 1.1% multiple positives and 0.8% all negatives.

In the top plot of Fig. 4, we show the data frequency distribution for each class. It is clear that most of the classes account for less than 5% of the overall documents which corroborates the unbalance issue of data. The rest of three plots present precision, recall and F1 measures for hierarchical and flat classification methods. Hierarchical classification demonstrates significantly better precision and F1 than flat classification. The precision and F1 measures for flat minimum distance classifier drop steadily as the data frequency decreases. However our proposed method is able to maintain more than 90% precision and F1 measures for almost all classes. This confirms our previous claim that hierarchical classification can tackle data unbalance issue. Hierarchical classification exhibits better recall for high frequent data classes and lower recall for some low frequent ones as recall measure is related with original data statistics.

4. Conclusions

We have presented a hierarchical classification method in this work. It considers the intrinsic hierarchies of documents categorization and exploits the strength of RLS and KNN classifiers to improve classification performance. Experiments on large-scale real world individual income related tax documents have shown significant better classification results of our proposed approach over traditional non-hierarchical one. The framework is flexible and general enough to be applied to other documents with intrinsic hierarchies.

References

[1] N. Chen and D. Blostein. A survey of document image classification: problem statement, classifier architecture and performance evaluation. *Int. J. Doc. Anal. Recognit.*, 10:1–16, May 2007.

[2] S. Dumais and H. Chen. Hierarchical classification of web content. In *Proc. of ACM SIGIR*, pages 256–263, 2000.

[3] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001.

[4] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *Proc. Intl. Conf. on Mach. Learn. (ICML)*, pages 170–178, San Francisco, USA, 1997.

[5] H. Sako, M. Seki, N. Furukawa, H. Ikeda, and A. Imaizumi. Form reading based on form-type identification and form-data recognition. In *Proc. Intl. Conf. on Doc. Anal. Recognit. (ICDAR)*, volume 2, pages 926–930, Edinburgh, Scotland, 2003.

[6] P. Sarkar. Image classification: classifying distributions of visual features. In *Proc. Intl. Conf. on Pattern Recognition (ICPR)*, pages 472–475, Hong Kong, 2006.

[7] C. Shin, D. Doermann, and A. Rosenfeld. Classification of document pages using structure-based features. *Int. J. Doc. Anal. Recognit.*, 3(4):232–247, 2001.

[8] G.-R. Xue, D. Xing, Q. Yang, and Y. Yu. Deep classification in large-scale hierarchies. In *Proc. of ACM SIGIR*, pages 619–626, Singapore, 2008.

[9] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proc. Intl. Conf. on Mach. Learn. (ICML)*, pages 412–420, 1997.