Taxonomic Prediction with Tree-Structured Covariances

Matthew Blaschko, Wojciech Zaremba, Arthur Gretton

Taxonomic Prediction with Tree-Structured Covariances

Matthew B. Blaschko[1,2,3], Wojciech Zaremba[1,2], and Arthur Gretton[4]

[1] Center for Visual Computing, École Centrale Paris, France
[2] Équipe Galen, INRIA Saclay, Île-de-France, France
[3] Université Paris-Est, LIGM (UMR CNRS), École des Ponts ParisTech, France
[4] Gatsby Computational Neuroscience Unit, University College London, UK

Abstract. Taxonomies have been proposed numerous times in the literature in order to encode semantic relationships between classes. Such taxonomies have been used to improve classification results by increasing the statistical efficiency of learning, as similarities between classes can be used to increase the amount of relevant data during training. In this paper, we show how data-derived taxonomies may be used in a structured prediction framework, and compare the performance of learned and semantically constructed taxonomies. Structured prediction in this case is multi-class categorization with the assumption that categories are taxonomically related. We make three main contributions: (i) We prove the equivalence between tree-structured covariance matrices and taxonomies; (ii) We use this covariance representation to develop a highly computationally efficient optimization algorithm for structured prediction with taxonomies; (iii) We show that the taxonomies learned from data using the Hilbert-Schmidt Independence Criterion (HSIC) often perform better than imputed semantic taxonomies. Source code of this implementation, as well as machine readable learned taxonomies are available for download from `https://github.com/blaschko/tree-structured-covariance`.

1 Introduction

In many fields where large numbers of objects must be categorized, including computer vision, bioinformatics, and document classification, an underlying taxonomic structure is applied. While such taxonomies are useful visualization tools to organize data, and to talk about inter-relationships between (sub)categories, it is less clear whether taxonomies can help to perform structured learning, or whether learned taxonomies outperform those imposed by domain experts.

Several learning algorithms have been developed that make use of user-imposed taxonomies, with the main goal being to improve discriminative performance by using hierarchical structure. For example, [1] proposed a learning framework that incorporated semantic categories, and [2] implemented structured output prediction based on a fixed taxonomic structure. For the most part, these previous works have have found that taxonomic structure results in slight improvements in performance at best, while sometimes decreasing performance. The empirical results in this article give strong evidence that this may

be the result of the user-imposed taxonomy not being aligned to the feature similarities in the data.

In this article, we make use of a non-parametric dependence measure, the Hilbert-Schmidt Independence Criterion (HSIC), to learn taxonomies. We establish the equivalence between taxonomies and tree structured covariance matrices, and show that the latter constitute a natural way to encode taxonomies in structured prediction problems (indeed, the HSIC is a regularizer for structured output SVM when taxonomies are used). Moreover, we use this tree structured covariance representation to develop a highly efficient algorithm for structured prediction with taxonomies, such that it can be used in large scale problems.

A number of approaches have been proposed for the discovery of taxonomic structure and relationships between classes. Dependency graphs and co-occurrences were modeled in [3, 4]. [5] proposed to perform a top-down greedy partitioning of the data into trees. Hierarchical clustering has been employed in [6, 7]. Marszałek and Schmid first made use of a semantic hierarchy [8], and later proposed to do a non-disjoint partition into a "relaxed hierarchy" which can then be used for prediction [9]. [10] assume a given taxonomy and then uses a group lasso structured sparsity regularizer with overlapping blocks conforming to the taxonomic structure. In contrast, we do not make the assumption implicit in the group lasso that individual features are exactly aligned with category concepts. [11] perform hierarchical categorization using a taxonomic feature map and loss, but perform an explicit feature map and do not gain the computational advantages arising from the use of tree structured covariance matrices. [12] consider structured prediction of hierarchically organized image labels using a latent variable method to estimate missing annotations in a weakly supervised setting. None of these methods has identified the relationship between hierarchical prediction and tree-structured covariance matrices. [2] made use of a learning framework that is perhaps the most similar to that employed here, based on structured output prediction. However, they did not learn the taxonomy using a non-parametric dependence measure as we do, but instead used a fixed taxonomic structure.

While these works all make use of some clustering objective distinct from the learning procedure, in contrast, this work employs the Hilbert-Schmidt Independence Criterion, which interestingly is coupled with the learning algorithm in its interpretation as a direct optimization of the function prior in ℓ_2 regularized risk with a taxonomic joint kernel map (cf. Equation (13) and Section 5).

Recent works addressing the machine learning aspects of taxonomic prediction include [13], which embeds a taxonomic structure into Euclidean space, while in contrast our method can efficiently learn from taxonomic structures without this approximation. [14] learn a tree structure in order to improve computational efficiency by only evaluating a logarithmic number of classifiers, while [15] relax this tree structure to a directed acyclic graph. Such greedy methods are advantageous when the number of categories is too large to evaluate exactly, while the current article addresses the problem of efficient learning when exact evaluation is desired.

In experiments on the PASCAL VOC [16], Oxford Flowers [17], and WIPO-alpha [18] datasets, we show that learned taxonomies substantially improve over hand-designed semantic taxonomies in many cases, and never perform significantly worse. Moreover, we demonstrate that learning using taxonomies is widely applicable to large datasets, thanks to the efficiency of our algorithm.

Our paper is organized as follows: in Section 2, we review structured output SVMs, following [19]. We proceed in Section 3 to establish the equivalence of taxonomies and tree structured covariance matrices. In Section 4, we show how tree structured covariance matrices may be incorporated into a structured output learning algorithm, and in particular that this representation of taxonomic structure results in substantial computational advantages. In Section 5, we determine how to learn edge lengths of a taxonomy given a fixed topology using the Hilbert-Schmidt Independence Criterion. Finally, Section 6 contains our experimental results.

2 Taxonomic prediction

Given a training set of data $\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n$, a structured output SVM with slack rescaling [19, 20] optimizes the following learning objective

$$\min_{w \in \mathbb{R}^d, \xi \in \mathbb{R}} \frac{1}{2}\|w\|^2 + C\xi \tag{1}$$

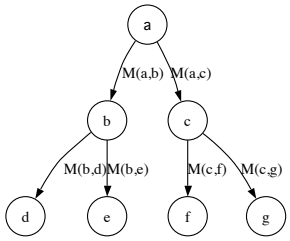
$$\text{s.t.} \sum_i \max_{\tilde{y}_i \in \mathcal{Y}} \left(\langle w, \phi(x_i, y_i) - \phi(x_i, \tilde{y}_i)\rangle - 1\right) \Delta(y_i, \tilde{y}_i) \geq -\xi \tag{2}$$

$$\xi \geq 0 \tag{3}$$

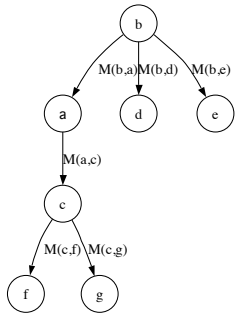
where ϕ is a joint feature map, and $\Delta(y_i, \tilde{y}_i)$ measures the cost of the erroneous prediction $\tilde{y}_i$ when the correct prediction should be y_i.

Cai and Hofmann proposed a special case of this learning framework in which $\mathcal{Y}$ is taxonomically structured [21]. In that setting, $\phi(x_i, y_i)$ decomposes as $\phi_y(y_i) \otimes \phi_x(x_i)$ and $\phi_y(y_i)$ is a binary vector that encodes the hierarchical relationship between classes. In particular, a taxonomy is defined to be an arbitrary lattice (e.g. tree) whose minimal elements (e.g. leaves) correspond to the categories. $\phi_y(y_i)$ is of length equal to the number of nodes in a taxonomy (equal to the number of categories plus the number of ancestor concepts), and contains non-zero entries at the nodes corresponding to predecessors of the class node. It is straightforward to extend this concept to non-negative entries corresponding to the relative strength of the predecessor relationship. The loss function employed may depend on the length of the shortest path between two nodes [22], or it may be the length of the distance to the nearest common ancestor in the tree [21].

We show in the next two sections that structured prediction with taxonomies is intimately tied to the concept of tree-structured covariance matrices.



(a) A binary rooted tree. Edges are annotated by their length. The tree metric is defined by the sum of the path lengths between two leaf nodes.



(b) Rerooting the tree by setting node "b" to the root. Distances between leaf nodes are preserved regardless of the rooting.

Fig. 1. An arbitrarily rooted binary tree may be rerooted without changing the pairwise distances between leaf nodes. Furthermore, rerooting has no effect on the value of $HSIC_{\text{cov}}$ (Section 5 and Theorem 2).

3 Tree-structured covariance matrices

Here we consider the structure of a covariance matrix necessary to encode taxonomic structure [23, 24].

Definition 1 (Partition property). *A binary matrix V of size $k \times (2k-1)$ has the partition property for trees of size k (i.e. having k leaves) if it satisfies the following conditions:*

1. *V contains the vector of all ones as a column*
2. *for every column w in V with more than one non-zero entry, it contains two columns u and v such that $u + v = w$.*

We now use this definition to construct a tree structured covariance matrix

Definition 2 (Tree covariance representation). *A matrix B is a tree-structured covariance matrix if and only if $B = VDV^T$ where D is a diagonal matrix with nonnegative entries and V has the partition property.*

This definition is chosen to correspond to [24, Theorem 2]. Such an encoding of tree-structured covariance matrices separates the specification of the topology of the tree, which is encoded in V, from the lengths of the tree branches, which is specified in D. As a concrete example, the tree structured covariance matrix

corresponding to Figure 1(a) is

$$V = \begin{pmatrix} 1\,1\,0\,1\,0\,0\,0 \\ 1\,1\,0\,0\,1\,0\,0 \\ 1\,0\,1\,0\,0\,1\,0 \\ 1\,0\,1\,0\,0\,0\,1 \end{pmatrix}, \tag{4}$$

$$D = \mathrm{diag}[0, M(a,b), M(a,c), M(b,d), M(b,e), M(c,f), M(c,g)]^T,$$

$$B = \begin{pmatrix} M(a,b)+M(b,d) & M(a,b) & 0 & 0 \\ M(a,b) & M(a,b)+M(b,e) & 0 & 0 \\ 0 & 0 & M(a,c)+M(c,f) & M(a,c) \\ 0 & 0 & M(a,c) & M(a,c)+M(c,g) \end{pmatrix}$$

Section 3.1 derives a mapping between tree structured covariance matrices and tree metrics, giving a one-to-one relationship and implicitly showing the NP-hardness of optimizing over tree-structured covariance matrices with arbitrary topology.

3.1 Properties of tree-structured covariances and tree metrics

In the sequel, the following lemma will be useful

Lemma 1. *B_{ij} contains the weighted path length from the root to the nearest common ancestor of nodes i and j.*

Proof. Each column of V can be associated with a node in the tree. Each row of V contains a set of binary variables that are equal to 1 iff a corresponding node in the tree is on the path to the leaf associated with that row. As V is binary, $B_{ij} = V_{i:}DV_{j:}^T$ sums over those elements, m, of D for which $V_{im} = V_{jm} = 1$. These elements are exactly the lengths of the branches leading to the common ancestors of nodes i and j. □

Definition 3 (Four point condition). *A metric M satisfies the four point condition if the following holds*

$$M(a,b) + M(c,d) \leq \max(M(a,c) + M(b,d), M(a,d) + M(b,c)) \quad \forall a,b,c,d \tag{5}$$

Theorem 1 (Equivalence of the partition property and the 4 point condition). *The following statements are equivalent*

1. *M is a tree metric.*
2. *M satisfies the four point condition.*
3. *$M(i,j) = B_{ii} + B_{jj} - 2B_{ij}$ where $B = VDV^T$ is a tree-structured covariance matrix.*

Proof. 1 $\iff$ 2 is shown in [25].

3 $\implies$ 1: Using Lemma 1, $M(i,j)$ is the length of the path from the root to node i (B_{ii}) plus the length of the path from the root to node j (B_{jj}) minus two times the length of the path to the nearest common ancestor of nodes i and

j (B_{ij}). $B_{ii} - B_{ij}$ is therefore the length from node i to the nearest common ancestor of i and j, and $B_{jj} - B_{ij}$ is the length from node j to their nearest common ancestor. $M(i,j)$ is simply the sum of the two subpaths.

$1 \implies 3$ is a consequence of [24, Theorem 2]. □

We note that [25] considered unrooted trees while Definition 1 and Lemma 1 makes use of the root of a tree. This can be rectified by choosing a root arbitrarily in an unrooted tree (Figure 1). Such a choice corresponds to a degree of freedom in the construction of B that is customarily eliminated by data centering, or by working in a canonical basis as in Definition 1. This is formalized in Theorem 2.

Theorem 2 (Centering trees with different roots but identical topology). *Trees with different roots but identical topology project to the same covariance matrix when centered:*

$$H_k B_1 H_k = H_k B_2 H_k, \tag{6}$$

where B_1 and B_2 have identical topology and edge weights, but different roots, and $H_k = I - \frac{1}{k} e_k e_k^T$ is a centering matrix, e_k being the length k vector of all ones.

Proof. We first note that the linear operator defined in part 3 of Theorem 1, $B_{ii} + B_{jj} - 2B_{ij}$, projects to the same metric all tree structured covariance matrices with identical topology and edge weights, but potentially different roots. This is clear as $M(i,j)$ is simply the sum of weights along the unique path from node i to node j. Consequently, this operator applied to $B_1 - B_2$ yields the zero matrix, yielding a system of linear equations describing the null space of the operator. The null space can be summarized in compact matrix notation as follows

$$C e_k e_k^T + e_k e_k^T C \tag{7}$$

where C is an arbitrary diagonal matrix. We can consequently write any matrix with a fixed topology and edge weights as the summation of the component that lies in the null space of the operator, and the component that is orthogonal to the null space

$$B_1 = B_\perp + C_1 e_k e_k^T + e_k e_k^T C_1, \tag{8}$$

where $B_\perp$ is the component that is orthogonal to the null space, and is identical for all matrices with the same tree topology and edge weights.

We have that $H_k e_k e_k^T = e_k e_k^T H_k = \mathbf{0}$, which yields $H_k(C e_k e_k^T + e_k e_k^T C) H_k = \mathbf{0}$. This in turn implies that

$$\begin{aligned} H_k(B_1 - B_2)H_k =& H_k(B_\perp + C_1 e_k e_k^T + e_k e_k^T C_1 - \\ & B_\perp - C_2 e_k e_k^T - e_k e_k^T C_2) H_k = \mathbf{0} \end{aligned} \tag{9}$$

$$H_k B_1 H_k = H_k B_2 H_k. \tag{10}$$

□

4 Structured prediction with tree-structured covariances

Given the concepts developed in Section 3, we find now that the specification of joint feature maps and loss functions for taxonomic prediction is much simplified. We may assume that a taxonomy is specified that encodes the loss function Δ for a given problem, which need not be the same as a taxonomy for specifying the feature map ϕ. For the minimal path distance, $\Delta(y, \tilde{y}) = M(y, \tilde{y})$ for M defined as in Theorem 1. For Δ equal to the distance to the nearest common ancestor, we may use $B_{\tilde{y}\tilde{y}} - B_{y\tilde{y}}$. We have used the minimal path distance in the experimental section whenever taxonomic loss has been employed. The standard taxonomic structured loss functions therefore only require as an input a tree-structured covariance matrix B_{loss}, which need not be the same matrix as the one used to define a feature map (0-1 loss is recovered by using the identity matrix).

We now turn to the tree-structured joint kernel map (cf. Section 2). Given a tree-structured covariance matrix B and its decomposition into $B = VDV^T$, we may compactly define $\phi_y : \mathcal{Y} \mapsto \mathbb{R}^{2k-1}$ as the function that selects the kth column of $D^{\frac{1}{2}}V^T$ when y specifies that the sample belongs to the kth class.[5] Making use of the representer theorem for structured prediction with joint kernel maps [26], we know that the solution to our structured prediction objective lies in the span of our training input data $X \subset \mathcal{X}$ crossed with the output space, $\mathcal{Y}$. Assuming a kernel matrix K_x with associated reproducing kernel Hilbert space $\mathcal{F}$ such that the i, jth entry of K_x corresponds to $\langle \phi_x(x_i), \phi_x(x_j) \rangle_{\mathcal{F}}$, we have that the solution may be written

$$\sum_{1 \leq i \leq n} \sum_{y \in \mathcal{Y}} \alpha_{iy} \phi(x_i, y) \tag{11}$$

and that the corresponding joint kernel matrix decomposes as $K_x \otimes B$. Although the size of the joint kernel matrix is $n \cdot k \times n \cdot k$, we may make use of several properties of the Kronecker product to avoid high memory storage and costly matrix operations.

Looking specifically at Tikhonov regularized risk:

$$\min_g \lambda \|g\|_{\mathcal{H}}^2 + \ell(g, \mathcal{S}) = \min_\alpha \lambda \alpha^T (K_x \otimes B)\alpha + \ell(\alpha, \mathcal{S}) \tag{12}$$

where ℓ is some loss function (we have overloaded the notation in the kernelized case). Interestingly, we may use the identity from Theorem 2.3 of [27]

$$\alpha^T (K_x \otimes B)\alpha = \text{Tr}[K_x \tilde{\alpha}^T B \tilde{\alpha}] \tag{13}$$

where $\tilde{\alpha} \in \mathbb{R}^{n \times k}$ is the matrix such that $\text{vec}\, \tilde{\alpha} = \alpha$.

In the case of a structured output SVM, where we have a quadratic regularizer with linear constraints, we can make use of many optimization schemes, that, e.g. require repeated efficient multiplication of a vector with the Hessian:

$$(K_x \otimes B)\alpha = \text{vec}\, B\tilde{\alpha}K_x. \tag{14}$$

[5] A rooted tree with k leaves can be encoded with at most $2k - 1$ nodes (Figure 1).

Using the popular SVMstruct framework [19, 20] in this case generates a large number of non-sparse constraints and is very memory inefficient, requiring the storage of a number of kernel values proportional to the number of tuples in $\mathcal{X} \times \mathcal{Y} \times \mathcal{X} \times \mathcal{Y}$.[6] This indicates that the resulting memory requirements for such a scheme are $\mathcal{O}(n^2k^2)$, while making use of optimization with Equation (14) requires only $\mathcal{O}(n^2 + k^2 + nk)$ memory, and standard large scale kernel learning methods may be applied off-the-shelf to reduce the dominating $\mathcal{O}(n^2)$ component [28]. We have used a cutting plane training to efficiently train our taxonomic predictors, giving the same convergence guarantees as SVMstruct, but with substantially less expensive computation for cutting plane inference.

Cutting plane optimization requires finding a setting of $\tilde{y}$ that minimizes the right hand side of Equation (2). In the kernelized setting, we substitute for w as in Equation (12), and search for parameters $\beta \in \mathbb{R}^{nk \times 1}$ and $\delta \in \mathbb{R}$ that give the kernel coefficients and offset of the linear constraint

$$\delta - \alpha^T (K_x \otimes B)\beta \geq \xi. \tag{15}$$

Using Equation (14) enables us to solve this cutting plane iteration efficiently, both in terms of computation and memory usage. A reference implementation of this efficient optimization scheme is available for download from `https://github.com/blaschko/tree-structured-covariance`.

In the next section, we discuss how to learn taxonomies from data that are suitable for learning in this structured prediction model.

5 Optimizing tree structure covariances with the Hilbert-Schmidt Independence Criterion

In this section, we show how a non-parametric dependence test may be employed to learn taxonomies that can then be employed in the construction of a joint feature map for taxonomic prediction.

The Hilbert-Schmidt Independence Criterion (HSIC) is a kernel statistical measure that may be used to measure the dependence between empirical data observations and matrices that encode the hypothesized taxonomic structure of a data set [3]. The HSIC is defined to be the Hilbert-Schmidt norm of the cross covariance operator C_{xy} between mappings from the input space $\mathcal{X}$ and from the label space $\mathcal{Y}$. For characteristic kernels [29],[7] this is zero if and only if X and Y are independent. Given a finite sample of size n from $\Pr_{X,Y}$, the HSIC is

$$HSIC := \mathrm{Tr}[H_n K H_n L] \tag{16}$$

where K is the Gram matrix for samples from $\Pr_X$ with (i,j)th entry $k(x_i, x_j)$, and L is the Gram matrix with kernel $l(y_i, y_j)$.

[6] This follows from an analogous argument to the one used in binary classification that the storage requirements of a SVM are proportional to the Bayes rate, and therefore linear in the number of i.i.d. training samples.

[7] e.g. the Gaussian Kernel on $\mathbb{R}^d$.

To define our kernel matrix on the output space, we consider a family of functions proposed several times in the literature in the context of HSIC [3, 30]. In particular, we define the kernel in terms of a label matrix $\Pi \in \{0,1\}^{k \times n}$, and a covariance matrix, $B \in \mathbb{R}^{k \times k}$, that encodes the relationship between classes. Given these matrices, $L = \Pi^T B \Pi$. The HSIC with this kernel over $\mathcal{Y}$ is

$$HSIC_{\text{cov}} := \text{Tr}[H_n K H_n \Pi^T B \Pi]. \tag{17}$$

As pointed out by [31], $H_k \Pi H_n = \Pi H_n$, which in conjunction with Theorem 2 indicates that $HSIC_{\text{cov}}$ is identical regardless of how the tree is rooted (cf. Figure 1). We note that L is characteristic over $\mathcal{Y}$ whenever $\text{rank}[B] \geq k-1$ and the null space of B is empty or contains e_k.

When K_x is centered, the functional form of Equation (13) is identical to Equation (17), indicating that the regularizer is $HSIC_{\text{cov}}$ with $\tilde{\alpha}$ in place of Π. While our derivation has focused on tree-structured covariance matrices, this novel theoretical result is applicable to arbitrary covariances over $\mathcal{Y}$, indicating a tight coupling between non-parametric dependence tests and regularization in structured prediction.

With this fundamental relationship in place, we consider in turn optimizing over tree structured covariance matrices with fixed and arbitrary topology. The learned taxonomies may then be employed in structured prediction.

5.1 Optimization over tree-structured covariance matrices

Theorem 2 gives a convenient decomposition of a tree structured covariance matrix into a binary matrix encoding the topology of the tree and a positive diagonal matrix encoding the branch lengths. One such consequence of the existence of this decomposition is

Theorem 3. *The set of trees with identical topology is a convex set.*

Proof. [24] Given two tree structured covariance matrices with the same topology, $B = VDV^T$ and $\tilde{B} = V\tilde{D}V^T$, any convex combination can be written

$$\eta B + (1-\eta)\tilde{B} = V\left(\eta D + (1-\eta)\tilde{D}\right)V^T \tag{18}$$

for arbitrary $0 \leq \eta \leq 1$. □

Optimization of such covariance matrices with fixed topology is consequently significantly simplified. For D^* maximizing the HSIC subject to a norm constraint, a closed form solution is given by

$$D^* \propto \text{diag}\left[V^T \Pi^T H_n K_x H_n \Pi V\right]. \tag{19}$$

We note that this optimization is analogous to that in [3] for tree structured covariance matrices with arbitrary topology. In that work, a closed form solution for arbitrary positive definite matrices was found, which was later projected onto the space of tree-structured matrices using a *numerical taxonomy* algorithm with

tight approximation bounds. We have employed the method of [3] for comparison in the experimental results section. Theorems 1 and 2 justify the equivalence of our procedures for learning tree-structured covariance matrices with both fixed and arbitrary covariance matrices.

6 Experimental results

We perform an empirical study on two popular computer vision datasets, PASCAL VOC [16] and Oxford Flowers [17], and on the WIPO text dataset [18].

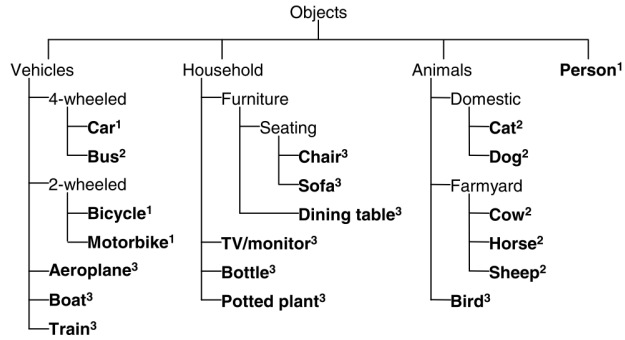
6.1 PASCAL VOC

We evaluate the performance of semantic vs. visual taxonomies on the PASCAL VOC 2007 dataset. To construct features for this data, we have employed results from the best performing submission to the 2007 classification challenge, `INRIA_Genetic`, which won all but one category. Our feature vector is constructed by concatenating variance normalized class prediction scores, after which a Gaussian kernel is applied, setting the σ parameter to the median of the pairwise distances in the feature space. As the parameters of the prediction functions were trained on data separate from the test images, this is a proper kernel over the test data set.[8] By construction, we are certain that the relevant visual information is contained within this feature representation, indicating that it is appropriate to use it to optimize the taxonomic structure. Furthermore, the `INRIA_Genetic` method did not make use of taxonomic relationships, meaning that no imputed class relationships will influence the taxonomy discovery algorithm.

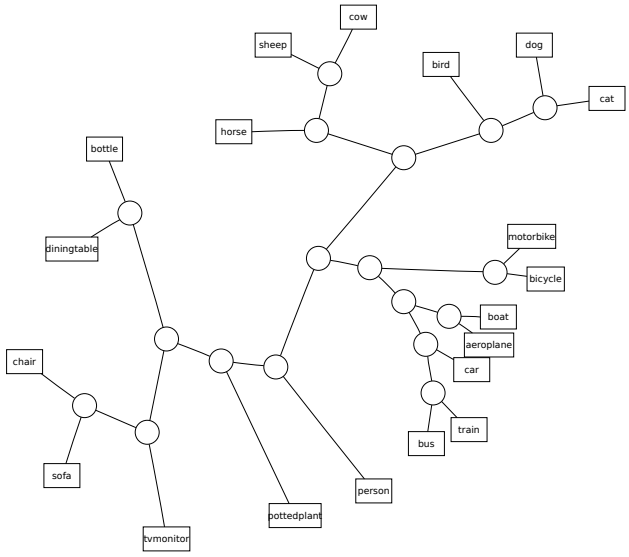
The semantic taxonomy was transcribed from the one proposed by the competition organizers [16]. As they do not provide edge lengths for their taxonomy (i.e. relative similarities for each subclass), we have learned these optimally from data using Equation (19). We have also learned a taxonomy with unconstrained topology, which is presented in Figure 2. Interestingly, the semantic topology and the learned topology are very close despite the learning algorithm's not having access to any information about the topology of the semantic taxonomy.

We have performed classification on the PASCAL VOC data set using the taxonomic prediction method described in Section 2. We trained on the first 50% of the competition test set, and report results as ROC curves on the second 50%. We emphasize that the results are designed for comparison between semantic and learned visual taxonomies, and are not for comparison within the competition framework. We additionally compare to the multi-class prediction method proposed by [32]. Results are shown in Figure 3.

[8] The learned taxonomy is available for download from `https://github.com/blaschko/tree-structured-covariance`. We note that this taxonomy is not appropriate to apply to the VOC 2007 dataset as that would involve training on the test set. However, as subsequent years of the VOC challenge use a disjoint set of images but the same classes, the taxonomy is applicable in those settings.



(a) Semantic taxonomy from [16].

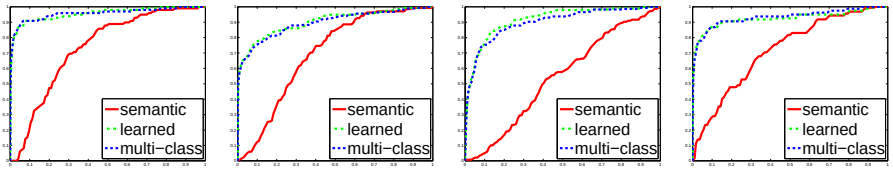


(b) Learned visual taxonomy.

Fig. 2. The semantic and learned taxonomies for the PASCAL VOC dataset. The semantic and visual taxonomies are very close, despite that the construction of the visual taxonomy made no use of the semantic relationships.

6.2 Oxford Flowers

In the second set of experiments, we have compared semantic to visual taxonomies on the Oxford Flowers data set. To construct a rich image representation, we have made use of the features designed by the authors of the dataset. The image representations consist of information encoding color, shape, (local) gradient histograms, and texture descriptors [17]. These features have resulted in high performance on this task in benchmark studies. We have constructed kernel matrices using the mean of Gaussian kernels as described in [33].



(a) aeroplane (b) bicycle (c) bird (d) boat

Fig. 3. ROC curves for the PASCAL VOC dataset. The learned visual taxonomy performs consistently better than the semantic taxonomy. Multi-class classification was performed with a multi-label generalization of [32]. Only the first four classes are shown due to space constraints. The other classes show qualitatively the same relationship between methods.

Table 1. Classification scores for the Oxford Flowers data set. The semantic taxonomy (Figure 4(a)) gives comparatively poor performance, likely due to the strong mismatch between the biological taxonomy and visual similarity. The learned visual taxonomy (Figure 4(b)), however, maintains good performance compared with one-vs.-rest classification.

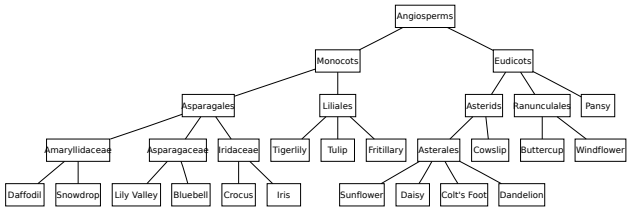
One vs. rest [33]	Semantic Taxonomy	Learned Taxonomy
84.9 ± 1.9	56.3 ± 6.3	$\mathbf{87.7 \pm 2.6}$

The topology of the semantic taxonomy was constructed using the Linnaean biological taxonomy, while edge distances were computed by optimizing D according to Equation (19). The topologies of the semantic taxonomy and the learned visual taxonomy are given in Figure 4.

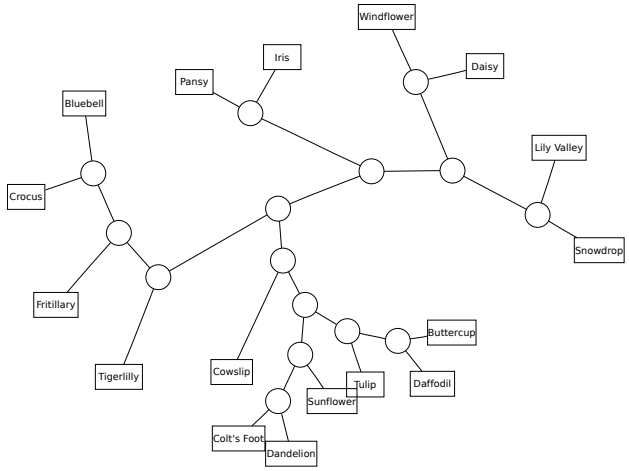
We have additionally performed classification using the semantic and learned visual taxonomies. We have applied the taxonomic prediction method described in Section 2. The results are presented in Table 1. In line with previous results on taxonomic prediction, the performance of the taxonomic method with a visual taxonomy performs comparably to 1-vs.-rest classification (here we report the results from [33], which use an identical kernel matrix to our method). However, we note that the semantic taxonomy performs very poorly, while the learned taxonomy maintains good results. We hypothesize that this is due to the strong mismatch between the semantic relationships and the visual ones. In this case, it is inappropriate to make use of a semantic taxonomy, but our approach enables us to gain the benefits of taxonomic prediction without requiring an additional information source to construct the taxonomy.

6.3 Text categorization

We present timing and accuracies on the WIPO data set [18], a hierarchically structured document corpus that is commonly used in taxonomic prediction [21]. Kernel design was performed simply using a bag of words feature representation combined with a generalized Gaussian χ^2 kernel with the bandwidth parameter



(a) Semantic taxonomy constructed using biological information.



(b) Learned taxonomy.

Fig. 4. Semantic and visual taxonomies on the Oxford Flowers dataset. The topologies of the two taxonomies differ significantly, indicating a strong mismatch between the semantic hierarchy and visual similarity.

set to the median of the pairwise χ^2 distances. The topology, V, of the tree structure was constructed using the taxonomy provided by the data set organizers. The loss function, Δ, was either set to 0-1 loss, or the taxonomic distance between two concepts. The taxonomic distance between two concepts was measured as the unweighted path length between the two leaves in the taxonomy (i.e. not making use of the learned taxonomy but instead fixing edge lengths to 1).

We have computed results using a number of covariance structures, as well as a number of loss functions. Table 2 lists these settings and shows their numerical accuracies. We emphasize that the results correspond to the learning setting proposed by [21] when the covariance matrix is tree-structured. Any differences in performance for this column are due to our using a more recent version of the data set with a comparatively naïve feature representation, while Cai and Hofmann made use of an unspecified kernel function computed using a proprietary software system [21].

Table 2. Losses on the WIPO data set (lower is better). The columns correspond to varying covariance structures, while the rows correspond to different loss functions. For the covariance structures, I corresponds to a standard multi-class feature map [32], B^* is learned using the method proposed in [3] for learning taxonomies without fixed topology, and D^* is learned from Equation (19). Each system was trained with a structured output support vector machine optimizing the loss on which it is evaluated.

	I	B^*	$H_k V D^* V^T H_k$	$V D^* V^T$
0-1	0.281 ± 0.027	$\mathbf{0.278 \pm 0.042}$	0.284 ± 0.037	0.362 ± 0.028
taxonomic	0.950 ± 0.100	$\mathbf{0.833 \pm 0.179}$	1.125 ± 0.071	1.120 ± 0.028

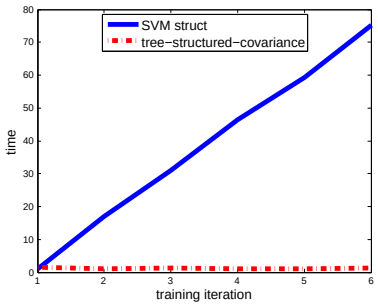


Fig. 5. Computation time for constraint generation using the proposed method of optimization vs. the popular SVMstruct optimization package [19, 20]. The proposed optimization is several orders of magnitude faster than SVMstruct for this problem, and has constant computation time per iteration, while SVMstruct has computation that grows linearly with the training iteration.

We focus on the efficiency of the optimization using our strategy, and the kernelized variant of SVMstruct [19, 20]. We compare the empirical time per cutting plane iteration in Figure 5. We note that timing results are presented as a fraction of the first training iteration to account for differences in vector and matrix libraries employed in our implementation vs. SVMstruct. Nevertheless, our implementation was several orders of magnitude faster than SVMstruct at all iterations of training due to the avoidance of naïve looping over cached kernel values as employed by their general purpose framework. In the SVMstruct implementation of taxonomic prediction, the joint kernel function was implemented by multiplying K_{ij} by $B_{y_i y_j}$, which were both kept in memory to optimize computation time. The computation time of our algorithm is constant per iteration, in contrast to SVMstruct, which grows approximately linearly with high slope as the number of support vectors grows. In later training iterations, a single kernelized cutting plane iteration of SVMstruct can take several minutes, while our method takes only several milliseconds. The number of cutting plane iterations required by both methods is identical.

7 Conclusions

In this work, we have compared taxonomies learned from data with semantic taxonomies provided by domain experts, where these taxonomies are used to impose structure in learning problems. While a semantic taxonomy provides a measure of prior information on class relationships, this may be unhelpful to the desired learning outcome when the features available are not in accord with this structure. Indeed, in such cases, we have shown that the imposition of prior taxonomic information may result in a significant performance penalty.

By contrast, we have observed that learned taxonomies based on feature similarity can do significantly better than hand-designed taxonomies, while never performing significantly worse than alternatives. Moreover, we have shown that the taxonomic structure may be encoded in a tree-structured covariance: as a result, we were able to develop a highly computationally efficient learning algorithm over taxonomies. Software and machine readable tree-structured covariance matrices are available for download from `https://github.com/blaschko/tree-structured-covariance`.

Acknowledgements
This work is partially funded by the European Research Council under FP7/ERC Grant 259112, and by the Royal Academy of Engineering through the Newton alumni scheme.

References

1. Zweig, A., Weinshall, D.: Exploiting object hierarchy: Combining models from different category levels. In: ICCV. (2007)
2. Binder, A., Müller, K.R., Kawanabe, M.: On taxonomies for multi-class image categorization. IJCV (2012)
3. Blaschko, M.B., Gretton, A.: Learning taxonomies by dependence maximization. In: NIPS. (2009)
4. Lampert, C.H., Blaschko, M.B.: A multiple kernel learning approach to joint multi-class object detection. In Rigoll, G., ed.: Pattern Recognition: Proceedings of the 30th DAGM Symposium. (2008) 31–40
5. Tibshirani, R., Hastie, T.: Margin trees for high-dimensional classification. JMLR **8** (2007) 637–652
6. Fan, X.: Efficient multiclass object detection by a hierarchy of classifiers. In: CVPR. (2005)
7. Griffin, G., Perona, P.: Learning and using taxonomies for fast visual categorization. In: CVPR. (2008)
8. Marszałek, M., Schmid, C.: Semantic hierarchies for visual object recognition. In: CVPR. (2007)
9. Marszałek, M., Schmid, C.: Constructing category hierarchies for visual recognition. In: ECCV. (2008)
10. Zhao, B., Li, F.F.F., Xing, E.P.: Large-scale category structure aware image categorization. In: NIPS. (2011) 1251–1259

11. Mittal, A., Blaschko, M.B., Zisserman, A., Torr, P.H.S.: Taxonomic multi-class prediction and person layout using efficient structured ranking. In: ECCV. (2012) 245–258
12. McAuley, J., Ramisa, A., Caetano, T.: Optimization of robust loss functions for weakly-labeled image taxonomies. IJCV (2012) 1–19
13. Weinberger, K., Chapelle, O.: Large margin taxonomy embedding for document categorization. In: NIPS. (2009) 1737–1744
14. Bengio, S., Weston, J., Grangier, D.: Label embedding trees for large multi-class tasks. In: NIPS. (2010) 163–171
15. Gao, T., Koller, D.: Discriminative learning of relaxed hierarchy for large-scale visual recognition. In: ICCV. (2011) 2072–2079
16. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes (VOC) challenge. IJCV **88**(2) (2010) 303–338
17. Nilsback, M.E., Zisserman, A.: Delving deeper into the whorl of flower segmentation. Image and Vision Computing (2009)
18. World Intellectual Property Organization: WIPO-alpha data set, `http://www.wipo.int/` (2009)
19. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: ICML. (2004)
20. Joachims, T., Finley, T., Yu, C.N.J.: Cutting-plane training of structural SVMs. Mach. Learn. **77**(1) (2009) 27–59
21. Cai, L., Hofmann, T.: Hierarchical document categorization with support vector machines. In: CIKM. (2004)
22. Wang, K., Zhou, S., Liew, S.C.: Building hierarchical classifiers using class proximity. In: VLDB. (1999)
23. Cavalli-Sforza, L.L., Edwards, A.W.F.: Phylogenetic analysis: Models and estimation procedures. American Journal of Human Genetics **19** (1967) 223–257
24. Corrada Bravo, H., Wright, S., Eng, K., Keleş, S., Wahba, G.: Estimating tree-structured covariance matrices via mixed-integer programming. In: AISTATS. (2009)
25. Buneman, P.: The recovery of trees from measures of dissimilarity. In Kendall, D.G., Tautu, P., eds.: Mathematics in the Archeological and Historical Sciences. Edinburgh University Press (1971) 387–395
26. Lafferty, J., Zhu, X., Liu, Y.: Kernel conditional random fields: representation and clique selection. In: ICML. (2004)
27. Magnus, J.R., Neudecker, H.: Matrix Differential Calculus with Applications in Statistics and Econometrics. Wiley (1988)
28. Bottou, L., Chapelle, O., DeCoste, D., Weston, J.: Large-Scale Kernel Machines. MIT Press (2007)
29. Fukumizu, K., Gretton, A., Sun, X., Schölkopf, B.: Kernel measures of conditional dependence. In: NIPS. (2008) 489–496
30. Song, L., Smola, A., Gretton, A., Borgwardt, K.M.: A dependence maximization view of clustering. In: ICML. (2007)
31. Blaschko, M.B., Gretton, A.: Taxonomy inference using kernel dependence measures. Technical Report 181, Max Planck Inst. for Bio. Cybernetics (2008)
32. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. JMLR **2** (2002) 265–292
33. Gehler, P., Nowozin, S.: On feature combination methods for multiclass object classification. In: ICCV. (2009)