
Learning Hierarchies at Two-class Complexity

Sandor Szedmak
ss03v@ecs.soton.ac.uk

Craig Saunders
cjs@ecs.soton.ac.uk

John Shawe-Taylor
jst@ecs.soton.ac.uk
ISIS Group, Electronics and Computer Science
University of Southampton, SO17 1BJ, United Kingdom

Juho Rousu
juho.rousu@cs.helsinki.fi
Department of Computer Science, PO Box 68, FI-00014
University of Helsinki, Finland

Abstract

It is assumed that to learn discriminative identification function when the output space is a labelled hierarchy is a much more complex problem than binary classification. In this presentation we show the complexity of this kind of problem can be detached from the optimisation model and can be expressed by an embedding into a Hilbert space. This allows a universal optimisation model processing Hilbertian inputs and outputs to be used to solve the optimisation task without tackling with the underlying structural complexity. The optimisation model is an implementation of a certain type of maximum margin regression, an algebraic generalisation of the well-known Support Vector Machine. The computational complexity of the optimisation scales only with the number of input-output pairs and it is independent from the dimensions of both spaces. Furthermore its overall complexity is equal to binary classification. Our approach can be easily be extended towards other structural learning problems with the same optimisation framework. We demonstrate the high performance of the proposed method on the WIPO and the Reuters datasets, where our task is to predict a complete classification hierarchy for each example.

1 Introduction

One of the most popular streams that has recently emerged from machine learning research is to find efficient methods for structural learning. Several researchers

introduced approaches to this kind of problems, including [5], [1], [3], [6] and [4]. These methods directly incorporate the structural learning into a specially chosen optimisation framework. Here we show an alternative formulation where the structural complexity of the hierarchy, or other structural learning tasks, can be separated from the computational complexity of the optimisation problem realizing the base learning algorithm. We have three fundamental phases of the structural learning:

Embedding The structures of the input and output objects are represented as abstract vectors in properly chosen Hilbert spaces reflecting the similarity and the dissimilarity of the objects.

Optimisation The optimisation phase is implemented via a universal solver which tries to find the best similarity based matching between the input and the output representations. Since these representation are expressed as general vectors, the optimiser needs not directly tackle the underlying structural complexity.

Inversion The optimiser provides a decision function which emits a vector. The inversion phase has to find the best fitting output structure by projecting the image vector back. This is often referred to as the pre-Image problem, see e.g. [2] for example. If the embedding is realized as a bijective mapping the inversion task is well defined.

If the complexity is shifted from the algorithm towards the embedding problem in this fashion, then a critical question that arises is what kind of theory is appropriate to answer the questions about the generalisation ability; namely, how does one measure the complexity of function class with vector outputs?

2 Formulation of the optimisation problem

Assume we have a sample S of pairs $\{(\mathbf{y}_i, \mathbf{x}_i) : \mathbf{y}_i \in \mathcal{Y}, \mathbf{x}_i \in \mathcal{X}, i = 1, \dots, m\}$ independently and identically generated by an unknown multivariate distribution. Also we have two embeddings of the input and output objects into Hilbert spaces respectively called feature space and label space, that are represented by the functions $\phi : \mathcal{X} \rightarrow \mathcal{H}_\phi$ and $\psi : \mathcal{Y} \rightarrow \mathcal{H}_\psi$.

The Maximum Margin Regression(MMR) is a certain type of Support Vector Machine with vector outputs that is realized on this sample by the following optimisation problem

$$\begin{aligned}
\min \quad & \frac{1}{2} \text{tr}(\mathbf{W}^T \mathbf{W}) + C \mathbf{1}^T \boldsymbol{\xi} \\
\text{w.r.t.} \quad & \{\mathbf{W} | \mathbf{W} : \mathcal{H}_{\phi(x)} \rightarrow \mathcal{Y}, \mathbf{W} \text{ linear operator}\}, \\
& \{\mathbf{b} | \mathbf{b} \in \mathcal{H}_\psi, \text{ bias vector}\}, \\
& \{\boldsymbol{\xi} | \boldsymbol{\xi} \in \mathbb{R}^m, \text{ slack or error vector}\} \\
\text{s.t.} \quad & \langle \boldsymbol{\psi}(\mathbf{y}_i), (\mathbf{W}\boldsymbol{\phi}(\mathbf{x}_i) + \mathbf{b}) \rangle_{\mathcal{H}_\psi} \geq 1 - \xi_i, \quad i = 1, \dots, m, \\
& \boldsymbol{\xi} \geq \mathbf{0}.
\end{aligned} \tag{1}$$

where $\mathbf{0}$ and $\mathbf{1}$ denote the vectors with components 0 and 1 respectively.

Calculating the dual gives

$$\begin{aligned}
\min \quad & \sum_{i,j=1}^m \alpha_i \alpha_j \overbrace{\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle}^{\kappa_{ij}^\phi} \overbrace{\langle \boldsymbol{\psi}(\mathbf{y}_i), \boldsymbol{\psi}(\mathbf{y}_j) \rangle}^{\kappa_{ij}^\psi} - \sum_{i=1}^m \alpha_i, \\
\text{w.r.t.} \quad & \{\alpha_i | \alpha_i \in \mathbb{R}\}, \\
\text{s.t.} \quad & \sum_{i=1}^m (\boldsymbol{\psi}(\mathbf{y}_i))_t \alpha_i = 0, \quad t = 1, \dots, \dim(\mathcal{H}_\psi), \\
& 0 \leq \alpha_i \leq C, \quad i = 1, \dots, m,
\end{aligned} \tag{2}$$

where we can write the values of inner products in the objective as kernel items where κ_{ij}^ϕ and κ_{ij}^ψ stand for the elements of the kernel matrices for the feature vectors and for the label vectors respectively. Hence, the vector labels are kernelized as well. The synthesised kernel is the element-wise product of the input and the output kernels, an operation that preserves positive semi-definiteness. The explicit occurrences of the embedded labels in the dual can be eliminated by recognising that the constraints $\sum_{i=1}^m (\psi(\mathbf{y}_i))_t \alpha_i = 0$, $t = 1, \dots, \dim(\mathcal{H}_\psi)$, can be replaced with $\sum_{i=1}^m \kappa_{ij}^\psi \alpha_i = 0$, $j = 1, \dots, m$, without modifying the optimum solution.

3 Implementation of the Hierarchy Learning

As mentioned previously in this paper we will focus on the case where the output space is a labelled hierarchy. The hierarchy learning is realized via an embedding of each path going from a node to the root of the tree. Let V be the set of nodes in the tree. A path $p(v) \subset V$ is defined as a shortest path from the node v to the root of the tree and its length is equal to $|p(v)|$. The set $I = 1, \dots, |V|$ gives an indexing of the nodes. The embedding is realized by a vector valued function $\psi : V \rightarrow \mathbb{R}^{|V|}$, and the components of $\psi(v)$ are given by

$$\psi(v)_i = \begin{cases} r & \text{if } v_i \notin p(v), \\ sq^k & \text{if } v_i \in p(v) \text{ and } k = |p(v)| - |p(v_i)|, \end{cases} \quad (3)$$

where r, q, s are the parameters of the embedding. The parameter q can express the diminishing weight of the nodes being closer to the root. If $q = 0$, assuming $0^0 = 1$, then the intermediate nodes and the root are disregarded, thus we have a simple multiclass classification problem. The value of r can be 0 but some experiments show it may help to improve the classification performance. We might conjecture the best choice of the parameters are those which minimises the correlation between all pairs of the label vectors. In this case the optimal values can be derived by an auxiliary optimisation problem. Note that this is one particular choice of embedding, and that others can be used for hierarchical outputs (or indeed different structural domains). We simply choose this one here as an illustration: it satisfies some aspects of our notion of similarity between hierarchies (nodes in the leaves have different similarity than those closer to the root) and it allows us to easily carry out the inversion step necessary for prediction.

4 Experiments

We tested the presented learning approach on two publicly available document collection that have an associated classification hierarchy:

- Reuters Corpus Volume 1, RCV1. 2500 documents were used for training and 5000 for testing. As the label hierarchy we used the 'CCAT' family of categories, which had a total of 34 nodes, organised in a tree with maximum depth 3. The tree is quite unbalanced, half of the nodes residing in depth 1.
- WIPO-alpha patent dataset [7]. The dataset consisted of the 1372 training and 358 testing document comprising the D section of the hierarchy. The number of nodes in the hierarchy was 188, with maximum depth 3. Each document in this dataset belongs to exactly one leaf category, hence it contains no multiple or partial paths.

Both datasets were processed into bag-of-words representation with TFIDF weighting. No word stemming or stop-word removal was performed.

Table 1: Prediction losses $l_{0/1}$ and l_{Δ} , precision, recall and F1 values obtained using different learning algorithms. All figures are given as percentages. Precision and recall are computed in terms of totals of microlabel predictions in the test set.

REUTERS	$l_{0/1}$	l_{Δ}	P	R	F1
SVM	32.9	0.61	94.6	58.4	72.2
H-SVM	29.8	0.57	92.3	63.4	75.1
H-RLS	28.1	0.55	91.5	65.4	76.3
H-M ³ - l_{Δ}	27.1	0.58	91.0	64.1	75.2
H-M ³ - $l_{\bar{H}}$	27.9	0.59	85.4	68.3	75.9
MMR _{lin}	27.8	0.71	82.7	60.4	69.8
MMR _{poly}	26.4	0.70	85.2	59.1	69.8
WIPO-alpha	$l_{0/1}$	l_{Δ}	P	R	F1
SVM	87.2	1.84	93.1	58.2	71.6
H-SVM	76.2	1.74	90.3	63.3	74.4
H-RLS	72.1	1.69	88.5	66.4	75.9
H-M ³ - l_{Δ}	70.9	1.67	90.3	65.3	75.8
H-M ³ - $l_{\bar{H}}$	65.0	1.73	84.1	70.6	76.7
MMR _{lin}	47.1	1.77	77.8	77.8	77.8

The test results are summarised in Table 1. The values for comparison are taken from [4] which used the same data sets. Briefly the other methods are as follows: SVM is a flat SVM trained on each node individually, H-SVM is an SVM where only training examples for which the parent node has a positive label were used. In both of these cases the data is post-processed in a root-down fashion such that once a negative prediction is encountered on a path, then all subsequent nodes on the path are relabelled as negative; see [4] for details. H-M³ is an efficient approach to structured prediction that can be trained with different loss functions, but however solves a similar optimisation problem to that given in [5]. The loss function $l_{0/1}$ considers a prediction of a path being correct if the full path from the node to the root is correct and the loss function l_{Δ} applies the symmetric difference to measure the correctness; precision, recall and F1 values for the entire tree are also given. More details about the possible loss functions for hierarchies are in [4].

The results for the MMR algorithm are computed by using a linear kernel with penalty parameter $C = 1$. The parameters r, q, s are set to $(0, 0, 1)$ in case of the Reuters dataset and to $(-0.1, 0.1, 1)$ for the WIPO data. In the latter case, since it has a balanced tree and only the leaves have to be considered, the optimal values can be estimated via a simple auxiliary problem. The former case seems to be harder to estimate and we use one of the simplest settings. In case of the Reuters dataset we give a result computed by polynomial kernel to show that a sophisticated input kernel can improve the expected accuracy.

Table 2 demonstrates the computational efficiency of our approach. It must be noted that all of the other structured prediction algorithms take between 40 minutes and several hours to train and predict. For example a flat SVM trained on each node individually takes approximately 20 seconds for each node when using SVM-light.

Table 2: The computational times of the optimiser in seconds

	Reuters	WIPO-alpha
mmr _{lin}	2.8	1.9

5 Conclusion

We introduced an alternative framework to structural learning. It opens the door for new theoretical foundation and allows us to tackle very large-scale problems efficiently. In this paper we only considered the scenario where the output spaces were hierarchies, however the structure of the optimisation problem can easily be extended towards other kind of representations living in a Banach space; since linearity of the margin constrains in the primal is the only restriction that we require.

References

- [1] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden markov support vector machines. In *ICML'03*, pages 3–10, 2003.
- [2] G.H. Bakir, J. Weston, and B. Schlkopf. Learning to find pre-images. volume 16, pages 449–456, Cambridge, MA, USA, 2004. MIT Press.
- [3] O. Dekel, J. Keshet, and Y. Singer. Large margin hierarchical classification. In *ICML'04*, pages 209–216, 2004.
- [4] J. Rousu, C.J. Saunders, S. Szedmak, and J. Shawe-Taylor. Learning hierarchical multi-category text classification models. In *ICML*. 2005.
- [5] B. Taskar, C. Guestrin, and D. Koller. Max margin markov networks. In *NIPS 2003*. 2003.
- [6] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6(Sep):1453–1484, 2005.
- [7] WIPO. *World Intellectual Property Organization*. 2001. <http://www.wipo.int/classifications/en>.