Learning from Out-Tree Dependent Data

Tony Jebara, Columbia University, jebara@cs.columbia.edu

Assume we are given a training dataset containing X_t input samples for $t = 1 \ldots T$ in some arbitrary order. One quantity to evaluate or manipulate is the likelihood of the dataset $p(X_1, \ldots, X_T|\theta)$ given some model. A popular method to recover a model of the dataset is to find the model that maximizes the likelihood score. An additional standard assumption most unsupervised methods make when given a dataset is that it is composed of independently identically distributed samples. In other words, $p(X_1, \ldots, X_T|\theta) = \prod_{t=1}^{T} p(X_t|\theta)$. This *iid* assumption can be inappropriate for many real datasets. Consider instead that we first sampled a tree connectivity over T nodes. Then children X_t are sampled from their parents $X_{\pi(t)}$ using conditional distributions $p(X_t|X_{\pi(t)}, \theta)$ according to this tree structure (as is the case in a single-parent family tree). More formally, the structure we are dealing with is an out-tree. This is as an acyclic graph $\mathcal{T}$ with a set of T vertices $\mathcal{X}$ and edges $\mathcal{E}$ such that each node $\mathcal{X}_t$ has at most one parent node $\mathcal{X}_{\pi(t)}$. Rooted out-trees are trees with directed edges pointing away from a well-defined root. For instance, $\mathcal{X}_1 \leftarrow \mathcal{X}_2 \leftarrow \mathcal{X}_3$ is an out-tree rooted at $\mathcal{X}_3$. Conversely, rooted in-trees have all directed edges from other nodes point towards the root. The previous 3-chain example is thus also an in-tree rooted at node $\mathcal{X}_1$. Many directed trees are neither in-trees nor out-trees. For instance, the tree $\mathcal{X}_1 \rightarrow \mathcal{X}_2 \leftarrow \mathcal{X}_3 \rightarrow \mathcal{X}_4$ is a valid directed tree but neither a rooted in-tree nor a rooted out-tree.

If we knew the latent out-tree structure $\mathcal{T}$ that generated our T samples, the likelihood of the data would factorize as a product of conditionals of each node given its parent. However, in general, the structure is unknown. Consider treating it as a random variable and using Bayes' rule to obtain a posterior distribution over tree structures as follows (assuming a uniform prior over out-trees):

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

where the *partition function* $Z = p(\mathcal{X})T^{T-1}$ ensures the likelihood sums to unity. Recovering Z involves summing $p(\mathcal{X}|\mathcal{T})$ over $\mathcal{T}$, the set of all out-trees, Γ. This is an unwieldy computation since there are T^{T-1} possible out-trees connecting T observation vertices. Instead, we consider breaking up the summation into all possible choices of the root of the out-tree $r = 1 \ldots T$ and a summation over the subset Γ_r of all T^{T-2} out-trees rooted at node r. It is straightforward to show that all subsets of out-trees with different roots are distinct, in other words $\Gamma_i \cap \Gamma_j = \{\}$ if $i \neq j$. Furthermore, their union forms the set of all out-trees $\Gamma = \cup_{j=1}^{T}\Gamma_j$. Thus, the partition function Z is decomposable as the following sum:

$$Z \;=\; \sum_{r=1}^{T}\sum_{\mathcal{T}\in\Gamma_r}\prod_{t=1}^{T} p(X_t|X_{\pi(t)}) \;=\; \sum_{r=1}^{T} p(X_r)\sum_{\mathcal{T}\in\Gamma_r}\prod_{t\neq r}^{T} p(X_t|X_{\pi(t)})$$

where we have used the property that the root has no parent node. To efficiently recover Z we will instead recover the individual components of the above sum over r denoted as Z_r by making an appeal to the directed variant of Kirchoff's *Matrix Tree Theorem*, namely Tutte's *Directed Matrix Tree Theorem*. The directed matrix tree theorem does not quite sum over all directed trees. It sums over a *subset*: rooted out-trees. To apply Tutte's theorem we compute an asymmetric β weight matrix of size $T \times T$ populated by all pairwise conditional probabilities according to $\beta_{uv} = p(X_u|X_v)$. Note that we will assume $\beta_{vv} = 0$ since there are no edges between a node and itself. The matrix β allows us to rewrite Z_r as a product of edges in β instead of a product of nodes:

$$Z_r \;=\; \sum_{\mathcal{T}\in\Gamma_r}\prod_{t\neq r}^{T} p(X_t|X_{\pi(t)}) \;=\; \sum_{\mathcal{T}\in\Gamma_r}\prod_{uv\in\mathcal{T}} \beta_{uv}.$$

The out-tree Laplacian matrix Q is then obtained as $Q = diag(\beta\vec{1}) - \beta$. Here, take $\vec{1}$ to be the ones column vector and note that the $\mathrm{diag}(\vec{v})$ operator gives a diagonal matrix with $\vec{v}$ on its diagonal. Note that the Laplacian Q is not symmetric. The directed matrix tree theorem asserts that the number (or weight) of out-trees rooted at node r is Z_r and is given by the matrix cofactor $[Q]_r$ obtained by deleting the r'th row and r'th column of the matrix Q. The precise formula is $Z_r = |[Q]_r|$. Reinserting this formula into the above gives the total partition function as:

$$Z \;=\; \sum_{r=1}^{T} p(X_r) Z_r \;=\; \sum_{r=1}^{T} p(X_r) |[\mathrm{diag}(\beta\mathbf{1}) - \beta]_r|$$

which is now efficient to evaluate. Interestingly, Z is the sum of determinants of the minors of the Laplacian. This is also known as an *immanent*. If β is symmetric, all terms in the summation above are identical and the immanent simply becomes a determinant. This is the case, for example, if the conditional distributions of parent and child satisfies $p(X_t|X_{\pi(t)}) = p(X_{\pi(t)}|X_t)$. In addition, it is known that the log determinant of a symmetric Laplacian matrix is a concave function of the edge-weights. In the asymmetric case, however, concavity is lost. A naive calculation of Z requires $\mathcal{O}(T^4)$ however it is possible to recover Z in $\mathcal{O}(T^{2.6})$ by using a singular value decomposition of β. This is more efficient than enumerating all T^{T-1} out-trees to ensure a normalized likelihood. An interesting property is that the partition function Z forms a finitely exchangeable *otdid* or out-tree dependent identically distributed likelihood as follows:

$$p(X_1, \ldots, X_T) \;=\; \frac{1}{T^{T-1}} \sum_{r=1}^{T} p(X_r) \left|[\mathrm{diag}(\beta\mathbf{1}) - \beta]_r\right|. \tag{1}$$

Theorem 1 *If the conditional dependence of a child node given a parent node degenerates into the marginal $p(X_t|X_{\pi(t)}) \to p(X_t)$ the* otdid *likelihood simplifies into the* iid *likelihood.*

Proof 1 *Work backwards by writing the likelihood as a product over nodes given parents, removing dependence on parents and simplifying:*

$$p(X_1, \ldots, X_T) = \frac{1}{T^{T-1}} \sum_{r=1}^{T} p(X_r) \sum_{\mathcal{T} \in \Gamma_r} \prod_{t \neq r}^{T} p(X_t|X_{\pi(t)}) = \frac{1}{T^{T-1}} \sum_{r=1}^{T} \sum_{\mathcal{T} \in \Gamma_r} \prod_{t=1}^{T} p(X_t) = \prod_{t=1}^{T} p(X_t).$$

A generalization of *iid* likelihood emerges by integrating over latent out-tree structure. To perform unsupervised learning, we maximize this *otdid* likelihood over the parameters θ that govern the conditional distribution of a child given its parent. For example, we considered parameterizing a linear Gaussian conditional relationship. An Expectation-Maximization algorithm is straightforward to derive and leads to efficient unsupervised learning. This Bayesian treatment of out-trees predicts labels more accurately than support vector machines if the data obeys a tree structure such as in the taxonomy datasets below which were introduced by Kemp et al. in NIPS 2003.

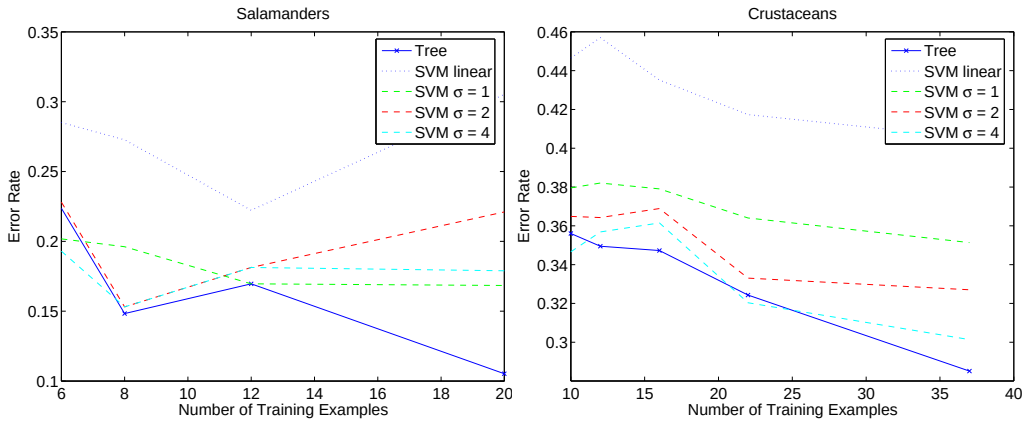

Figure 1: Labeling error rates (averaged over tasks) for Out-Trees and SVMs.