

---

# Learning with Taxonomies: Classifying Documents and Words

---

**Thomas Hofmann**

Department of Computer Science  
Brown University  
th@cs.brown.edu

**Lijuan Cai**

Department of Computer Science  
Brown University  
ljcai@cs.brown.edu

**Massimiliano Ciaramita**

Department of Cognitive and Linguistic Sciences  
Brown University  
massi@brown.edu

## Abstract

Automatically extracting semantic information about word meaning and document topic from text typically involves an extensive number of classes. Such classes may represent predefined word senses, topics or document categories and are often organized in a taxonomy. The latter encodes important information, which should be exploited in learning classifiers from labeled training data. To that extent, this paper presents an extension of multiclass Support Vector Machine learning which can incorporate prior knowledge about class relationships. The latter can be encoded in the form of class attributes, similarities between classes or even a kernel function defined over the set of classes. The paper also discusses how to specify and optimize meaningful loss functions based on the relative position of classes in the taxonomy. We include experimental results for text categorization and for word sense classification.

## 1 Introduction

Many real-world classification tasks are multiclass problems involving large numbers of classes. This is in particular true for application domains like information retrieval and natural language processing, where classes may correspond to document categories or word senses: several thousand or even tens of thousands of classes are not uncommon. For instance, the International Patent Classification (IPC) scheme [8] consists of approximately 69,000 classes (called groups) that are used to categorize patent documents and WordNet 2.0 [3] consists of almost 80,000 word senses (called synsets) defined by lexicographers to classify the meaning of English nouns. Multiclass problems of this scale pose a severe challenge for learning algorithms and classification accuracies obtained by even the best classification methods are often disappointingly poor.

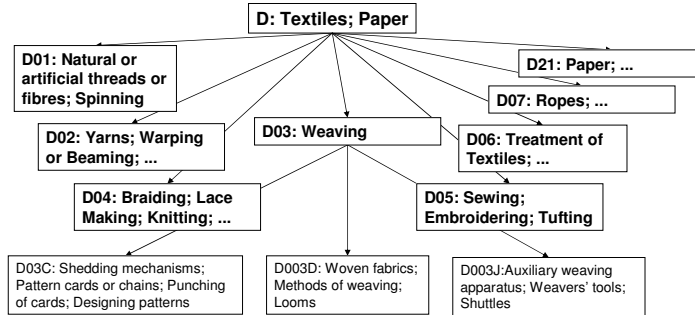


Figure 1: Part of the IPC classification hierarchy rooted at section 'D' which contains a total of 160 main groups. Only classes and subclasses for D03 are shown.

There are at least three directions that can be pursued in order to improve the current state-of-the-art: developing better learning algorithms, deriving better feature representations, and using additional knowledge. In this paper, we focus on the third aspect and, more specifically, investigate how one can make use of existing prior knowledge about the relationship between classes in learning multiclass classifiers. Such knowledge is usually expressed and represented in the form of a *taxonomy* or *ontology*. Taxonomies are essential for organizing classification systems, which need to be used and maintained by human experts. It is thus reasonable to assume that specific taxonomies or ontologies will be available in most cases of practical interest. Indeed, the IPC categories are organized in a four level hierarchy (cf. Figure 1) and WordNet has a sophisticated lattice structure including many levels of super-senses (cf. Figure 2).

The idea we will pursue in this paper is to construct hierarchical discriminant functions as a superposition of simpler functions associated with nodes in a taxonomy and to use a regularization approach that will take advantage of the prior knowledge encoded by the taxonomy. The spirit of this approach is similar to the naive Bayes shrinkage technique presented in [4, 5] which performs taxonomy-specific interpolation to estimate class conditional feature distributions.

The rest of the paper is organized as follows: We will review multiclass SVMs in Section 2 and then generalize this formulation in Section 3 so that prior knowledge about class attributes and/or similarities can be taken into account. Section 4 presents the concrete application of this approach to learning with taxonomies and discusses a further extension in order to minimize specific loss functions derived from a taxonomy. Finally, we present an optimization algorithm in Section 5 and discuss experimental results in Section 6.

## 2 Multiclass Support Vector Machine Learning

We take the multiclass SVM learning approach of [7, 2] as our starting point. Let  $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$  be a set of  $n$  training examples.  $\mathbf{x}_i \in \mathbb{R}^D$  denotes patterns representing items such as documents, word contexts, images, etc. Each label  $y_i$  is an integer from  $\mathcal{Y} = \{1, \dots, Q\}$  where  $Q$  is the number of possible classes or categories (for simplicity classes are identified with the integers from 1 to  $Q$ ). Let us introduce a weight vector  $\mathbf{w}_y$  for every class  $1 \leq y \leq Q$ . We will refer to the stacked weights by

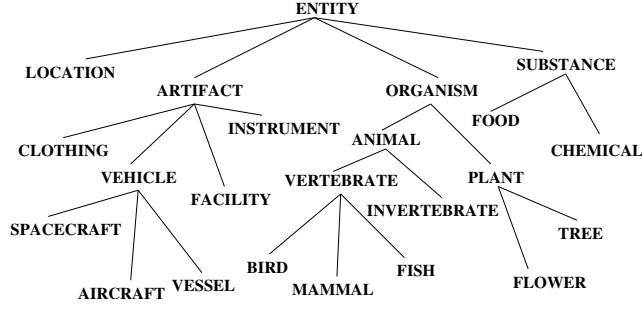


Figure 2: A simplified representation of the portion of the WordNet noun hierarchy below the concept node “entity”.

$\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_Q)'$ . Then we can define a linear discriminant function<sup>1</sup>

$$F(\mathbf{x}, y; \mathbf{w}) \equiv \langle \mathbf{x}, \mathbf{w}_y \rangle \quad (1)$$

and a corresponding classification function  $f$  as

$$f(\mathbf{x}; \mathbf{w}) = \operatorname{argmax}_{y \in \mathcal{Y}} F(\mathbf{x}, y; \mathbf{w}). \quad (2)$$

(2) is also known as the Winner-Take-All (WTA) rule. The multiclass margin of a weight vector with respect to an instance  $(\mathbf{x}_i, y_i)$  can be defined as

$$\gamma_i(\mathbf{w}) \equiv \langle \mathbf{w}_{y_i}, \mathbf{x}_i \rangle - \max_{y \neq y_i} \{ \langle \mathbf{w}_y, \mathbf{x}_i \rangle \}, \quad (3)$$

based on which one can apply the large margin principle to determine the weight vector achieving optimal separation

$$\mathbf{w}^* = \operatorname{argmax}_{\mathbf{w}: \|\mathbf{w}\|=1} \min_{i=1}^n \gamma_i(\mathbf{w}). \quad (4)$$

This can equivalently be written as a norm minimization problem, which can also be augmented by slack variables to yield the following soft-margin multiclass formulation

$$\min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \quad (5a)$$

$$\text{s.t. } \gamma_i(\mathbf{w}) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \quad (5b)$$

Notice that every non-linear constraints in (5b) can be expanded into  $Q - 1$  linear constraints of the form

$$\langle \mathbf{w}_{y_i} - \mathbf{w}_y, \mathbf{x}_i \rangle \geq 1 - \xi_i, \quad \forall y \neq y_i \quad (6)$$

so that (5) indeed corresponds to a convex quadratic program with  $n \cdot Q$  linear constraints.

### 3 Support Vector Machine Learning with Class Attributes

We would like to extend the above multiclass SVM formulation to cases, where classes are not just arbitrary numbers, but can be characterized by attribute vectors

<sup>1</sup>One can also introduce explicit bias terms  $b_y$  for every class, but this would complicate the presentation and leads to further complications in the optimization algorithm. We thus restrict ourselves to this simpler setting.

$\lambda(y) \in \mathfrak{R}^S$ . This should be carried out in a way that recovers the standard multiclass setting as a special case of an orthogonal attribute representation with  $S = Q$  and  $\lambda_s(y) = \delta_{ys}$ , i.e. a case where each class is interpreted as a binary attribute of its own. To that extent, we propose to redefine the discriminant functions  $F$  in (1) as

$$F(\mathbf{x}, y; \mathbf{w}) = \langle \mathbf{w}, \Phi(\mathbf{x}, y) \rangle \quad (7)$$

where  $\Phi(\mathbf{x}, y) = \lambda(y) \otimes \mathbf{x}$ . Here  $\otimes$  denotes a tensor product, i.e.  $\Phi(\mathbf{x}, y) \in \mathfrak{R}^{D \cdot S}$  is a vector containing all products of coefficients from the first and second vector argument. Writing out  $\Phi(\mathbf{x}, y)$  one gets

$$\Phi(\mathbf{x}, y) = \begin{pmatrix} \lambda_1(y) \cdot \mathbf{x} \\ \lambda_2(y) \cdot \mathbf{x} \\ \vdots \\ \lambda_S(y) \cdot \mathbf{x} \end{pmatrix}, \quad (8)$$

and for  $\lambda_s(y) = \delta_{ys}$  this simply reduces to

$$\Phi(\mathbf{x}, y) = \begin{pmatrix} \vdots \\ 0 \\ \mathbf{x} \\ 0 \\ \vdots \end{pmatrix} \leftarrow y\text{-th position.} \quad (9)$$

Notice that in this latter case  $\langle \mathbf{w}, \Phi(\mathbf{x}, y) \rangle = \langle \mathbf{w}_y, \mathbf{x} \rangle$  and (7) indeed reduces to the formulation in (1). In general, it is straightforward to show that one can re-write  $F$  as an additive superposition of linear discriminants as follows

$$F(\mathbf{x}, y; \mathbf{w}) = \sum_{s=1}^S \lambda_s(y) \langle \mathbf{w}_s, \mathbf{x} \rangle, \quad (10)$$

where  $\mathbf{w}_s \in \mathfrak{R}^D$  is a weight vector associated with the  $s$ -th class attribute.

It is of conceptual and computational interest to derive the dual SVM formulation for this more general definition of  $F$ . Standard calculations of forming the Lagrangian function and optimizing over the primal variables lead to the dual quadratic program

$$\alpha^* = \operatorname{argmax}_{\alpha} \left\{ -\frac{1}{2} \sum_{i,j} \sum_{y \neq y_i} \sum_{y' \neq y_j} \alpha_{iy} \alpha_{jy'} \langle \Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, y), \Phi(\mathbf{x}_j, y_j) - \Phi(\mathbf{x}_j, y') \rangle \right. \\ \left. + \sum_i \sum_{y \neq y_i} \alpha_{iy} \right\} \quad (11a)$$

$$\text{s.t. } \alpha_{iy} \geq 0, \quad \sum_{y \neq y_i} \alpha_{iy} \leq C. \quad (11b)$$

Notice that the upper bound is a consequence of the introduction of shared slack variables  $\xi_i$  for every training instance. Moreover, we would like to point out that  $\langle \Phi(\mathbf{x}_i, y), \Phi(\mathbf{x}_j, y') \rangle = \langle \lambda(y), \lambda(y') \rangle \langle \mathbf{x}_i, \mathbf{x}_j \rangle$ , which follows immediately from the definition of  $\Phi$  and thus

$$\langle \Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, y), \Phi(\mathbf{x}_j, y_j) - \Phi(\mathbf{x}_j, y') \rangle \quad (12) \\ = \langle \lambda(y_i) - \lambda(y), \lambda(y_j) - \lambda(y') \rangle \langle \mathbf{x}_i, \mathbf{x}_j \rangle.$$

Herein one can simply replace the inner products  $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$  by the values of any kernel function  $k(\mathbf{x}_i, \mathbf{x}_j)$  like in standard SVM learning.

## 4 Support Vector Machine Learning with Taxonomies

### 4.1 Deriving Class Attributes from Taxonomies

The application of the method presented in the previous section to classification problems with pre-defined taxonomies is straightforward. The main idea is to encode the relationship between classes, expressed in the taxonomy, in terms of a class attribute representation. We define a taxonomy as an arbitrary lattice (e.g. tree) whose minimal elements (e.g. leaves) correspond to the classes. Non-minimal elements, i.e. interior nodes corresponding to super-classes are denoted by  $\mathcal{Z} = \{z_1, \dots, z_R\}$ . Then we define the class attributes by

$$\lambda_r(y) = \begin{cases} v_r, & \text{if } z_r \text{ is a superclass of class } y \\ 0, & \text{otherwise,} \end{cases} \quad (13)$$

where  $v_r \geq 0$  are non-negative weights that are all chosen to be equal to 1 in the simplest case, such that  $\lambda_r$  becomes an indicator function. Other choices for  $v_r$  are, for example, setting all  $v_r$  equal to a constant for nodes  $z_r$  at the same depth in the lattice.

Defining class attributes via common predecessors in the taxonomy leads to a very intuitive decomposition of the discriminant function into contributions from all nodes along the paths from a root to a specific leaf. Hence (7) becomes

$$F(\mathbf{x}, y; \mathbf{w}) = \sum_{z: y \prec z} \lambda_z(y) \langle \mathbf{w}_z, \mathbf{x} \rangle, \quad (14)$$

where we use the relation  $\prec$  to denote that a node  $y$  is a successor of a node  $z$  (interior or terminal).

### 4.2 Hierarchical Loss Functions

A shortcoming of the approach presented so far is that it is based on the standard misclassification loss. However, in many applications the actual loss of an incorrect prediction will depend on the relation of the classes. In particular, it is reasonable to assume that confusing classes that are “nearby” in the taxonomy is less costly/severe than predicting a class that is “far away” from the correct class. Hence, we would like to work with general loss functions  $\Delta \in \mathfrak{R}^{Q \times Q}$  where  $\Delta(y, \hat{y})$  denotes the loss in predicting  $\hat{y}$ , when the true class is  $y$ . We assume that  $\Delta(y, y) = 0$  and that  $\Delta(y, \hat{y}) > 0$  for  $y \neq \hat{y}$ . Two problems need to be addressed: (i) how to define meaningful loss functions for taxonomies and (ii) how to modify the SVM formulation to more directly minimize (an upper bound on) the desired loss.

**Defining Hierarchical Loss Functions** As far as the first question is concerned, we propose to design loss functions in the following way. With each node  $z$  in the taxonomy we associate two costs  $c_z \geq 0$  and  $\bar{c}_z \geq 0$ ;  $c_z$  denotes the cost of assigning an item of  $z$  to a different node, whereas  $\bar{c}_z$  denotes the cost of assigning to node  $z$  an item that belongs to a different node. Here an item gets assigned to a node  $z$  if its predicted class  $\hat{y}$  is a successor of  $z$  in the taxonomy. It seems reasonable to assume that such costs can be solicited from domain experts in real-world applications. For instance, in text categorization, the cost of a node may be proportional to the number of readers/customers that make routing/filtering decision based on a particular node in the taxonomy. Now the loss of predicting a class  $\hat{y}$  instead of  $y$  can be defined formally as

$$\Delta(y, \hat{y}) = \sum_{\substack{z: y \prec z \\ \hat{y} \not\prec z}} c_z + \sum_{\substack{z: y \not\prec z \\ \hat{y} \prec z}} \bar{c}_z. \quad (15)$$

---

**Algorithm 1** Generalized multiclass SVM algorithm.

---

initialize  $\alpha_{iy} = 0$ ,  $\psi_i = 0$  and  $G_{iy} = 0$ ,  $\epsilon \in [0.1; 0.001]$   
pick instance index  $i$  at random  
**repeat**  
    optimize dual problem in (11) over  $\{\alpha_{iy} : y \in \mathcal{Y} - \{y_i\}\}$  using LOQO  
    update  $G_{iy}$  and  $\psi_i$   
    set  $i = \operatorname{argmax}_i \{\psi_i\}$   
**until**  $\psi_i < \epsilon$

---

Notice that the loss depends on the costs associated with nodes in the symmetric difference of the predecessors of the true and predicted class. In the case of a tree, the loss involves the costs for nodes on the path to the first common predecessor in the tree.

**Soft Margin Maximization with Arbitrary Loss Functions** It remains to generalize the presented SVM formulation to accommodate an arbitrary loss function  $\Delta$ . We propose to do that by scaling the penalties for margin violations proportional to the loss. Hence, the linear margin constraints take the following form

$$\langle \mathbf{w}, \Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, y) \rangle \geq 1 - \frac{\xi_i}{\Delta(y_i, y)} \quad \forall y \neq y_i, \quad (16)$$

leading to minor modifications in the constraints of the dual problem, specifically a different upper bound constraint on the dual variables

$$\sum_{y \neq y_i} \frac{\alpha_{iy}}{\Delta(y_i, y)} \leq C. \quad (17)$$

The rationale behind (16) is that  $\frac{1}{n} \sum_{i=1}^n \xi_i$  will now provide an upper bound on the training loss with respect to  $\Delta$ , the proof of which is straightforward.

## 5 Optimization Algorithm

Notice that the constraints in the dual problem (11) factorize over the instance index  $i$ , which means the constraints do not couple dual variables belonging to different training instances. This can be exploited in an optimization procedure which iteratively performs subspace optimization over all dual variables  $\alpha_{iy}$  belonging to the same training instance. Notice that since class attribute vectors  $\lambda(y)$  are in general not orthogonal, we can not use the fixed point method proposed in [2]. However, we can use a standard QP solver instead. In our experiments, we have used the LOQO optimization package [6].

The remaining issue is how to select the next training instance for the subspace ascent. Following [2] we propose to derive a criterion based on the violation of the KKT conditions

$$\alpha_{iy} [\xi_i - \Delta(y_i, y) (1 - \langle \mathbf{w}, \Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, y) \rangle)] = 0. \quad (18)$$

To that extent we define

$$G_{iy} \equiv \Delta(y_i, y) \left[ \sum_j \sum_{y' \neq y_j} \alpha_{jy'} \langle \Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, y), \Phi(\mathbf{x}_j, y_j) - \Phi(\mathbf{x}_j, y') \rangle - 1 \right] \quad (19)$$

and furthermore  $\psi_i \equiv \max_{y: \alpha_{iy} > 0} G_{iy} - \min_y G_{iy}$ . At every step of the algorithm, the next instance for subspace optimization is selected based on  $i^* = \operatorname{argmax}_i \psi_i$ .

	flat 0/1	tax 0/1	flat $\Delta$	tax $\Delta$	rel. improv.
<i>4 training instances per class</i>					
Classification accuracy	28.32	28.32	27.47	29.74	+ 5.01 %
$\Delta$ -loss	1.36	1.32	1.30	1.21	+ 12.40 %
<i>2 training instances per class</i>					
Classification accuracy	20.20	20.46	20.20	21.73	+7.57 %
$\Delta$ -loss	1.54	1.51	1.39	1.33	+13.67 %

Table 1: Results on the WIPO-alpha corpus, section D with 160 groups using 3-fold cross validation over 572 training instances. 'flat' is a standard SVM multiclass model, 'tax' the hierarchical architecture. '0/1' denotes training based on the classification loss, ' $\Delta$ ' refers to training based on the tree loss.

	flat 0/1	tax 0/1	flat $\Delta$	tax $\Delta$	naive Bayes	perceptron
Accuracy	71.5	71.5	71.0	71.9	68.0	70.4
$\Delta$ -loss	0.1612	0.1622	0.1648	0.1604	-	-

Table 2: Results for word sense classification. 'flat' is a standard SVM multiclass model, 'tax' the hierarchical architecture. '0/1' denotes training based on the classification loss, ' $\Delta$ ' refers to training based on the tree loss. As a reference we have included results obtained by naive Bayes and the multiclass perceptron learning.

## 6 Experiments

### 6.1 Text Categorization

The first set of experiments has been conducted on a corpus of patents released by the World Intellectual Property Organization and known as the WIPO-alpha collection. We have restricted ourselves to one of the 8 sections, namely section D, in which there are a total of 1,710 documents in the WIPO-alpha collection. For our experiments, we have indexed the title and claim tags. We have furthermore sub-sampled the training data to investigate the effect of the training set size. Document parsing, tokenization and term normalization have been performed with the MindServer retrieval engine<sup>2</sup>

As one can see from the results summarized in Table 1, the proposed SVM learning architecture improves performance over the standard multiclass SVM in terms of classification accuracy as well as the tree loss derived from the taxonomy (with  $c_z = \bar{c}_z = \frac{1}{2}$ ). As expected the gain in terms of the test loss using  $\Delta$  is more substantial.

### 6.2 Word Sense Classification

The second set of experiments concerns a word sense disambiguation task based on the Senseval-2 data set. We restricted ourselves to the task of disambiguating nouns. The data consists of 5,266 textual passages (3,512 for training and 1,754 for testing) containing an ambiguous noun which has been manually annotated using the word sense inventory of WordNet. In these experiments the hierarchical classifiers use a simplified version of the WordNet taxonomy that consists only of two layers. The leaf layer comprises all the word senses, while the superordinate layer consists of 26 broad semantic classes designed by the lexicographers that develop the WordNet

<sup>2</sup><http://www.recommind.com>

ontology. More details about the experimental setup of this second evaluation set can be found in [1].

The results for word sense classification are summarized in Table 2. Compared to the text categorization data, the achieved gains are less substantial, however, to our best knowledge the accuracy of the hierarchical SVM is better than that of any previously published result on this task. We are currently investigating tasks involving more than just two hierarchy levels, where we expect to see a more significant improvement.

## 7 Conclusion

We have presented an extension of multiclass SVM learning that can incorporate arbitrary class attributes and have demonstrated how to apply this technique to classification problem involving taxonomies. Preliminary experimental results for text categorization have shown substantial improvement in terms of accuracy and test loss, whereas the improvements for word sense classification have been less significant, but are slightly better than previous results reported in the literature.

## Acknowledgments

We would like to thank Ioannis Tsochantaridis, Alex Smola, Thorsten Joachims, Mark Johnson, and Yasemin Altun for helpful discussions. We are grateful to Jan Puzicha and the rest of the RecomMind team for making the MindServer retrieval engine available. The WIPO-alpha corpus is distributed courtesy of the World Intellectual Property Organization.

## References

- [1] M. Ciaramita, T. Hofmann, and M. Johnson. Hierarchical semantic classification: Word sense disambiguation with world knowledge. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, 2003.
- [2] K. Crammer and Y. Singer. On the algorithmic implementation of multi-class kernelbased vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.
- [3] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts, 1998.
- [4] Andrew McCallum, Ronald Rosenfeld, Tom Mitchell, and Andrew Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 359–367, 1998.
- [5] Kristina Toutanova, Francine Chen, Kris Popat, and Thomas Hofmann. Text classification in a hierarchical mixture model for small training sets. In *Proceedings of the Tenth International ACM Conference on Information and Knowledge Management (CIKM)*, 2001.
- [6] R. J. Vanderbei. LOQO: An interior point code for quadratic programming. *Optimization Methods and Software*, 11:451–484, 1999.
- [7] J. Weston and C. Watkins. Multi-class support vector machines, 1998.
- [8] WIPO World Intellectual Property Organization. International patent classification. URL, 2001. <http://www.wipo.int/classifications/en/>.