Multiple Kernel Learning for Prediction on Unknown Graph Structures

Simon Cousins and John Shawe-Taylor
Department of Computer Science
University College London
London, UK
s.cousins@cs.ucl.ac.uk

Mario Marchand
Département dinformatique et génie logiciel
Université Laval
Québec (QC), Canada
mario.marchand@ift.ulaval.ca

Juho Rousu and Hongyu Su
Helsinki Institute for Information Technology
Dept of Information and Computer Science
Aalto University, Finland
juho.rousu@aalto.fi

Abstract

The paper considers the problem of structured output learning when the output graph structure is not known and must be inferred during learning. Building on the work of [1] in which a random set of spanning trees is used to approximate the margin, the paper shows that there exists a unit L_1-norm constrained combination of spanning trees that achieves the full margin of the unknown output graph. It is shown that optimising the margin under this constraint corresponds to a convex optimisation problem that combines structured output and multiple kernel learning. Inference of the maximum violators is achieved by a generalisation of the method of [1] that searches the K-best lists for the chosen trees. Structure is interpreted as the final weights attached to edges and it is shown that the exponentially large set of all possible spanning trees of the complete output graph can be efficiently explored using a simple implementation of the maximum spanning tree algorithm. Preliminary experiments with the method show encouraging results in agreement with the theoretical analysis.

1 Introduction

Structured output prediction is an increasingly popular method for learning about real-world problems. It provides a learning framework that is able to leverage the information provided by label correlations to improve prediction accuracy and has been used successfully across a wide range of applications. The *StructuredSVM*[2] is a popular approach to learning the parameters of these prediction models but despite being a convex optimisation procedure, its applicability is often limited by the computationally expensive training procedure induced by the margin constraints that require each training example to be compared to the inferred label. For certain models exact inference can be performed efficiently (e.g. tree-structured graphs, planar Ising models, matchings), however exact inference is generally known to be intractable for most graph structures. A further factor limiting the application of structured output prediction to real-world problems is the absence of an explicit output graph structure on which to learn. This is traditionally estimated before learning begins using heuristics that take into consideration prior knowledge of the problem domain and the scalability of the solution. In this paper we propose to address both of these issues by learning using a weighted combination of spanning trees over the complete output graph.

2 Related work

This paper builds on the work presented in [1], where the authors showed that a random sample of spanning trees could obtain a significant fraction of the margin obtainable on the complete output graph, here we show that we can obtain a larger margin and we actively include and remove trees according to the MKL criteria. In [7] the authors sought to find the optimal tree structure prior to learning using a maximum spanning tree, however edge scores were defined using heuristics. Our edge scores are determined by their current contribution to the margin and our goal is to learn a structure more complex than a tree. In [8] the authors introduce a tree inducing regularisation term that penalises non-tree structures, which could possibly ignore higher-order interactions between variables. In [10] the authors deal with the case of efficiently learning from a large set of kernels, providing a solution to the case where the kernel was parameterised by a small number of real valued parameters that were differentiable, however they asked whether efficient methods existed for kernels that come from a combinatorial set. Here we address this question by showing that the maximum spanning tree algorithm can be used to generate such candidate kernels for the ensemble.

3 MKL Formulation

We consider the general supervised learning problem with an arbitrary input space $\mathcal{X}$ and output space $\mathcal{Y}$ consisting of the set of all ℓ-dimensional multilabel vectors $\mathbf{y} = (y_1, y_2, \ldots, y_\ell)$, where each label $y_i \in Y_i$ takes on one of r_i possible positive integer values. Each example $(x, \mathbf{y})$ is mapped to the joint feature space $\boldsymbol{\phi}(x, \mathbf{y}) = \rho(x) \otimes \psi(\mathbf{y})$ of the input ρ and output ψ feature space. Our goal is to find the predictor $\mathbf{w}$ residing within the joint feature space that best models the distribution of possible labels $\mathbf{y} \in \mathcal{Y}$ conditioned on the input $x \in \mathcal{X}$. Let $F(x, \mathbf{y}; \mathbf{w}) = \langle \mathbf{w}, \boldsymbol{\phi}(x, \mathbf{y}) \rangle$ be the score of example $(x, \mathbf{y})$ evaluated at $\mathbf{w}$. We assume joint feature map $\boldsymbol{\phi}$ is defined over the complete graph G consisting of ℓ-nodes $\binom{\ell}{2}$ undirected edges and the score function is given by $F(x, \mathbf{y}; \mathbf{w}) = \sum_{(i,j) \in G} \langle \mathbf{w}_{ij}, \boldsymbol{\phi}_{ij}(x, y_i, y_j) \rangle$.

In [1] the authors showed that under the assumption of a normalised joint feature space $||\boldsymbol{\phi}(x, \mathbf{y})|| = 1$ and a unit norm predictor $||\mathbf{w}|| = 1$, the score function over the complete output graph can be expressed as the expectation of score function over trees $\langle \mathbf{w}, \boldsymbol{\phi}(x, \mathbf{y}) \rangle = \frac{\ell}{2} \mathbf{E}_{T \sim U(G)} \langle \mathbf{w}_T, \boldsymbol{\phi}_T(x, \mathbf{y}) \rangle$, where $U(G)$ is the uniform distribution over the set of all spanning trees $S(G)$ for the complete graph G. Here $\mathbf{w}_T$ is the projection of $\mathbf{w}$ onto tree T with $(\mathbf{w}_T)_{(i,j)} = \mathbf{w}_{i,j}$ if $(i, j) \in T$ and zero otherwise, and $\boldsymbol{\phi}_T$ is defined similarly. This allows us to express the score function as $\langle \mathbf{w}, \phi(x, \mathbf{y}) \rangle = \sum_{T \in S(T)} \hat{a}_T \langle \hat{\mathbf{w}}_T, \hat{\boldsymbol{\phi}}_T(x, \mathbf{y}) \rangle$ where $\hat{a}_T = \sqrt{\frac{\ell}{2}} ||\mathbf{w}_T|| / \ell^{\ell-2}$ and $\sum_{T \in S(T)} \hat{a}_T \leq 1$ (see [1] for further details).

If we assume a margin $\gamma > 0$ can be achieved by learning over the complete graph, then this can be replicated by assigning a weight $\hat{a}_T$ to each tree $T \in S(G)$ and learning over this ensemble of trees. This leads to our argument that by reformulating the problem in terms of unit L_1-norm multiple kernel learning over the space of all possible trees, we can achieve a margin at least as large as that obtainable by learning over the complete output graph. This comes from the fact that weights $\hat{a}_T$ represent a feasible but not necessarily optimal solution to the unit L_1-norm MKL problem. It can be shown that by dropping the hard margin assumption and introducing slack variables, the large margin learning problem reduces to the familiar looking optimisation presented in Definition 1.

Definition 1. *L_1-norm MKL for Spanning Trees*

$$\min_{\mathbf{w}_T, \xi} \quad \frac{1}{2} \left(\sum_{T \in S(G)} ||\mathbf{w}_T|| \right)^2 + C \sum_{k=1}^{m} \xi_k \tag{1}$$

$$\textit{s.t.} : \min_{\mathbf{y} \neq \mathbf{y}_K} \sum_{T \in S(G)} \langle \mathbf{w}_T, \hat{\boldsymbol{\phi}}_T(x_k, \mathbf{y}_k) - \hat{\boldsymbol{\phi}}_T(x_k, \mathbf{y}) \rangle \geq 1 - \xi_k, \;\; \xi_k \geq 0 \forall \;\, k, .$$

4 Optimisation

To run the optimisation we make use of the dual form of the optimisation given in Defintion 1. We alternate between solving the structured output problem using a fixed ensemble of spanning trees and updating the weights of trees in, and possibly adding trees to, the ensemble.

4.1 Structured Output Learning

We use a variant of the Frank-Wolfe algorithm [3] during the structured output learning stage, selecting a particular example and updating its dual variables in the direction proposed by the inference scheme. The problem of exact inference is addressed by extending the K-best dynamic programming algorithm first presented in [1] to the case where each tree $T \in S(G)$ has a weight given by $\lambda_T \geq 0$.

Lemma 1. *Let* $\mathbf{y}_K^* = \mathop{\mathbf{argmax}}_{\mathbf{y} \in \mathcal{Y}_{\mathcal{T},K}} F(x, \mathbf{y}; \mathbf{w})$ *be the highest scoring multilabel in* $\mathcal{Y}_{\mathcal{T},K}$*, where* $\mathcal{T} = \{T \in S(G) : \lambda_T > 0\}$. *Suppose that*

$$F(x, \mathbf{y}_K^*; \mathbf{w}) \geq \sum_{T \in S(G)} \lambda_T \hat{F}_T(x, \mathbf{y}_{T,K}; \mathbf{w}_T) = \theta_x(K),$$

where $F_T(x, \mathbf{y}) = \langle \mathbf{w}_T, \boldsymbol{\phi}_T(x, \mathbf{y}) \rangle$*, Then it follows that* $F(x, \mathbf{y}_K^*; \mathbf{w}) = \max_{\mathbf{y} \in \mathcal{Y}} F(x, \mathbf{y}; \mathbf{w})$. *Furthermore, if* $F(x, \mathbf{y}_k^*; \mathbf{w}) < \theta_x(K)$ *then* $\max_{\mathbf{y} \in \mathcal{Y}} F(x, \mathbf{y}; \mathbf{w}) \leq \theta_x(K)$.

This lemma states that we can use any K that satisfies the criteria above and be sure that $\mathbf{y}_K^*$ is the maximum scoring example on F. This inference step is used to both make predictions and find the direction in which to update the model. However during the model update it may be more efficient to perform only approximate inference and this lemma provides us with a method of measuring the maximum degree of sub-optimality of our inference i.e. $F(x, \mathbf{y}_K^*; \mathbf{w}) / \theta_x(K)$. These approximate rates of inference can be used in conjunction with the Frank-Wolfe rates of convergence for approximate convergence presented in [3]. Note that the disadvantages of performing approximate inference are much less pronounced at the beginning of the optimisation when we are likely to be far from the true solution, both in terms of the active constraints and the tree weightings used in the ensemble, here small steps towards candidate violators can significantly improve the value of the objective function.

4.2 Tree Learning

In the traditional MKL setup where there is a fixed set of kernels, we can simply iterate between updating kernel weights and solving the quadratic program until some convergence criteria is met. Our problem is more difficult in that we have an exponentially large set of kernels, $|S(G)| = \ell^{\ell-2}$, making it intractable to consider them all at once. To overcome this we propose to incrementally add and remove kernels from the active set $\mathcal{T} := \{T \ : \ \lambda_T > 0\}$ based upon their violation of the KKT conditions for optimality. Let $V_T(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{k,\bar{\mathbf{y}}} \sum_{j,\tilde{\mathbf{y}}} \alpha_k(\bar{\mathbf{y}}) \alpha_j(\tilde{\mathbf{y}}) \tilde{K}_T(x_k, \bar{\mathbf{y}}; x_j, \tilde{\mathbf{y}})$, where $\tilde{K}_T$ is the kernel function for feature mapping $\tilde{\boldsymbol{\phi}}_T(x_k, \bar{\mathbf{y}}) = \boldsymbol{\phi}_T(x_k, \mathbf{y}_k) - \boldsymbol{\phi}_T(x_k, \bar{\mathbf{y}})$ and $\alpha_k(\bar{\mathbf{y}})$ is the dual variable for the constraint $\sum_{T \in S(G)} \langle \mathbf{w}_T, \hat{\boldsymbol{\phi}}_T(x_k, \mathbf{y}_k) - \hat{\boldsymbol{\phi}}_T(x_k, \mathbf{y}) \rangle \geq 1 - \xi_k$. When adding a tree to $\mathcal{T}$ we look to find the tree $T^* = \mathbf{argmax}_{\,T \in S(G)} V_T(\boldsymbol{\alpha})$. By decomposing the kernel into its edge components we can write the objective function we are optimising simply as a sum over edges

$$V_T(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{(v,v') \in T} \sum_{k,\bar{\mathbf{y}}} \sum_{j,\tilde{\mathbf{y}}} \alpha_{k,r} \alpha_{j,s} \langle \tilde{\boldsymbol{\phi}}_{vv'}(x_k, \bar{y}_v, \bar{y}_{v'}), \tilde{\boldsymbol{\phi}}_{vv'}(x_j, \tilde{y}_v, \tilde{y}_{v'}) \rangle$$

It is a well-known that the solution of this optimisation can be found using the maximum spanning tree algorithm [4] and we see that the search over an exponentially large set of kernels can be reduced to a simple algorithm that runs in $\mathcal{O}\left(|E| \log(|V|)\right)$ time.

5 Experiments

We evaluate the performance of the L_1 norm spanning trees (L_1-TA) method and compare it to the original random spanning tree algorithm (L_2-RTA), an L_1-norm MKL formulation (L_1-RTA), where a fixed set of trees are weighted according to the MKL framework and a standard SVM trained and optimised individually for each node. We use three real-world datasets, Emotions, Scene and Medical, along with two artificial ones, Circle10 and Circle50 generated according to the method outlined in [5]. We use 5-fold cross validation to select the model parameters and compute the results. The margin slack parameter C is evaluated over the set $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ and the number of initial trees $\mathcal{T}_0$ is tested over $\{5, 10, 20, 40\}$. Each method is presented with the same initial set of trees $\mathcal{T}_0$ where RTA maintains an equal weight for each tree, L_1-RTA adjusts the weights of $\mathcal{T}_0$ using MKL and L_1-TA, adjusts weights whilst adding and removing trees according to a criteria defined by the violation of the KKT conditions for optimality. For each dataset we use a linear kernel for the input space and the size of our K-best list is restricted by the number of labels in that particular dataset. We report both the multilabel and microlabel accuracies, where multilabel loss is indicates correct labelling and the microlabel accuracy measures fraction of correct labels i.e. $1 - \Delta(\mathbf{y}, \mathbf{y}')$. In Table 1 we present the multilabel and microlabel accuracies, using the different algorithms and implementing the fixed margin requirement. On the first four datasets we observe that the performance of each spanning tree method is relatively similar, all on average outperforming the standard SVM. On Circle50 we see that L_1-TA significantly outperforms the other methods in terms of both multilabel accuracy. We believe that this is due to the large number of labels in the Circle50 dataset and the inability of a relatively small number of random spanning trees to fully express the relationships between these labels. On the other hand, the L_1-TA criteria for including new trees is able to directly exploit the discriminative edges, whilst the re-weighting removes redundant trees from the ensemble and adds those with additional explanatory power.

Table 1: Fixed margin learning out of sample performance.

	Multilabel accuracy (%)				**Microlabel accuracy (%)**			
	SVM	L_2-RTA	L_1-RTA	L_1-TA	SVM	L_2-RTA	L_1-RTA	L_1-TA
Emotions	22.2	29.00	29.19	29.93	77.6	77.76	77.63	78.15
Scene	53.8	69.70	69.74	69.90	90.2	90.82	90.84	90.92
Medical	8.2	40.91	40.34	41.17	97.4	97.87	97.89	97.88
Circle10	71.1	96.47	96.88	96.97	95.3	99.47	99.54	99.58
Circle50	30.2	30.58	31.21	49.47	94.3	92.01	91.72	93.93

Margin scaling

The optimisation framework and experiments presented thus far focus on finding the predictor $\mathbf{w}$ that minimises the expected $0/1$ loss, however this measure of loss is not always suitable when considering structured outputs. To account for the accuracy of a prediction an arbitrary loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ can be introduced into the constraints. We follow [6] and scale the margin required by the loss of the candidate labelling $\mathbf{y}$ resulting in the the constraints $\langle \mathbf{w}, \tilde{\boldsymbol{\phi}}(x_k, \mathbf{y}) \rangle \geq \Delta(\mathbf{y}_k, \mathbf{y}) - \xi_k$ for each $\mathbf{y} \in \mathcal{Y}$. The Hamming loss function $\Delta(\mathbf{y}, \mathbf{y}') = \ell^{-1} \sum_{v=1}^{\ell} I[y_v \neq y'_v]$ is a natural choice for structured output learning, as well as benefiting from decomposability properties. The maximum violator in this case is given by $\hat{\mathbf{y}}_k = \mathbf{argmax}_{\mathbf{y} \in \mathcal{Y}_{\mathcal{T},K}} F(x_k, \mathbf{y}; \mathbf{w}) + \Delta(\mathbf{y}_k, \mathbf{y})$, which can be handled easily by the K-best inference algorithm due to the decomposability of the Hamming loss function over the edges of a tree. In Table 2 we present the mutliabel and microlabel accuracies having implemented the algorithms using the re-scaled margin requirement. The results are similar to those obtained for the fixed margin case with L_1-TA again significantly outperforming on the Circle50 dataset. On four out of five datasets, we observe a noticeable improvement in the accuracy of the predictor when the margin required is scaled by the Hamming loss function. We

Table 2: Margin scaled learning out of sample performance.

	Multilabel accuracy (%)			**Microlabel accuracy (%)**		
	L_2-RTA	L_1-RTA	L_1-TA	L_2-RTA	L_1-RTA	L_1-TA
Emotions	32.05	32.39	32.39	81.27	81.44	81.39
Scene	68.93	68.93	68.57	90.90	90.90	90.83
Medical	39.02	39.31	43.65	98.01	98.01	98.18
Circle10	96.47	96.98	97.19	99.47	99.62	99.66
Circle50	38.84	38.54	56.67	94.69	94.14	95.66

believe that this is largely due to the Hamming loss function being a more *natural* target to minimise in structured output problems. As a consequence, in large graph problems especially, we expect that a smaller number of active constraints are required as we no longer have to satisfy a margin of 1 for each violator that differs from the true labelling by only a single label. We believe that the Scene dataset didn't benefit from this adjustment due to its low label density, which effectively reduces the problem to multiclass in nature.

.

References

[1] Mario Marchand, Hongyu Su, Emilie Morvant, Juho Rousu, and John Shawe-Taylor. Multilabel structured output learning with random spanning trees of max-margin markov networks. In *Proceedings of Neural Information Processing Systems (NIPS)*, 2014.

[2] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, page 104. ACM, 2004.

[3] Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletscher. Block-coordinate {Frank-Wolfe} optimization for structural {SVMs}. In *Proceedings of The 30th International Conference on Machine Learning*, pages 53–61, 2013.

[4] Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[5] Wei Bian, Bo Xie, and Dacheng Tao. Corrlog: Correlated logistic models for joint prediction of multiple labels. In *International Conference on Artificial Intelligence and Statistics*, pages 109–117, 2012.

[6] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. *Advances in neural information processing systems*, 16:25, 2004.

[7] Joseph K Bradley and Carlos Guestrin. Learning tree conditional random fields. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 127–134, 2010.

[8] Ofer Meshi, Elad Eban, Gal Elidan, and Amir Globerson. Learning max-margin tree predictors. *arXiv preprint arXiv:1309.6847*, 2013.

[9] Andreas Argyriou, Raphael Hauser, Charles A Micchelli, and Massimiliano Pontil. A dc-programming algorithm for kernel selection. In *Proceedings of the 23rd international conference on Machine learning*, pages 41–48. ACM, 2006.

[10] Peter Gehler and Sebastian Nowozin. Infinite kernel learning. In *NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, 2008.

[11] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *The Journal of Machine Learning Research*, 7:1531–1565, 2006.