
Semi-Supervised Learning with Trees

C. C. Kemp, T. L. Griffiths, S. Stromsten & J. B. Tenenbaum

Department of Brain and Cognitive Sciences, MIT, Cambridge, MA 02139
{ckemp,gruffydd,sean.s,jbt}@mit.edu

Abstract

We describe a nonparametric Bayesian approach to generalizing from few labeled examples, guided by a larger set of unlabeled objects and the assumption of a latent tree-structure to the domain. The tree (or a distribution over trees) may be inferred using the unlabeled data. A prior over concepts generated by a mutation process on the inferred tree(s) allows efficient computation of the optimal Bayesian classification function from the labeled examples. Our approach performs well across six real-world datasets, and extends naturally to handle two difficult problems: learning from very sparse data, and learning from positive examples only.

1 Introduction

People have remarkable abilities to learn concepts from very limited data, often just one or a few labeled examples per class. *Semi-supervised learning* approaches in machine learning try to match this ability, by extracting strong inductive biases from a much larger sample of *unlabeled* data. A general strategy is to adopt some metric-space model \mathcal{M} that underlies both the observed features of the unlabeled data and the new concept C to be learned. The unlabeled data can be used to identify the model \mathcal{M} , and an assumption that C is somehow “smooth” with respect to \mathcal{M} – or in Bayesian terms, can be assigned a strong prior conditional on \mathcal{M} – provides the inductive bias needed to generalize C successfully from very few labeled examples.

Existing approaches make different assumptions within this framework. Transductive Support Vector Machines [1] take the metric space to be a high-dimensional feature space defined by the kernel, and by maximizing the margin of C on the unlabeled data, effectively assume that C is maximally smooth with respect to the density of unlabeled data in feature space. The Laplacian method of Belkin and Niyogi assumes the data concentrate on a low-dimensional (Riemannian) manifold, and that C is smooth on that manifold [2]. The manifold geometry is estimated from the unlabeled data using simple bottom-up techniques based on a graph of nearest neighbors. This approach performs quite well in classifying data with a natural manifold structure, e.g., handwritten digits. Methods of [3] and [4] combine aspects of both these approaches.

Many domains are more structured than generic kernel-based approaches assume, yet do not have an underlying low-dimensional Riemannian geometry. We would like to design semi-supervised approaches that can exploit other kinds of metric structures. In particular, trees arise prominently in both natural and human-generated domains (e.g., in biology, language and information retrieval). Here we propose an approach to semi-supervised learning based on embedding the data in an ultrametric space – a rooted tree \mathcal{T} with the data located at leaf nodes equidistant from the root. The concept C is generated from a stochastic mutation process oper-

ating over branches of \mathcal{T} . The tree \mathcal{T} can be inferred from unlabeled data using either bottom-up methods (agglomerative clustering) or more complex probabilistic methods. The mutation process defines a prior over all possible (transductive) concepts – all possible labelings of the unlabeled data – favoring those that maximize a tree-specific notion of “smoothness”. Figure 1 illustrates this *Tree-Based Bayes (TBB)* approach.

TBB classifies unlabeled data by integrating over all $2^{|X|}$ hypotheses defined on the data set X , and is thus an instance of optimal Bayesian concept learning [5]. In general, optimal Bayes is of theoretical interest only [6], because the sum over hypotheses is intractable and it is difficult to specify sufficiently powerful and noise-resistant priors for real-world domains. Here, by working in the semi-supervised setting and defining the prior in terms of a tree-based mutation process, the approach becomes efficient and empirically successful, even when the data are not strongly tree-structured.

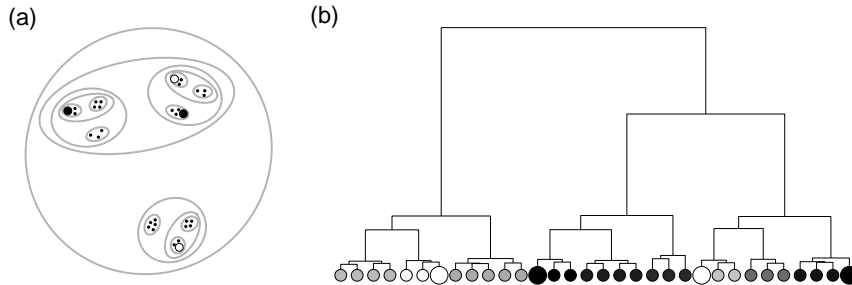


Figure 1: Schematic illustration of the Tree-Based Bayesian approach to semi-supervised learning. (a) We observe a set of unlabeled objects (small points) with some latent hierarchical structure (gray ellipses), along with two positive and two negative examples of a new concept (black and white circles). (b) Inferring the latent tree, and treating the concept as generated from a mutation process on the tree, we can classify the unlabeled objects.

The next section describes TBB, as well as a simple heuristic method, *Tree Nearest Neighbor (TNN)*, which we show approximates TBB in the limit of high mutation rate. Section 3 presents experiments comparing these algorithms with other approaches on a range of data sets. Section 4 illustrates how the Bayesian formulation extends to handle very impoverished forms of data, produced by single-class sampling (labeled data drawn from just one class) or sparse sampling (mostly missing feature values).

2 Tree-Based Bayes

We begin with a domain of objects $X = \{x_1, \dots, x_t\}$. Each object x_i is associated with a vector of observable features (also denoted x_i). We are given labels $Y_l = \{y_1, \dots, y_s\}$ for the first s objects ($s < t$), and our goal is to infer the labels $Y_u = \{y_{s+1}, \dots, y_t\}$ of the $t - s$ unlabeled points. We refer to $X_l = \{x_1, \dots, x_s\}$ as the “labeled data”, and the complement $X_u = X \setminus X_l$ as the “unlabeled data”. For now we assume a binary classification problem, with $Y = Y_l \cup Y_u \in \{-1, 1\}^{|X|}$; generalization to the multi-class is straightforward.

We also assume an ultrametric binary tree \mathcal{T} with the objects X for its leaves. Expressed in terms of the tree, the learning problem is to find an indicator function over its leaves. Our intuition is that, at least in some domains, there exists a tree \mathcal{T} such that natural concepts tend to assign the same label to most pairs of nearby leaves in \mathcal{T} . Our *Tree-Based Bayes (TBB)* approach grounds this “tree-proximity” principle in a generative model for object features in tree-structured domains. We first show how to define the ingredients of Bayesian concept learning in terms of a particular tree \mathcal{T} , extending previous work by ourselves and others [5, 6, 7, 8, 9]. We then consider the relevant case for semi-supervised learning, where \mathcal{T} is unknown

but may be inferred with the help of unlabeled data.

Let H be our hypothesis space. Our approach is non-parameteric, in that we take H to be the set of all possible binary labelings h of objects in our domain X : $h \in \{-1, 1\}^{|X|}$. We define $D = \{X_l, Y_l\}$ to denote the labeled examples, consisting of the labeled objects X_l and their labels Y_l . For each unlabeled object $x_i \in X_u$, we want to compute $p(y_i = 1|D, \mathcal{T})$, the probability that x_i is a positive instance of the concept given the examples and the tree \mathcal{T} . Summing over all hypotheses in H we have

$$p(y_i = 1|D, \mathcal{T}) = \sum_{h \in H} p(y_i = 1|h, D, \mathcal{T})p(h|D, \mathcal{T}). \quad (1)$$

Define $H(y_i^+)$ to be the subset of hypotheses that label x_i positive. Because $p(y_i = 1|h, D, \mathcal{T}) = 1$ if $h \in H(y_i^+)$, and 0 otherwise,

$$p(y_i = 1|D, \mathcal{T}) = \sum_{h \in H(y_i^+)} p(h|D, \mathcal{T}) \quad (2)$$

$$= \frac{1}{p(D|\mathcal{T})} \sum_{h \in H(y_i^+)} p(D|h, \mathcal{T})p(h|\mathcal{T}). \quad (3)$$

Let $H(Y_l)$ be the subset of hypotheses consistent with the labels Y_l (the version space). For all hypotheses $h \notin H(Y_l)$, the likelihood $p(D|h, \mathcal{T})$ equals zero. For hypotheses consistent with Y_l , $p(D|h, \mathcal{T}) = p(X_l|h, \mathcal{T})$. Expanding the denominator $p(D|\mathcal{T}) = \sum_h p(D|h, \mathcal{T})p(h|\mathcal{T})$, we can write Equation 3 as:

$$p(y_i = 1|D, \mathcal{T}) = \frac{\sum_{h \in H(Y_l) \cap H(y_i^+)} p(X_l|h, \mathcal{T})p(h|\mathcal{T})}{\sum_{h \in H(Y_l)} p(X_l|h, \mathcal{T})p(h|\mathcal{T})}. \quad (4)$$

The likelihood $p(X_l|h, \mathcal{T})$ depends on how the labeled set X_l was chosen, which is often unknown. A typical assumption is that X_l was drawn randomly from all objects in the domain X . Then $p(X_l|h, \mathcal{T})$ is independent of h and cancels from Equation 4, leaving:

$$p(y_i = 1|D, \mathcal{T}) = \frac{\sum_{h \in H(Y_l) \cap H(y_i^+)} p(h|\mathcal{T})}{\sum_{h \in H(Y_l)} p(h|\mathcal{T})}. \quad (5)$$

That is, the probability that x_i is a positive instance reduces to the fraction of hypotheses consistent with the examples that label x_i positively, as measured under the prior probability $p(h|\mathcal{T})$. Other sampling paradigms – and hence different likelihood functions $p(X_l|h, \mathcal{T})$ in Equation 4 – might be adopted when learning from small training sets. Consider identifying genetic markers for a disease that afflicts one person in 10,000. A training set for this problem might be constructed by “balanced sampling”, e.g., taking data from 20 patients with the disease and 20 healthy subjects. Randomly sampling subjects from the entire population would require using a huge training set to have a good chance of including anyone with the disease.

Although many real-world data sets are constructed in a manner closer to balanced sampling, our analysis is more tractable under random sampling (via Equation 5). Hence we assume random sampling for now, but our experiments will explore both paradigms, and in Section 4 we consider an alternative sampling paradigm (based on Equation 4) where the labeled data come from just a single class.

2.1 Bayesian classification with a mutation model

In many tree-structured domains – not only in biology, but in other fields as well – it is natural to think of entities as the result of some evolutionary process, and to think of features or concepts as arising from a history of stochastic events or mutations. Independent of its naturalness, a

mutation process over \mathcal{T} induces a sensible “smoothness” prior $p(h|\mathcal{T})$ and enables efficient computation of Equation 5, via belief propagation on a Bayes net.

Our mutation model combines aspects of several previous proposals for probabilistic learning with trees [8, 9, 10]. Let L be a feature corresponding to the class label. Suppose that L is defined at all nodes of \mathcal{T} , not just the leaves. We model the development of L as a Poisson arrival process with a parameter, λ , that will be called the mutation rate. The probability that the feature changes value along a branch b of length $|b|$ is the probability of an odd number of arrivals along that branch:

$$p(L \text{ changes along } b) = \frac{1 - e^{-2\lambda|b|}}{2}. \quad (6)$$

Note that mutations are assumed to be symmetric: a feature is just as likely to switch from 1 to 0 as from 0 to 1. The approach, however, can readily accommodate other probabilistic models of mutation.

The prior $p(h|\mathcal{T})$ is the probability of generating the indicator function h over leaves of \mathcal{T} under the mutation process. The resulting distribution favors labelings that are “smooth” with respect to \mathcal{T} . Regardless of λ , it is always more likely for L to stay the same than to switch its value along a branch (Equation 6 is always less than 1/2). Thus labelings that do not require very many mutations are preferred, and the two hypotheses that assign the same label to all leaf nodes receive the most weight. Because mutations are more likely to occur along longer branches, the prior also favors hypotheses in which label changes occur between clusters (where branches tend to be longer) rather than within clusters (where branches tend to be shorter).

The independence assumptions implicit in the mutation model allow the right side of Equation 5 to be computed efficiently. The value of L at any child in the tree depends only on the value of its parent and the number of mutations that occurred between parent and child. We can therefore set up a Bayes net with the same topology as \mathcal{T} that captures the joint probability distribution over all nodes. We associate with each branch a conditional probability table that specifies the value of the child conditioned on the value of the parent (based on Equation 6), and set the prior probabilities at the root node to the uniform distribution. Evaluating Equation 5 now reduces to a standard problem of inference in a Bayes net: computing the marginal probability of a group of evidence nodes (corresponding to the labeled examples D for the denominator, and the labeled examples plus x_i for the numerator). The tree structure makes this computation efficient (and allows specially tuned inference algorithms, as in [9]).

2.2 Tree Nearest Neighbor

A Bayesian formulation based on the mutation process provides a principled approach to learning with trees, but we can imagine simpler algorithms that instantiate similar intuitions. For instance, we could build a nearest-neighbour classifier using the metric of distance in the tree \mathcal{T} . It is clear how this *Tree Nearest Neighbor (TNN)* algorithm reflects the assumption that nearby leaves in \mathcal{T} are likely to have the same label, but it is not necessarily clear when and why this simple approach should work well. An analysis of Tree-Based Bayes provides some insight here. We can show that these two algorithms, TBB and TNN, become equivalent when the λ parameter of TBB is set sufficiently high.

Theorem 1 *For each ultrametric tree \mathcal{T} , there is a λ such that TNN and TBB produce identical classifications for all examples with a unique nearest neighbour.*

A full proof is available at <http://www.mit.edu/~ckemp/nips03.pdf>; we sketch the main steps here. Let the “ L -skeleton” of \mathcal{T} be the subtree consisting of all paths from the labeled leaf nodes to the root. Suppose that N_ℓ is any labeled node, and N_α is the most recent ancestor of N_ℓ with two labeled descendants. We establish the theorem by showing that every node in the L -skeleton between N_ℓ and N_α has a posterior distribution that favours y_ℓ once λ grows large. Assume that $q = p(N_\alpha \neq y_\ell | D) > 0.5$ (otherwise we are done), and

that the distance between N_ℓ and N_a is 1. Under the mutation model of the previous section, we can show that any skeleton node within d of N_ℓ has a posterior that favours y_ℓ , where $d = \frac{1}{2} + \frac{1}{4\lambda} \log(\frac{1}{2q-1})$. A worst case analysis then shows that $\lim_{\lambda \rightarrow \infty} d = 1$.

Given this equivalence, TNN is the superior algorithm when a high mutation rate is appropriate. It is not only faster, but numerically more stable. For large values of λ , the probabilities manipulated by TBB become very close to 0.5, and variables that should be different may become indistinguishable within the limits of computational precision. Our implementation of TBB therefore uses TNN when a sufficiently high value of λ is required.

2.3 Computing the tree

So far, in computing $p(y_i = 1|D, \mathcal{T})$, we have assumed that the tree \mathcal{T} is known. Generally in semi-supervised learning we will not know the best tree to use. We need to compute $p(y_i = 1|D, X_u)$, the classification of x_i conditioned on the labeled objects and their labels, $D = \{X_l, Y_l\}$, and the unlabeled data X_u . We can proceed using the full data X to infer the tree. In principle, this requires a sum over all trees on X :

$$p(y_i = 1|D, X_u) = \sum_{\mathcal{T}} p(y_i = 1|\mathcal{T}, D, X_u)p(\mathcal{T}|D, X_u). \quad (7)$$

As the sum over all trees is intractable, we consider two approaches to approximation. Markov Chain Monte Carlo (MCMC) techniques have been used to approximate similar sums over trees in Bayesian phylogenetics [11]. MCMC over trees can be quite costly, but as we show in Section 4, it does offer significant benefits when the data are very sparse. A simpler approach is to assume that most of the probability $p(\mathcal{T}|D, X_u)$ is concentrated on or near the most probable tree \mathcal{T}^* , and approximate Equation 7 as simply $p(y_i = 1|\mathcal{T}^*, D, X_u)$. This expression is just what we computed above in Equation 5 for a known tree \mathcal{T} (assuming the new concept is conditionally independent of the other features of objects, X_u , given \mathcal{T}). We can estimate \mathcal{T}^* through more or less sophisticated means. The experiments in Section 3 use a greedy bottom-up method (average-link agglomerative clustering on the full data X) that is comparable in complexity to the neighborhood graph constructions used in [2].

3 Experiments: tree-based versus manifold-based approaches

We compared four algorithms: TBB, TNN, the Laplacian approach of Belkin and Niyogi, and generic Nearest Neighbor. We used four data sets which we expected to have a tree-like structure, because they arose from biological taxonomies (Beetles, Crustaceans, Salamanders and Worms, with respective sizes $|X| = 192, 56, 30$ and 286). We also used two sets expected to have a low-dimensional manifold structure, because they arose from human motor behaviors (Digits and Vowels, with $|X| = 10,000$ and 990 , respectively).

Each “tree” set describes the external anatomy of a group of species, based on data available at <http://biodiversity.uno.edu/delta/>. One feature in the Beetles set, for example, indicates whether a beetle’s body is “strongly flattened, slightly flattened to moderately convex, or strongly convex.” These tree sets do not include class labels, so we chose features at random to stand in for the class label. We averaged across ten such choices for each dataset. Our Digits set is a subset of the MNIST data, and Vowels is taken from the UCI repository.

Our experiments focused on learning from very small labeled sets. The number of labeled examples was always set to a small multiple ($m = 1, 2, 3, 5$, or 10) of the total number of classes. The algorithms were compared under random and balanced sampling, and training sets were always sampled with replacement. For each training-set size m , we averaged across 20 random splits of the data into X_l and X_u . Free parameters for TBB (λ) and Laplacian (number of nearest neighbors, number of eigenvectors) were chosen using randomized leave-one-out cross-validation.

Figure 2 shows the performance of the algorithms under random sampling. TBB outperforms

the other algorithms across the four “tree” sets, but the difference between TBB and Nearest Neighbor is rather small. The results suggest a substantial advantage for TBB over Laplacian, in (presumably) tree-structured domains. As expected, this pattern is reversed on the Digits set, but it is encouraging that the tree-based methods can improve on Nearest Neighbour even for datasets that are not normally associated with trees. Surprisingly, TBB performs a little better than Laplacian on the vowel set, but neither method beats Nearest Neighbor.

Figure 3 shows that balanced sampling is better able to discriminate between the algorithms. There is a clear advantage here for TBB on the “tree” sets. Much more than the other algorithms, TBB copes well when the class proportions in the training set do not match the proportions in the population, and it turns out that many of the features in the taxonomic datasets are unbalanced. Since the Digits and Vowels sets have classes of approximately equal size, the results for balanced sampling are similar to those for random sampling.

While not conclusive, our results suggest that TBB may be the method of choice on tree-structured datasets, and is robust even for data that are not clearly tree-structured.

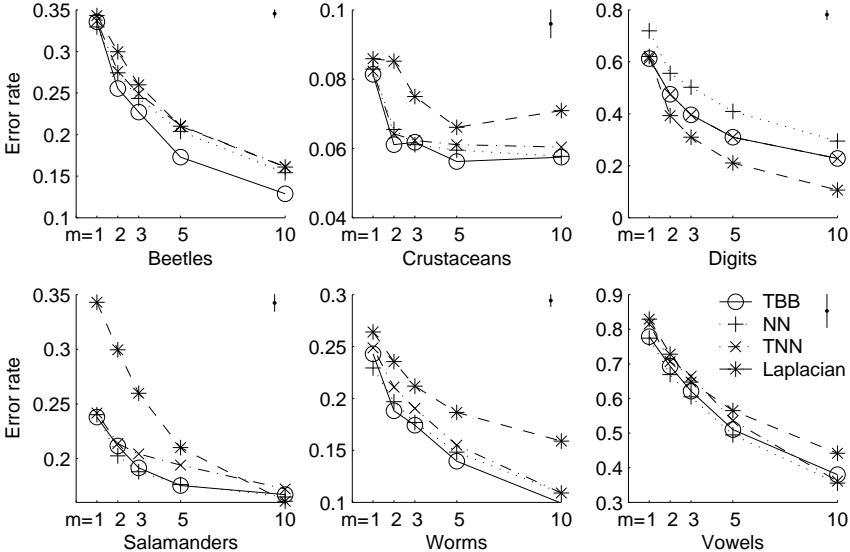


Figure 2: Error rates under random sampling for six datasets, described in the text. The total number of labeled examples is m multiplied by the number of classes. Mean standard error bars for each data set are shown in the upper right corner of the plot.

4 Conclusion and Extensions

We have shown how to make optimal Bayesian concept learning tractable in a semi-supervised setting, by assuming a latent tree structure that can be inferred from the unlabeled data and defining a prior for concepts based on a mutation process over the tree. The Bayesian framework supports a number of extensions to the basic semi-supervised paradigm. We close by briefly considering two extreme cases of impoverished data: learning with labeled examples drawn from a single class only, and learning from sparsely observed data.

Discriminative techniques like the Nearest Neighbour and Laplacian approaches do not naturally apply when the labeled data include only positive examples of a single class. Our Bayesian approach can be adapted to this setting by choosing $p(X_l|h, \mathcal{T})$ in Equation 4 appropriately. The restriction to positive examples means that $p(X_l|h, \mathcal{T})$ becomes proportional to $1/|h^+|^m$, where $|h^+|$ is the number of objects labeled positive by h , and m is the number of ex-

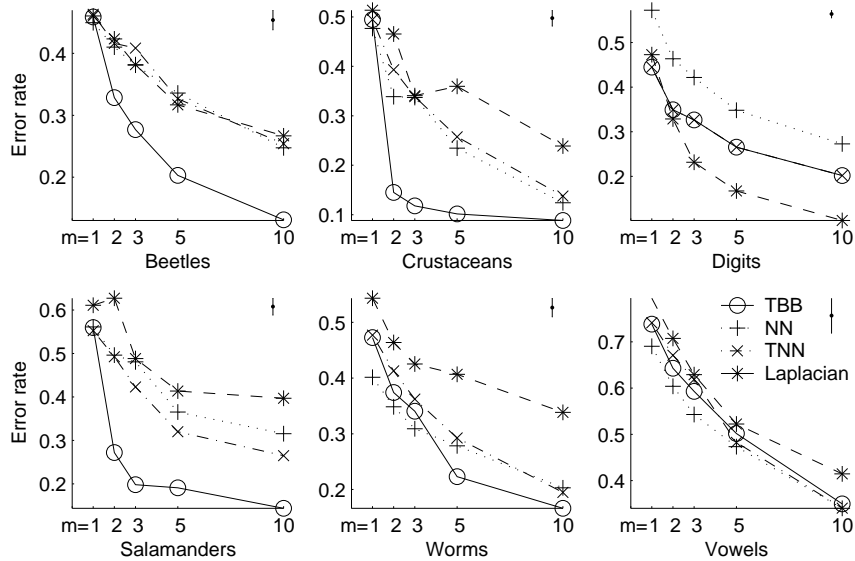


Figure 3: Results under balanced sampling, with m labeled examples from each class.

amples (assuming sampling with replacement). The classification probability $p(y_i = 1|D, \mathcal{T})$ can no longer be computed in closed form by belief propagation. We use a greedy search to find a small number of hypotheses with high posterior, by positing minimal mutations along an extended skeleton of the labeled examples: “on” mutations along paths in \mathcal{T} from examples to the root, and “off” mutations along offshoots of those paths. We then approximate Equation 4 by summing over just these hypotheses. Figure 4a shows the extended skeleton for one concept in the UCI Zoo data set. Figure 4b shows a comparison between this approach and MIN, a simpler algorithm which always chooses the single smallest subtree consistent with the positive examples. When the concept to be learned corresponds to a single subtree, both methods do well, but when it does not (as in Figure 4a), MIN can dramatically overgeneralize. TBB always converges rapidly on the true concept.

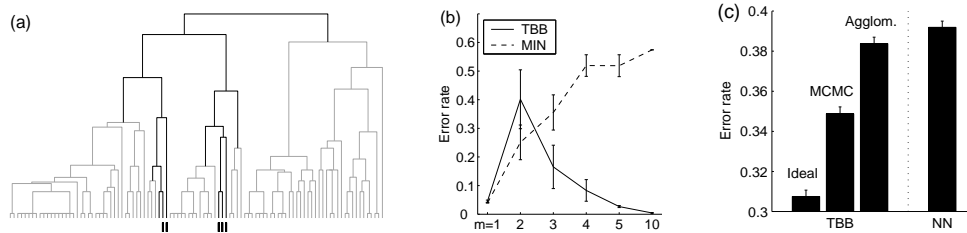


Figure 4: (a) All nodes dominating any of the five members of a single class from the Zoo dataset, used in approximating Equation 4. (b) Performance of the MIN and hypothesis-averaging algorithms given m labeled examples of this class. (c) Results of inferring the tree for TBB from extremely sparse data with three different methods, compared with NN.

In many tree-structured domains of interest (e.g., in biology), there may not be sufficient data X to allow a single good tree structure \mathcal{T} to be discovered using simple greedy clustering. We can still exploit the tree structure by using MCMC techniques to approximate the sum over trees in Equation 7. For objects with binary or discrete features, we define a generative distribution over the features of X , $p(X|\mathcal{T})$, using the same mutation process that generated our prior over concepts in Section 2.1. We take the distribution over trees $p(\mathcal{T}|D, X_u)$ to be

proportional to $p(X|\mathcal{T})$, effectively assuming a uniform prior on \mathcal{T} and ignoring the negligible contribution of the labels Y .

To test MCMC under conditions where \mathcal{T} is hard to identify, we generated artificial data sets consisting of $t = 20$ objects. Each data set was based on a “true” tree \mathcal{T}_0 , with objects x_i at leaves of \mathcal{T}_0 . Each object x_i was represented by a vector of 20 binary features generated from the mutation process over \mathcal{T}_0 , with high λ . Moreover, most feature values were missing; on average, the algorithms saw only 5 of the 20 features for each object. For each data set, we created 20 test concepts from the same mutation process. The algorithm saw 4 labeled examples of each test concept and had to infer the labels of the other 16 objects. This experiment was repeated for 10 random trees \mathcal{T}_0 . Our MCMC approach was inspired by an algorithm for reconstruction of phylogenetic trees [11], which uses Metropolis-Hastings over tree topologies with two kinds of proposals: local (nearest neighbor interchange) and global (subtree pruning and regrafting). Figure 4c shows the mean classification error rate, based on 1600 samples after a burn-in of 400 iterations. Figure 4c also shows results from Nearest Neighbor and two versions of TBB that used a single tree: “agglom”, using the tree constructed by agglomerative clustering over X (ignoring missing features), and “ideal”, using the true tree \mathcal{T}_0 . The ideal learner beats all others, because the true tree is impossible to identify with such sparse data. Using MCMC over trees brings TBB substantially closer to the ideal than simpler alternatives that ignore the tree structure (NN) or consider only a single tree (“agglom”).

Advocates of Bayesian approaches to traditional supervised classification tasks have often emphasized how the framework supports useful auxiliary inferences: e.g., active learning (selecting the unlabeled points whose labels would be most informative), and estimating confidence on the inferred class labels. Both of these issues are most pressing when learning from few labeled examples in the semi-supervised setting, and we expect them to be handled naturally by further extensions to our Bayesian approach. More generally, it would be valuable to formulate Bayesian approaches to semi-supervised learning for other kinds of metric-space structures, including (but not limited to) Riemannian manifolds. The necessary computations will almost surely be less tractable, but even approximate techniques could be useful. When faced with a new domain of data where the form of the underlying structure is unknown, Bayesian methods for model selection could then be used (given sufficient unlabeled data) to choose among approaches that assume trees, manifolds, or other canonical representational forms.

References

- [1] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.
- [2] M. Belkin and P. Niyogi. Semi-supervised learning on Riemannian manifolds. 2002. Submitted to *Journal of Machine Learning Research*.
- [3] M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. In *NIPS*, volume 14, 2002.
- [4] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *NIPS*, volume 15, 2003.
- [5] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
- [6] D. Haussler, M. Kearns, and R. Schapire. Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension. *Machine Learning*, 14(1), 1994.
- [7] N. E. Sanjana and J. B. Tenenbaum. Bayesian models of inductive generalization. In *NIPS*, volume 15, 2003.
- [8] C. Kemp and J. B. Tenenbaum. Theory-based induction. In *Proceedings of the 25th Annual Conference of the Cognitive Science Society*, 2003. To appear.
- [9] L. Shih and D. Karger. Learning classes correlated to a hierarchy. 2003. Unpublished manuscript.
- [10] J.-P. Vert. A tree kernel to analyze phylogenetic profiles. *Bioinformatics*, 1(1):1–9, 2002.
- [11] H. Jow, C. Hudelot, M. Rattray, and P. Higgs. Bayesian phylogenetics using an RNA substitution model applied to early mammalian evolution. *Molecular Biology and Evolution*, 19(9):1951–1601, 2002.