Tree Dependent Identically Distributed Learning

Tony Jebara
Computer Science Department
Columbia University
New York, NY 10027

Philip M. Long
Center for Computational Learning Systems
Columbia University
New York, NY 10027

Abstract

We view a dataset of points or samples as having an underlying, yet unspecified, tree structure and exploit this assumption in learning problems. Such a tree structure assumption is equivalent to treating a dataset as being tree dependent identically distributed or *tdid* and preserves exchangeability. This extends traditional *iid* assumptions on data since each datum can be sampled sequentially after being conditioned on a parent. Instead of hypothesizing a single best tree structure, we infer a richer Bayesian posterior distribution over tree structures from a given dataset. We compute this posterior over (directed or undirected) trees via the Laplacian of conditional distributions between pairs of input data points. This posterior distribution is efficiently normalized by the Laplacian's determinant and also facilitates novel maximum likelihood estimators, efficient expectations and other useful inference computations. In a classification setting, *tdid* assumptions yield a criterion that maximizes the determinant of a matrix of conditional distributions between pairs of input and output points. This leads to a novel classification algorithm we call the Maximum Determinant Machine. Unsupervised and supervised experiments are shown.

1 Introduction

In many applied datasets, input data points or samples collectively exhibit additional structure which can be exploited by a classification or learning machine. For instance, data may lie on a low dimensional linear subspace or nonlinear manifold. Another possibility is that data may be clustered into several subcomponents. Alternatively, data may have some graph structure that ties points through some adjacency matrix. In this article, we consider the case where data points obey not a single graph structure but rather a *distribution* of structures, more specifically, a distribution over trees.

Traditionally, imposing additional structure in data is made via parametric assumptions and priors using, for instance, a Bayesian approach which fully models the generative distributions that produced the data. Typically, data is assumed to have been generated in an independent identically distributed or *iid* manner (Box & Tiao, 1992; Ghahramani & Beal, 1999). Alternatively, in support vector machines and their variants, we impose parametric assumptions on the *relationship* between inputs and outputs or on the *flexibility* of decision boundaries that separate data into different classes (Scholkopf & Smola, 2001). Other viable approaches include assumptions that data is confined to a linear subspace as in principal components analysis (PCA) or to a nonlinear manifold as in kernel-PCA and its variants (Scholkopf & Smola, 2001; Tenenbaum et al., 2000). Similarly, assumptions on the presence of clustering or clusters within the data are also useful and exploitable via spectral clustering or mixture modeling (Meila & Shi, 2001; Ghahramani & Beal, 1999). Finally, we can also adopt graph-theoretic assumptions about the data by, for example, building a graph or a *Laplacian* from the dataset and using diffusion or other graph algorithms in the learning process (Kondor & Lafferty, 2002; Belkin & Niyogi, 2001).

The graphical modeling literature brings an alternative way to impose additional structure to data by making graph-structured independence assumptions (Heckerman et al., 1995; Jordan, 2004). However, these are primarily imposed on the relationship between dimensions or random variables that constitute each datum and less frequently on the relationship between individual data points themselves. For instance, a tree structure could be hypothesized between various

random variables in a multi-variate learning problem and efficient algorithms for finding such a tree structure from mutual information computations are known (Chow & Liu, 1968). A more expressive route is to manipulate *distributions* over tree graph structures which are possible in a discriminative classification or generative Bayesian setting (Jaakkola et al., 1999; Meila & Jaakkola, 2000). These involve distributions over graph structures between random variables or dimensions in the dataset. In this article, we instead manipulate distributions over graph structures that interconnect data points or individual sample datums with dependency links, not just links between the random variables that constitute each datum[1].

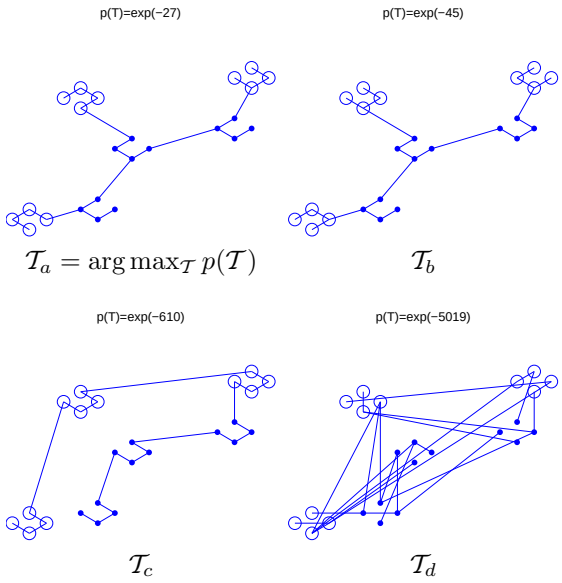


Figure 1: Sample tree structures and their likelihoods for a 2D binary classification dataset.

Thus, we will hypothesize a distribution over tree structures that interconnect the samples in a dataset. This is equivalent to making the assumption that the data in our dataset was generated according to a tree dependent identically distributed (*tdid*) sampling scheme. Applied settings where tree structures are known to exist between data points include biological datasets where each datum is a gene and phylogenetic trees are known to interconnect the genes through a tree-like evolution structure. For any given dataset, we assume the true tree structure is unknown. However, estimating a single best tree structure hypothesis might be unreliable. It is more cautious to consider a distribution over possible tree structures that interconnect the samples. This is shown in Figure 1 where the most likely tree structure and several sample tree structures are drawn interconnecting a dataset of points with the likelihood weight of each tree shown above. This article will make tree structure distribution assumptions between samples and manipulate the posterior over trees leading to efficient ways to regularize and learn from data in both supervised and unsupervised settings.

This paper is organized as follows. Section 2, introduces the *tdid* sampling assumption where a distribution is inferred over *undirected* tree structures that connect data points. In Section 3, we generalize the distributions to *directed* tree structures. Section 4 provides a novel variant of maximum likelihood for *tdid* settings. In Section 5, the *tdid* assumption is exploited in a supervised input-output or classification setting to make label predictions. Section 6 provides theoretical arguments for tree structure assumptions on data. We conclude with experiments and discussion.

2 Tree dependent identically distributed data

In most unsupervised learning problems, we are provided with a training dataset containing X_t input samples for $t = 1 \dots T$. It is traditional to assume, given a model, these samples are independent and identically distributed (*iid*). For instance, the likelihood given a parametric model Θ is:

$$p(X_1, \dots, X_T|\Theta) \quad = \quad \prod_{t=1}^{T} p(X_t|\Theta).$$

From this starting point, it is straightforward to perform inference. For example, we can find the maximum likelihood model Θ^* and evaluating its likelihood numerically. Bayesian analogs of these procedures include performing Bayesian inference over Θ and evaluating the Bayesian evidence of the dataset. However, making an *iid* assumption might be too radical. A more conservative yet necessary assumption we will make in this paper are that our finite samples are only *exchangeable*. This merely states that the likelihood or probability $p(X_1, \dots, X_T|\Theta)$ for finite T is invariant to reordering or permutations of the arguments $\{X_1, \dots, X_T\}$.

Let us instead consider a situation where samples are not *iid* but rather are generated according to an unknown tree structure. More specifically, we assume a dataset is composed of *tree dependent identically distributed* or *tdid* samples. Recall that a tree is a directed

[1] However, an even richer setting would be to manipulate graph structure distributions that interconnect *both* random variables that constitute each data point as well as interconnect the multiple samples in a dataset. For instance, consider a database of patients connected by a family tree where each patient record has a tree structure between variables.

graph $\mathcal{G}$ with a set of T vertices $\mathcal{V}$ and edges $\mathcal{E}$ such that each node $\mathcal{V}_t$ has at most one parent node $\mathcal{V}_{\pi(t)}$. If we knew the tree structure $\mathcal{T}$ that governed our T samples, the likelihood of the data under *tdid* assumptions factorizes as follows:

$$p(X_1, \ldots, X_T|\mathcal{T}, \Theta) \quad = \quad \prod_{t=1}^{T} p(X_t|X_{\pi(t)}, \Theta). \quad (1)$$

However, in general we do not know the tree structure so we therefore treat it as a random variable. This allows us to use Bayes' rule to obtain a posterior distribution over tree structures as follows:

$$p(\mathcal{T}|X_1, \ldots, X_T) \quad = \quad \frac{\prod_{t=1}^{T} p(X_t|X_{\pi(t)})p(\mathcal{T})}{p(X_1, \ldots, X_T)}.$$

In the above we have omitted the Θ conditioning variable for brevity. We will assume without loss of generality that the distribution of the root of the tree given its null parent is uniform:

$$p(X_{root}|\{\}) \quad = \quad \text{constant}$$

and the prior over tree structures is uniform[2] as well:

$$p(\mathcal{T}) \quad = \quad \text{constant}.$$

We still need to select conditional distributions for X_t given its parent point $X_{\pi(t)}$ to fully specify the posterior. For now, any choice for $p(X_t|X_{\pi(t)})$ is possible as long as the distribution is symmetric, in other words, $p(X_t|X_{\pi(t)}) = p(X_{\pi(t)}|X_t)$. In fact, since we will ultimately require the posterior remain normalized, it is technically possible to use any non-negative kernel function $k(X_t, X_{\pi(t)})$ and simply set the conditional distribution $p(X_t|X_{\pi(t)}) \propto k(X_t, X_{\pi(t)})$. For example, we may assume that the points are in a Euclidean space such that $X_t \in \mathbb{R}^d$ and the conditional of X_t given its parent $X_{\pi(t)}$ is a Gaussian centered at the parent with spherical covariance $p(X_t|X_{\pi(t)}) = \mathcal{N}(X_t|X_{\pi(t)}, \sigma^2 I)$. This is equivalent to choosing an RBF kernel for k.

Let us now compute the posterior as a product of edges in the tree instead of a product over nodes given their parents as follows:

$$p(\mathcal{T}|X_1, \ldots, X_T) \; = \; \frac{1}{Z}\prod_{t=1}^{T} p(X_t|X_{\pi(t)}) \; = \; \frac{1}{Z}\prod_{uv\in\mathcal{T}} \beta_{uv}.$$

Here we have written the distribution as a product of the edges uv in the tree $\mathcal{T}$ where we defined $\beta_{uv} = p(X_u|X_v)$ Note that $\beta_{uv} = \beta_{vu} \geq 0$ and that we will assume $\beta_{vv} = 0$ which is never iterated over in the product above since there are no edges between a node and itself. For the Gaussian case mentioned earlier, we would therefore use $\beta_{uv} = \mathcal{N}(X_u|X_v, \sigma^2 I)$. Writing the posterior in terms of β_{uv} as a product of edges effectively discards the directionality implied by a parent-child relationship and this is why we previously required that the conditional distributions remain symmetric. Next, we note the scalar Z which is the partition function that serves to normalize the posterior distribution and is defined as:

$$Z \quad = \quad \sum_{\mathcal{T}} \prod_{uv\in\mathcal{T}} \beta_{uv}.$$

Computing this partition function by enumerating all possible trees is intractable since, according to *Cayley's formula*, there are T^{T-2} possible trees connecting T observation vertices. We adopt a key result from (Jaakkola et al., 1999; Meila & Jaakkola, 2000), namely Kirchoff's classic *Matrix Tree Theorem*, which provides an efficient way to compute the partition function Z by simple linear algebra operations on the $T \times T$ symmetric matrix β whose entries are all the β_{uv} values. Recall, these are merely the conditional distributions between all pairs of points X_u and X_v. We first define a function that maps matrices to matrices $\bar{Q}(\beta)$ where $\bar{Q} : \Re^{T\times T} \to \Re^{T\times T}$ has entries that are defined as follows:

$$\bar{Q}_{uv}(\beta) \quad = \quad \begin{cases} -\beta_{uv} & u \neq v \quad u, v \in [1, T] \\ \sum_{w=1}^{T} \beta_{vw} & u = v \quad u, v \in [1, T] \end{cases}$$

This is similar to the graph *Laplacian* (the discrete analog of the Laplace-Beltrami operator) obtained from weights β_{uv} between nodes. Also, we define another function $Q(\beta)$ that maps matrices to smaller matrices where $Q : \Re^{T\times T} \to \Re^{(T-1)\times(T-1)}$. The output matrix from the Q function is the same as the output matrix of the $\bar{Q}$ function except the last row and column are omitted. Thus Q is merely a minor of $\bar{Q}$.

$$Q_{uv}(\beta) \quad = \quad \begin{cases} -\beta_{uv} & u \neq v \quad u, v \in [1, (T-1)] \\ \sum_{w=1}^{T} \beta_{vw} & u = v \quad u, v \in [1, (T-1)] \end{cases}$$

Using the *Matrix Tree Theorem*, the partition function that normalizes the posterior over trees is then efficiently given by the determinant:

$$Z \quad = \quad |Q(\beta)|.$$

Remarkably, this only requires $\mathcal{O}(T^3)$ operations instead of $\mathcal{O}(T^{T-2})$ operations. It is also interesting to note that the determinant or the value of Z is the same for any $(T-1) \times (T-1)$ minor of the matrix $\bar{Q}(\beta)$.

The quantity Z (also known as the determinant of the graph Laplacian (Jakobson & Rivin, 2002)) and its

[2]Other priors or even mixtures of priors on tree structure are feasible yet the uniform prior avoids assuming an ordering on $X_1, \ldots, X_T$ maintaining exchangeability.

analog for directed trees are useful for various learning problems which we will enumerate shortly. Furthermore, there exist many uses of the normalized posterior distribution over trees given our *tdid* dataset $X_1, \ldots, X_T$. In fact, we also have implicitly described a fully generative model from our joint distribution $p(X_1, \ldots, X_T, \mathcal{T})$. One useful procedure is to sample from such a generative model for various applications including MCMC sampling methods. For instance, we sample a tree structure from the prior distribution $p(\mathcal{T})$ and then sample points from the tree sequentially starting with its root by sampling each datum from a Gaussian centered on the parent. This generative model contrasts the usual assumption that $X_1, \ldots, X_T$ are independent and identically distributed samples.

Furthermore, if we have a normalized posterior distribution $p(\mathcal{T}|\ldots) = \frac{1}{Z}\prod_{uv\in\mathcal{T}}\beta_{uv}$, many interesting computations over trees are efficient because they do not need explicit enumeration of every possible tree structure. For instance, we can find the maximum of the posterior, $\arg\max_{\mathcal{T}} p(\mathcal{T}|\ldots)$ in polynomial time using the Chow-Liu algorithm (Chow & Liu, 1968). Furthermore, we may efficiently compute expectations under the tree distribution of functions over the data that are either additive or multiplicative over edges in a given tree (Meila & Jaakkola, 2000).

3 Directed tdid assumptions

So far, we have discarded the parent-child relationship by assuming symmetric conditional distributions and only considered the *tdid* scenario for undirected trees. This assumption is not necessary since it is also possible to consider a distribution over directed tree structures. This is facilitated by a directed variant of the *Matrix Tree Theorem*, namely Tutte's *Directed Matrix Tree Theorem* (West, 1996). We represent a distribution over trees with T nodes again using a $T \times T$ matrix, β which satisfies the properties $\beta_{vv} = 0$ and $\beta_{uv} \geq 0$ but *does not require symmetry*. For a directed tree $\mathcal{T}$, our probability distribution is again:

$$p(\mathcal{T}) \quad = \quad \frac{1}{Z}\prod_{uv\in\mathcal{T}}\beta_{uv} \tag{2}$$

where we are taking the products of edges along the directed tree that connect node u as a parent of node v, in other words $u \to v$. In other words, β_{uv} is the number or weight of the edges connecting node u to v in a digraph. This exactly coincides with the parent-child factorization of directed Bayesian networks we began with in earlier sections. However, we no longer need to enforce symmetry of the conditional distributions $p(X_t|X_{\pi(t)})$. Furthermore, if β happens to be symmetric, we reproduce the results from the previous sections. Once again note the intractability of the partition function Z due to the large number of directed tree structures.

To apply the *Directed Matrix Tree Theorem*, we first split the set of directed trees with T nodes which we denote with $\{\mathcal{T}\}$ into $2T$ different subsets. More specifically, we split the space into *in-trees* rooted at node $i \in [1, T]$ which we denote $\{\mathcal{T}_i^+\}$ and *out-trees* rooted at node $i \in [i, T]$ which we denote $\{\mathcal{T}_i^-\}$. An *in-tree* is an orientation of a tree having a root of out-degree 0 and all other vertices having out-degree 1. An *out-tree* or *branching tree* is an orientation of a tree having a root of in-degree 0 and all other vertices having in-degree 1. From β we compute the following two $T \times T$ matrices, the in-tree matrix $\bar{Q}^+$ and the out-tree matrix $\bar{Q}^-$ which only differ on their diagonals:

$$\begin{aligned}
\bar{Q}^+_{uv}(\beta) &= \begin{cases} -\beta_{uv} & u \neq v \quad u, v \in [1, T] \\ \sum_{w=1}^T \beta_{vw} & u = v \quad u, v \in [1, T] \end{cases} \\
\bar{Q}^-_{uv}(\beta) &= \begin{cases} -\beta_{uv} & u \neq v \quad u, v \in [1, T] \\ \sum_{w=1}^T \beta_{wv} & u = v \quad u, v \in [1, T] \end{cases}
\end{aligned}$$

The *Directed Matrix Tree Theorem* states the following key results[3]. The number (or weight) of in-trees rooted at node i is $Z_i^+(\beta)$ and is given by the matrix cofactor obtained by deleting the i'th row and i'th column of the matrix $\bar{Q}^+$:

$$Z_i^+(\beta) \quad = \quad |m_{ii}(\bar{Q}^+(\beta))|$$

where $m_{ii}(\bar{Q})$ is the sub-matrix obtained by deleting the i'th row and i'th column of matrix $\bar{Q}$. Similarly, the weight of out-trees rooted at node i is

$$Z_i^-(\beta) \quad = \quad |m_{ii}(\bar{Q}^-(\beta))|.$$

We can now precisely define the distribution over trees conditioned on the fact that the trees are in-tree structures rooted at node i as

$$p(\mathcal{T}|+, i) \quad = \quad \frac{1}{Z_i^+}\prod_{uv\in\mathcal{T}}\beta_{uv}.$$

The conditional over out-trees is similar. We can also recover the marginal distribution over directed trees by using these conditionals via

$$p(\mathcal{T}) \quad = \quad \sum_{\pm, i} p(\mathcal{T}|\pm, i)p(\pm, i).$$

Here, we have a marginal distribution $p(\pm, i)$ which has $2T$ entries and assigns a probability to selecting

[3]Technically, the *Directed Matrix Tree Theorem* (West, 1996) applies to a matrix of integer weights on trees yet can be extended to continuous rational values by assuming a matrix composed of rationals with a very large common denominator that is factored out from the probability leaving an integer matrix (Meila & Jaakkola, 2000).

a node $i \in [1, T]$ as the root for both the case of an in-tree and for an out-tree. The natural setting for the marginal distribution for each possible choice of root and in/out-tree structure is

$$p(\pm, i) \quad = \quad \frac{Z_i^{\pm}}{\sum_{j=1}^{T} Z_j^{+} + Z_j^{-}}.$$

Therefore, we can now succinctly write out a normalized distribution over directed trees by marginalizing over the choices of root and direction (in-tree or out-tree):

$$\begin{aligned} p(\mathcal{T}) &= \sum_{\pm, i} \frac{1}{\sum_{j=1}^{T} Z_j^{+} + Z_j^{-}} \prod_{uv \in \mathcal{T}} \beta_{uv} \\ &= \frac{2T}{\sum_{j=1}^{T} Z_j^{+} + Z_j^{-}} \prod_{uv \in \mathcal{T}} \beta_{uv} \end{aligned}$$

Thus, the partition function Z for the original Equation 2 for the distribution over directed trees can be recovered from the above equation as simply

$$Z \quad = \quad \frac{1}{2T} \sum_{i=1}^{T} |m_{ii}(\bar{Q}^{+}(\beta))| + |m_{ii}(\bar{Q}^{-}(\beta))|.$$

In the case when β is symmetric, all terms in the summation above are identical and the above partition function is the same as the one in the undirected case. Thus, we can see the case for directed trees is a natural generalization of the undirected tree case. This is a well known result since the *Directed Matrix Tree Theorem* reduces to the *Matrix Tree Theorem* when the digraph is symmetric (West, 1996). However, normalizing the distribution over directed trees for asymmetric β is more computationally demanding since we compute $2T$ determinants which requires $\mathcal{O}(T^4)$ operations. Thus, considering a posterior over directed trees allows us to work with asymmetric conditional distributions between parent and child nodes. For instance, we may choose a Gaussian relationship for the conditional $p(X_t|X_{\pi(t)}) = \mathcal{N}(X_t|0, X_{\pi(t)}X_{\pi(t)}^T + \sigma^2 I)$ which is zero-mean yet has noise varying with the magnitude and direction of the parent.

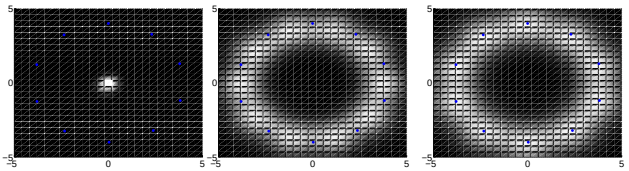
4 Maximum *tdid* likelihood

We can now consider computing the likelihood under *tdid* assumptions instead of *iid* assumptions and, if we treat the tree structure as unknown and use a uniform prior over trees, we obtain a likelihood $p(X_1, \ldots, X_T|\Theta)$ that is *exchangeable* yet not *iid*. Here, we have reinserted the Θ parameter that modifies the conditional relationship $p(X_t|X_{\pi(t)}, \Theta)$ between pairs of data-points. Note that the *tdid* likelihood given a model Θ and a uniform prior over trees is merely the partition function

$$\begin{aligned} p(X_1, \ldots, X_T|\Theta) &= \sum_{\mathcal{T}} p(X_1, \ldots, X_T|\mathcal{T}, \Theta) p(\mathcal{T}) \\ &= \sum_{\mathcal{T}} \prod_{uv \in \mathcal{T}} \beta_{uv}(\Theta) \quad = \quad Z(\Theta) \end{aligned}$$

For both the directed and undirected cases, the determinants that constitute the partition function are invariant to reordering of $\{X_1, \ldots, X_T\}$ which clearly indicates that this likelihood is still exchangeable. It now becomes straightforward to compute a maximum likelihood setting of a model Θ^* by maximizing the *tdid* likelihood instead of the *iid* likelihood. Another interesting property is that the *tdid* likelihood is a convex combination of Laplacian determinants which are log-concave in the edge weights β_{uv} as shown in (Jakobson & Rivin, 2002). We also *conjecture* that if $\beta_{uv}(\Theta)$ is log-concave in its parameters Θ (i.e. it is in the exponential family) then $Z(\Theta)$ might be log-concave and have no local minima. This is due to Equation 1 in $\{\Theta, \mathcal{T}\}$ which might maintain log-concavity under integration over $\mathcal{T}$ via Prekopa's theorem (West, 1996). In either case, we may perform maximum likelihood estimation, possibly via gradient ascent in Θ over the determinants in $Z(\Theta)$ noting that $\frac{\partial}{\partial A}|A| = |A|A^{-1}$.

For instance, for jointly Gaussian conditionals between X_t and $X_{\pi(t)}$ via $p(X_i|X_{\pi(t)}, \Theta) = \mathcal{N}(X_i|\Theta X_{\pi(t)}, I)$ we would adjust Θ to maximize *tdid* likelihood. In Figure 2 we show a toy experiment of 10 data points arranged in a circle and compute the likelihood surface after marginalizing over tree structures. The *tdid* log-likelihood is listed for each setting of the Θ matrix, including the maximum likelihood setting. The *tdid* likelihood is highest at the smoothest version of the density and captures the rotational relationship between the points.



$p(\mathcal{X}|\Theta) = 3e^{-31}$ $\quad$ $p(\mathcal{X}|\Theta) = 7e^{-19}$ $\quad$ $p(\mathcal{X}|\Theta) = 9e^{-8}$

$$\Theta = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \Theta = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \Theta = \begin{bmatrix} 0.8 & 0.6 \\ -0.6 & 0.8 \end{bmatrix}$$

Figure 2: Unsupervised maximum *tdid* likelihood estimation and density surface. The left-most plot corresponds to a zero-mean Gaussian with no dependence on a parent. The middle one correponds to an RBF-like dependence on the parent and the right-most plot corresponds to a rotational dependence on a parent.

5 Classification under tdid

We next consider the *tdid* assumptions in supervised learning problems. We focus on binary classification yet the results extend to multi-class classification and regression problems as well. In supervised scenarios, we have outputs or labels in addition to input samples. Instead of generating a label that depends on the input, we will also allow dependence of the label on a parent input and a parent label. In other words, for training samples X_t and corresponding labels y_t for $t = 1 \dots T$, we have the following likelihood under *tdid* assumptions for a known tree $\mathcal{T}$:

$$p(X_1, y_1, \dots, X_T, y_T|\mathcal{T}) = \prod_{t=1}^{T} p(X_t, y_t|X_{\pi(t)}, y_{\pi(t)}). \quad (3)$$

Once again, we have conditional distributions $p(X_t, y_t|X_{\pi(t)}, y_{\pi(t)})$ which may or may not be symmetric. Furthermore, we again choose uniform priors on the root and a uniform prior on tree structure to compute the posterior:

$$p(\mathcal{T}|X_1, \dots, X_T, y_1, \dots, y_T) = \frac{1}{Z} \prod_{uv \in \mathcal{T}} \beta_{uv}$$

where we have defined $\beta_{uv} = p(X_u, y_u|X_v, y_v)$. The partition function is again provided via the previous formulas for directed or undirected trees.

We may make more restrictive factorizations assumptions on the *tdid* formulation above. For instance, y_t might depend only on X_t. Alternatively, y_t might be independent of $X_{\pi(t)}$ given both $y_{\pi(t)}$ and X_t. However, these factorizations require specifying a relationship between inputs and outputs. A more radical assumption is that y_t is independent of input X data given its parent $y_{\pi(t)}$. In that case the conditional $p(X_t, y_t|X_{\pi(t)}, y_{\pi(t)})$ simplifies into $p(y_t|y_{\pi(t)})p(X_t|X_{\pi(t)})$. In an *iid* classification setting, this radical assumption makes learning impossible since output data is independent from input data. However, in a *tdid* setting, inputs and outputs are only conditionally independent *given* tree structure. When tree structure is unknown, dependence between inputs and outputs emerges without an explicit relationship between input and output spaces (parametric or otherwise). When we are wary of making explicit assumptions about the relationship between the input and output spaces (beyond a common *tdid* sampling scheme that underlies both), this may actually be a sound factorization.

Once again, tree structure is not available and must be treated as a hidden random variable. Furthermore, in a classification problem, one or more y_t labels are unobserved. One approach is to find the maximum likelihood setting of the unobserved labels while integrating over the nuisance parameter $\mathcal{T}$ since the tree structure in a general *tdid* problem is unavailable. Assume we have observed the first $t = 1 \dots U$ labels (for some U integer less than T) and the $t = (U+1) \dots T$ labels are unobserved. Denote the set of unobserved labels $\mathcal{Y} = \{y_{U+1}, \dots, y_T\}$. We will use the conditional posterior over unobserved labels, $l(\mathcal{Y})$, which is equal to the joint distribution marginalized over all possible tree structures[4]:

$$\begin{aligned} l(\mathcal{Y}) &= p(y_{U+1}, \dots, y_T|X_1, \dots, X_T, y_1, \dots, y_U) \\ &= \sum_{\mathcal{T}} p(y_{U+1}, \dots, y_T, \mathcal{T}|X_1, \dots, X_T, y_1, \dots, y_U) \\ &= \sum_{\mathcal{T}} \frac{p(X_1, \dots, X_T, y_1, \dots, y_T, \mathcal{T})}{p(X_1, \dots, X_T, y_1, \dots, y_U)} \\ &\propto \sum_{\mathcal{T}} p(X_1, \dots, X_T, y_1, \dots, y_T, \mathcal{T}) \\ &\propto \sum_{\mathcal{T}} \prod_{t=1}^{T} p(X_t, y_t|X_{\pi(t)}, y_{\pi(t)}) = \sum_{\mathcal{T}} \prod_{uv \in \mathcal{T}} \beta_{uv}^{\mathcal{Y}} \end{aligned}$$

where, in the last line, we have applied the definition in Equation 3. Here, we use the superscript on β to indicate that these β_{uv} values and β matrix are computed for a specific proposal setting of the $\mathcal{Y}$ variables. Remarkably, the value of the objective function is simply proportional to the partition function Z that normalizes the tree distribution for proposed values of the unlabeled outputs. Thus, we have the following general (directed or undirected) formula:

$$l(\mathcal{Y}) \propto \frac{1}{2T} \sum_{i=1}^{T} |m_{ii}(\bar{Q}^+(\beta^{\mathcal{Y}}))| + |m_{ii}(\bar{Q}^-(\beta^{\mathcal{Y}}))|$$

This $l(\mathcal{Y})$ is an objective function that we maximize to predict the unobserved labels, i.e. $\hat{\mathcal{Y}} = \arg\max_{\mathcal{Y}} l(\mathcal{Y})$. Therefore, we merely compute the partition function from the β matrix for every possible setting of the unlabeled output variables $\mathcal{Y}$. In the binary classification case, this requires computing the partition function 2^{T-U} times. Since the objective function is also a distribution over unlabeled outputs, it can be normalized by the sum of all partition functions evaluations for all settings of $\mathcal{Y}$:

$$l(\mathcal{Y}) = \frac{\sum_{i=1}^{T} |m_{ii}(\bar{Q}^+(\beta^{\mathcal{Y}}))| + |m_{ii}(\bar{Q}^-(\beta^{\mathcal{Y}}))|}{\sum_{\mathcal{Y}} \sum_{i=1}^{T} |m_{ii}(\bar{Q}^+(\beta^{\mathcal{Y}}))| + |m_{ii}(\bar{Q}^-(\beta^{\mathcal{Y}}))|}$$

The above framework suggests building a classification machine by labeling test data such that the partition

[4] We can also introduce parameters Θ that adjust the conditional relationship $p(X_t, y_t|X_{\pi(t)}, y_{\pi(t)}, \Theta)$ and integrate over them with a prior $p(\Theta)$ as well as integrate over tree structures. This would modify the β_{uv} values.

function of determinants is maximized. We call this classifier a *Maximum Determinant Machine* or *Max Det Machine* for short. Note that if the number of test points is large the 2^{T-U} determinant evaluations become expensive yet various improvements are possible. For prediction, we typically only need the optimal setting $\hat{\mathcal{Y}}$ and not a full distribution over $\mathcal{Y}$. In these scenarios, convex and semidefinite programming methods to maximize determinants may be used (Boyd & Vandenberghe, 2003). We may also approximate the optimal setting of $\mathcal{Y}$ by only solving for small subsets of variables within $\mathcal{Y}$ at a time. For instance, in the two-class classification case, we simply compute partition functions for every single binary variable in $\mathcal{Y}$ at a time to determine its class which requires $2(T-U)$ evaluations of Z. Finally, we may also selectively use Woodbury's formula $|A+VW^T| = |A||I+W^TA^{-1}V|$ to speed up various determinant computations.

6 VC dimension of tree classifiers

The above approaches use posteriors over tree structures to bias maximum likelihood estimation and classification predictions. An extreme setting of this posterior could select a single tree or hierarchy (with all probability mass on $\mathcal{T}^*$). We could then consider ways to cut the tree into two parts such that, to the greatest extent possible, members of one class fall in one part, and members of the other class fall into the other. This extreme approach imposes a strong inductive bias on the learning algorithm and can be motivated using frequentist and theoretical arguments. We next provide evidence for this argument by bounding the VC dimension of a classifier based on cutting a tree.

Many generalization guarantees have been proved in terms of the VC dimension (see (Vapnik & Chervonenkis, 1971; Talagrand, 1994)) which is defined as follows. Suppose X is a set, and F is a set of $\{-1,1\}$-valued functions defined on X. We say F shatters $x_1, ..., x_d$ if

$$\{(f(x_1), ..., f(x_d)) : f \in F\} = \{-1,1\}^d,$$

that is, if functions in F are able to assign an arbitrary sequence of values to $x_1, ..., x_d$. The VC dimension of F is the size of the largest set shattered by F.

Theorem 1 *Suppose $G = (V,E)$ is an undirected graph, and that E is a spanning tree for V. For an edge $e \in E$, say that a function f from V to $\{-1,1\}$ is* compatible with a cut at *e, if cutting the graph at e separates the graph into two components, and f is constant on each component. In other words, f is compatible with a cut at e if for all $v_1, v_2 \in V$, $f(v_1) = f(v_2)$ if and only if the path from v_1 to v_2 in G does not contain e.*

Let F be the set of all $f : V \to \{-1,1\}$ that are compatible with a cut at some edge in E. Then the VC-dimension of F is at most 3.

Proof: We will show that no four vertices can be shattered by F.

Assume for contradiction that $w_1, ..., w_4$ can be shattered. Consequently, there is an $f \in F$ such that $f(w_1) = f(w_2) = 1$ and $f(w_3) = f(w_4) = -1$. Let e be the edge that was cut to define f. Since $w_1, ..., w_4$ are shattered, there is also a $g \in G$ such that $g(w_1) = g(w_3) = 1$ and $g(w_2) = g(w_4) = -1$. Obviously g is different from f, so g is defined by a cut other than at e. Thus e must lie in one or the other of the components of the graph, after it is cut to define g. Suppose it is in the component on which g always takes the value -1 (the other case can be argued similarly). Then the fact that $g(w_1) = g(w_3) = 1$ implies that the path from w_1 to w_3 does not contain e. But the fact that w_1 and w_3 were assigned different values by f implies that the path from w_1 to w_3 does contain e, a contradiction. □

The fact that this VC-dimension is bounded implies, roughly, that if an algorithm is able to correctly identify the hierarchical structure, and if the view that the class is aligned with the structure is correct, then accuracy will be good. The above bound also directly implies a similar generalization guarantee for algorithms that seek to cut the graph in few places. Note that if the graph is cut in s places, then the region on which the classifier takes that value 1 is a union of at most s subtrees. Thus, general bounds on the VC-dimension of composite classes formed by taking unions of members of simpler classes (Blumer et al., 1989) imply that the VC dimension of the class of such s-cut hypotheses is $\mathcal{O}(s \log s)$. This provides generalization guarantees for algorithms that trade off between the number of cuts and the fit to the data. Once again, if the hierarchy is learned accurately, and the inductive bias that there is a good classifier with few cuts is justified, such algorithms should be expected to perform well.

Unfortunately, in realistic settings, the estimation of a single tree or hierarchy can be unstable. This motivates considering multiple hierarchies in an algorithm and voting over the results, to improve the stability of the resulting classifier as in our aforementioned Bayesian approach.

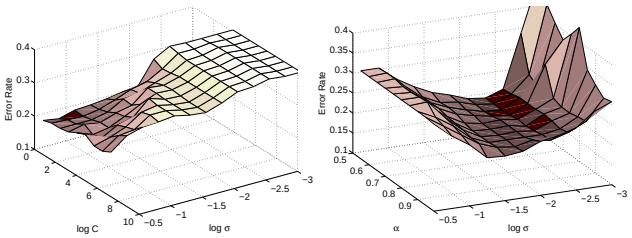
7 Experiments

To evaluate how tree structure can be used in classification problems, we compared the Max Det Machine against an SVM using an RBF kernel on a standard UCI classification problem, the Pima Indians dataset.

For the Max Det Machine, we set the conditional distribution to be the product of the following:

$$\begin{aligned} p(X_t|X_{\pi(t)}) &= \mathcal{N}(X_t|X_{\pi(t)}, \sigma^2 I) \\ p(y_t|y_{\pi(t)}) &= \alpha\delta(y_t = y_{\pi(t)}) + (1-\alpha)\delta(y_t \neq y_{\pi(t)}). \end{aligned}$$

Figure 3 summarizes the results across many settings of their classifier parameters. Here, we only use the Max Det Machine to predict a single output at a time, i.e. $T = U + 1$. Both methods perform well, achieving an error rate of roughly 20% at good settings of the σ, α and C parameters. Thus, the assumption of a common tree structure on inputs and outputs seems to approriately guide classification predictions.



Support Vector Machine Max Determinant Machine

Figure 3: Error Rate on Pima Indians Diabetes UCI Dataset. Various settings of the Max Det Machine's α parameter and the Support Vector Machine C regularization are explored while varying the RBF kernel's σ. Training and test set sizes were 384 samples each.

8 Discussion

We have seen how making *tdid* assumptions or imposing an unknown tree structure on data-points leads to an interesting and efficient posterior distribution on directed or undirected tree graphs. This posterior and its normalizer facilitate various efficient maximizations, expectations over structure, new variants of maximum likelihood estimation and new approaches to classification which are motivated by VC arguments. We are exploring other applications of these tree posterior distributions, including regression. Furthermore, other graph assumptions beyond tree-based ones may lead to efficient posterior distributions on the structures connecting data-points. We are exploring subsets of trees (such as lobsters or caterpillars), star-graphs as well other spectral restrictions on graphs via the eigenvectors and eigenvalues of the Laplacian (West, 1996). One promising avenue is provided by the discovery of yet another *Matrix Tree Theorem* by Chung and Chaiken for counting *rooted spanning forests* using determinant computations.

References

Belkin, M., & Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Neural Information Processing Systems*.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. K. (1989). Learnability and the Vapnik-Chervonenkis dimension. *JACM*, *36*, 929–965.

Box, G., & Tiao, G. (1992). *Bayesian inference in statistical analysis*. John Wiley & Sons.

Boyd, S., & Vandenberghe, L. (2003). *Convex optimization*. Cambridge University Press.

Chow, C., & Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, *14*, 462–467.

Ghahramani, Z., & Beal, M. (1999). Variational inference for Bayesian mixture of factor analysers. *Advances in Neural Information Processing Systems 12*.

Heckerman, D., Geiger, D., & Chickering, D. (1995). Learning Bayesian networks: the combination of knowlegdge and statistical data. *Machine Learning*, *20*, 197–243.

Jaakkola, T., Meila, M., & Jebara, T. (1999). Maximum entropy discrimination. *Advances in Neural Information Processing Systems 12*.

Jakobson, D., & Rivin, I. (2002). Extremal metrics on graphs. *Forum Math*, *14*.

Jordan, M. (2004). Graphical models. *Statistical Science (Special Issue on Bayesian Statistics)*, *19*, 140–155.

Kondor, R., & Lafferty, J. (2002). Diffusion kernels on graphs and other discrete input spaces. *Machine Learning: Ninth International Conference*.

Meila, M., & Jaakkola, T. (2000). Tractable Bayesian learning of tree belief networks. *Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence*.

Meila, M., & Shi, J. (2001). A random walks view of spectral segmentation. *Conference on AI and Statistics (AISTATS)*.

Scholkopf, B., & Smola, A. (2001). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press.

Talagrand, M. (1994). Sharper bounds for Gaussian and empirical processes. *Annals of Probability*, *22*, 28–76.

Tenenbaum, J., De Silva, V., & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*.

Vapnik, V. N., & Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, *16*, 264–280.

West, D. (1996). *Introduction to graph theory*. Prentice Hall.