THE USE OF CHI-SQUARED STATISTICS IN DETERMINING DEPENDENCE TREES

R.S. VALIVETI AND B.J. OOMMEN

SCS-TR-153
March 1989

School of Computer Science
Carleton University
Ottawa, Ontario
CANADA K1S 5B6

Both authors partially supported by Natural Sciences and Engineering Research Council of Canada.

The Use of Chi-squared Statistics in Determining Dependence trees

R.S. Valiveti and B.J. Oommen *
Carleton University
Ottawa, Canada, K1S 5B6

Abstract

In several pattern-recognition applications, it is often neccesary to approximate probability distributions with well defined, parametrized density functions. For the case of discrete-valued functions (and for the cases when the features are not necessarily normally distributed), a method known as the dependence-tree exists. This method is based on the metric known as the Expected Mutual Information Measure. This paper studies the suitability of a chi-squared based metric for the same purpose. For a restricted class of distributions, these two metrics are shown to be equivalent and stochastically optimal. For the general cases, the latter metric is almost as efficient as the optimal one, but is computationally far superior.

Keywords:

Pattern recognition, classification, probability distribution, approximation, closeness of approximation, dependence trees.

*Both authors partially supported by Natural Sciences and Engineering Research Council Of Canada.

1 Introduction

The design of pattern-recognition systems, communication systems and information-retrieval systems involve pattern-recognition. That is, the objective is to classify a specific sample (e.g. a hand-written character, a digital signal or a book in the library) into one of a set of known categories. If we denote the measurement made on a sample by X and the various classes by $\{\omega_i \mid i = 1, \ldots C\}$, the problem of pattern classification is indeed one in which we attempt to find a class ω_i which maximizes the likelihood of generating the sample $\mathbf{X}$. The latter naturally leads us to the most common decision procedure, which assigns the sample $\mathbf{X}$ to the class ω_i which maximizes the probability $\Pr(\omega_i \mid \mathbf{X})$. Since the probaility $\Pr(\omega_i \mid \mathbf{X})$ is not readily computable, this quantity (or a quantity related to it, is computed using the well acclaimed Bayes' rule, which relates the *a priori* probabilities to the *a posteriori* probabilities. It is worth observing that this term can be computed only by having an accurate knowledge of the distributions $\Pr(\mathbf{X} \mid \omega_i)$, referred to as the class-conditional distribution.

The measurements made on the sample $\mathbf{X}$, is generally an n-dimensional vector $[x_1, x_2, \ldots, x_N]^T$. The components of the vector $\mathbf{X}$, i.e. $x_1, x_2, \ldots, x_N$, are referred to as the features. The information about the distribution of the features within a class, is typically obtained by a process referred to as "training". During this process, the information system is presented with an array of samples and is simultaneously informed of the class from which the samples originated from. Typically, if the pattern classifier is of the parametric family of classifiers, it assumes the model for the distribution of features and uses the samples to estimate the parameters that characterize the distribution.

Owing to the fact that the number of samples is small, and that real machines often impose limitations on available storage, it is often necessary to restrict the amount of information that can be collected from the samples. The training process is simplified by making various assumptions about the distribution $\Pr(\mathbf{X} \mid \omega_i)$. The simplest assumption made about this distribution is that the features are statistically independent. Although this assumption (i.e. that of feature independence) simplifies the mathematical formula-

tion, it does not necessarily model the real life situation adequately.

To accurately predict the joint distribution of the features, the intention of the system designer is to capture as much of the information about the dependence of features as possible. Thus, in order to account for the dependence between normally distributed (continuous) random variables, it is easy to show that the covariance matrix is adequate; the result being a consequence of the fact that a normal random vector is completely defined by its mean and the covariance matrix. In the case when the features cannot be assumed to normally distributed, or when the random variables are of the discrete sort, a completely new method is required to capture the information about the stochastic dependence of the features.

This paper deals with the problem of representing the information regarding the stochastic dependence between the features. To render the nomenclature and the notation simple, we shall use the vector $\mathbf{X}$ to represent the sample and the measurements made on it. Furthermore in this paper we shall only consider the case when the feature measurements are discrete-valued. Clearly, this constitutes a large number of cases in the field of traditional pattern recognition, and indeed includes the entire field of problems encountered in information-retrieval.

The central issue in tackling this problem is one of approximating the joint distribution (or more exactly the joint density function) $\mathrm{P}(\mathbf{X})$, where $\mathbf{X}$ is the vector $[x_1, x_2, \ldots, x_N]^T$. Because of the largeness of the dimensionality of the vector $\mathbf{X}$ (i.e. because of the large magnitude of N), it is both infeasible and impractical to store estimates of the joint density function $\mathrm{P}(\mathbf{X})$ for all possible values of $\mathbf{X}$. It is infeasible because, if each feature x_i could take one of a set of k distinct values, the number of estimates that would have to be maintained is of the order of k^N. Obviously such a scheme would be impractical also. A second alternative is one of approximating $P(\mathbf{X})$ by a well defined and easily computable density function $P_a(\mathbf{X})$.

When using an approximation density function $P_a(\mathbf{X})$, the question is now one of measuring how well this function approximates the "real" density function $P(\mathbf{X})$. In order to quantify this "goodness" of approximation, we must define a distance measure, between

these two density functions. One such measure, which is an information theoretic measure, has been known to serve this need. It is given below:-

$$I(P, P_a) = \sum_{\mathbf{X}} P(\mathbf{X}) \log \frac{P(\mathbf{X})}{P_a(\mathbf{X})} \tag{1}$$

It is well known that $I(P, P_a) \geq 0$ and $I(P, P_a) = 0$ only if the approximation is exact (i.e. $P(\mathbf{X}) = P_a(\mathbf{X})$, for all $\mathbf{X}$). Hence $I(P, P_a)$ is a measure of the closeness of the approximation; the smaller this measure, the better the approximation. In other words, the intent in finding the best approximation for a given density function $P(\mathbf{X})$ is to find a density function $P_a(\mathbf{X})$ from the set of available approximations such that $I(P, P_a)$ is minimized.

1.1 Approximation Techniques

We have pointed out earlier that it is impractical to gather the estimates for the joint density function $\mathrm{P}(\mathbf{X})$, for all values of $\mathbf{X}$. Such a task would involve maintaining estimates of the **highest** order marginals, for these marginals will completely determine the lower-order marginals. Thus from a feasible and practical point of view, we are restricted to collecting information about the lower-order marginals (i.e. the joint density functions for a subset of the N features).

The basic method adopted in all the approximation methods is as follows: If we have all marginals of the k^{th} order available, the approximation $P_a(\mathbf{X})$ will be chosen so that its marginals of order k (and less) agree with the *estimates* of the corresponding marginals of the underlying density function $P(\mathbf{X})$. Among all such functions which satisfy the marginals constraint, the best approximation is the one that minimizes the closeness measure defined in (1). It is natural to expect that the approximation gets better with the availability of the estimates of higher-order marginals.

The first solution to this problem of approximating discrete distributions was presented by Chow *et. al.* in [2]. In this method, the approximation uses only the information gathered about the first and second-order marginals. This constraint naturally leads to an approximation which is based on the *Chain Rule*, with the modification that the conditional probabilities utilized in the approximation are of the type $Pr(x_i \mid x_j)$. This can concep-

tually be viewed as a spanning tree of a complete graph consisting of N nodes. A node x_i is said to be the parent of x_j, if the probability density function $Pr(x_j \mid x_i) \neq Pr(x_j)$. Approximations of this type will be called "tree-type approximations".

This method due to Chow *et. al.* derives a relation between the measure of closeness $I(P, P_a)$ defined in (1) and the measure of dependence between *all the pairs* of variables. This measure is the well known *Expected Mutual Information Measure*(EMIM) which is defined for all pairs of discrete-valued variables $\{x_i, x_j\}$ as:

$$I^*(x_i, x_j) = \sum_{x_i, x_j} Pr(x_i, x_j) \log \frac{Pr(x_i, x_j)}{Pr(x_i) Pr(x_j)} \tag{2}$$

The reader will recognize that this definition for the measure of dependence between variables is very similar to the closeness measure between two density functions (or distributions). In fact, $P(x_i, x_j)$ can be viewed as the underlying distribution, and the "approximating" distribution is the one which assumes the features x_i, x_j are statistically independent. The essential idea in [2] is then to compute EMIM for the $\binom{N}{2}$ pairs of variables, and then select the most dominant dependencies. It is shown later that this approximation can be related a tree called the *dependence tree.* A recently published result proves that the result of Chow *et. al.* yields an approximate dependence tree, which is nearly optimal, even if the criterion is to minimize the probability of misclassification [7]. This method finds immense application in all problems that include pattern classification as a sub-problem. In particular, the applications to the field of information retrieval can be found in [6].

In the second type of approximations proposed by Ku *et. al.* in [4], the dependence between the components of the feature vector are not restricted to a tree-type dependence model. Once this constraint is relaxed, a very general method results and the paper [4] describes an extremely interesting and fascinating iterative method to obtain the best approximation possible, with the available information. Since this problem is not the primary concern of our current study, the details of the iterative procedure are not included here.

The structure of the report is as follows. Section 2 describes the method presented in [2]. The current study focusses on the use of an alternative new measure, I_χ, used to quantify

the dependence between variables; Section 3 defines the new metric I_χ and derives some properties of this new metric. For a restricted class of probability distributions, the metrics I_χ and I^* are shown to equivalent (and thus optimal), as far as the problem of finding the dependence tree is concerned. While it is known that the new metric I_χ is non-optimal in the most-general case (i.e. for the case of arbitrary joint distributions), experimental results clearly demonstrate its excellent accuracy. These results point out that in cases when the underlying distribution is not based on a tree, the new metric generates a tree that is only marginally non-optimal. In the case when the dependence between variables is indeed of the tree type, the convergence to the "real" tree is observed in all experiments conducted. Experimental studies have provided evidence that the computation of the best tree based on the I_χ metric can be usually achieved in about 20–25% of the time required to perform the same operation, using the I^* metric. Section 4 presents the experimental results.

2 Dependence Tree Approximation

To introduce the method due to Chow *et. al.* [2], we use the following well established rule, *the Chain Rule*, which expresses the joint probability distribution in terms of conditional probabilities:

$$P(X) = Pr(x_1)Pr(x_2 \mid x_1)Pr(x_3 \mid x_1, x_2) \cdots Pr(x_N \mid x_1 \ldots x_{N-1}) \tag{3}$$

We notice that in this expression, each variable is conditioned on increasing number of other variables. In this context, it is important to point out that estimating the k^{th} term of this equation (i.e. a conditional probability term, in which a variable is conditioned on $k-1$ other variables), requires maintaining the estimates of all the k^{th} order marginals. If we restrict ourselves to computing only the (first and) second-order marginals during the "training" phase, we are assured that we can compute all the conditional probabilities of the form $Pr(x_i \mid x_j)$. Thus we explore the approximation that results if we ignore the conditioning on multiple variables, implying that we shall attempt to retain only dependencies on at most one variable. This leads use to the following approximation:

$$P_a(X) = \prod_{i=1}^{N} Pr(x_i \mid x_{j(i)}) \qquad \text{where} \qquad 0 \leq j(i) < i. \tag{4}$$

Notice that in this equation, each variable is dependent on exactly one variable that has appeared earlier in the equation. To include the case when $j(i) = 0$, we use the convention that x_0 is a dummy variable, which does not influence any other variable. With this understanding, $Pr(x_i \mid x_0)$ will be the same as $Pr(x_i)$.

In the above case, there is a "natural ordering" among the components $\{x_i \mid i = 1, 2, \ldots, N\}$ of the vector $\mathbf{X}$, such that the variable x_i was conditioned on $x_{j(i)}$, where $j(i) < i$. In the general case, the product approximation takes the form:

$$P_a(X) = \prod_{i=1}^{N} Pr(x_{m_i} \mid x_{m_{j(i)}}) \tag{5}$$

where, $m_1, m_2, \ldots, m_N$ is a permutation of the integers $\{1, 2, \ldots, N\}$.

The dependence assumed in the above equation can be given a graph theoretical interpretation. Consider a graph G, with N nodes, labelled as $\{x_1, x_2, \ldots, x_N\}$. In this graph, the edge $(x_{m_i}, x_{m_{j(i)}})$, represents the fact the variable x_{m_i} is (statistically) dependent on the variable $x_{m_{j(i)}}$. It is easy to see that G is indeed a tree, with the node x_1 as the root. This tree is completely defined by the permutation $(m_1, m_2, \ldots m_N)$ and the function $j(\cdot)$.

It is well known that there are $N^{(N-2)}$ spanning trees on a graph with N nodes. Also, each tree is associated with a unique approximation of the form given in (5). The problem of finding the "dependence tree" is one of finding the tree, for which the associated approximation is the best.

It is proved in [2] that the closeness measure, $I(P, P_a)$, computed for the product approximation in (5) can be expressed as:-

$$I(P, P_a) = -\sum_{i=1}^{N} I^*(x_{m_i}, x_{m_{j(i)}}) + \sum_{i=1}^{N} H(x_i) - H(\mathbf{P}) \tag{6}$$

where,

$$H(x_i) = -\sum_{x_i} Pr(x_i) \log Pr(x_i)$$

$$H(\mathbf{P}) = -\sum_{\mathbf{X}} P(\mathbf{X}) log P(\mathbf{X})$$

and

$$I^*(x_i, x_j) = \sum_{x_i, x_j} Pr(x_i, x_j) \log \frac{Pr(x_i, x_j)}{Pr(x_i) Pr(x_j)}$$

The problem of finding the dependence tree, is one of finding the permutation $(m_1, m_2, \ldots, m_N)$ and the function $j(\cdot)$ (or equivalently an array $(j_1, j_2, \ldots j_N)$) which will minimize the right-hand side of (6). Clearly $\sum_{i=1}^{N} H(x_i)$ and H(P) are independent of the approximation since they only depend on the underlying distribution $P(\mathbf{X})$. Therefore, in order to minimize the RHS of (6), we must maximize $\sum_{i=1}^{N} I^*(x_{m_i}, x_{m_{j(i)}})$.

It can be seen that this is essentially the same as the operation of finding the maximum spanning tree of the graph G, with N nodes, $x_1, x_2, \ldots, x_N$, where the edge between nodes x_i, x_j is assigned the weight $I(x_i, x_j)$. In practice, the probabilities required for computing the edge weights of the graph are not known *a priori*, they must be estimated from the samples. The process of finding the best dependence-tree can be algorithmically described as follows:

Algorithm Chow

Input: The set of s samples $X^1, X^2, \cdots, X^s$.

Output: The best dependence-tree τ as per the EMIM metric.

Method:

1. Estimate the first and second-order marginals from the various samples.
2. Compute the edge weights $I^*(x_i, x_j)$ for all pairs of nodes, and create the graph G.
3. Compute the Maximum Spanning Tree of G and submit it as the desired tree τ^*.

End Algorithm Chow

It is proved in [2] that **Algorithm Chow**, indeed finds the maximum likelihood estimate of the dependence tree. The significance of this result is tremendous, and although the result was proved in [2], we believe that the power of the result was not adequately stressed. Notice that from the set of all spanning trees of the graph G (totally N^{N-2} of them), the maximum likelihood estimate of the tree is the one which maximizes the likelihood of generating the actual occurence of samples. Thus to obtain such an estimate, the

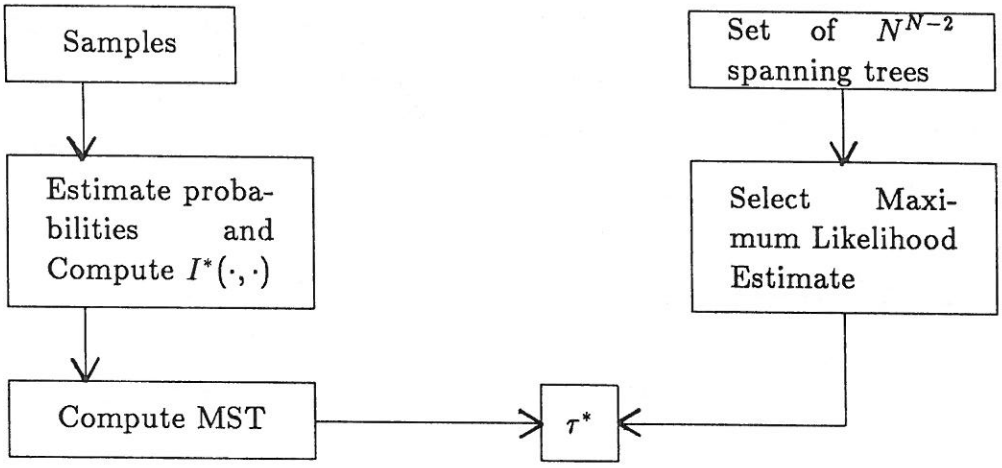


Figure 1: Finding the maximum likelihood tree

brute force *modus operandus* would suggest that the likelihood function be evaluated for every possible spanning tree, and then the tree which maximized this function would be the reported solution. Considering the large number of spanning trees, this is computationally infeasible except for graphs with a small number of nodes. **Algorithm Chow** is an elegant solution to this problem – it merely involves the straightforward computation of the MST; this being a computation of complexity $O(N^2)$. This result, is stated formally in **Theorem 0** and is illustrated pictorially in Figure 1.

Theorem 0

The dependence-tree produced by **Algorithm Chow** is the maximum likelihood tree, which can be obtained from the samples $X^1, X^2, \ldots, X^s$.

Proof:

A sketch of the proof of this result, was presented in [2]. A more detailed and complete proof can be found in [5]. □

In the light of the above result, it is clear that the metric defined in (2) is the best possible metric for capturing dependence information between *pairs* of variables, if the overall approximation is to be ideal in the sense of obtaining the minimum value of $I(P, P_a)$, as defined in (1). The only drawback with the the I^* measure is that it is computationally

very expensive and time consuming (especially for large values of N), as it involves the evaluation of $O(N^2k^2)$ logarithms where, as before, k is the number of values which each feature of the sample can take. We shall now present a new metric, called the I_χ metric, which essentially captures the χ^2-statistic of the distributions and which can be used as a measure of dependence between the variables.

3 The χ^2 statistic

We define the metric $I_\chi(x_i, x_j)$ to quantify the degree of dependence between two discrete (random) variables as follows:-

$$I_\chi(x_i, x_j) = \sum_{x_i, x_j} \frac{(Pr(x_i, x_j) - Pr(x_i)Pr(x_j))^2}{Pr(x_i)Pr(x_j)} \tag{7}$$

From the definition in (7), it is clear that I_χ has the following desirable characteristics of a metric capturing dependency information:

1. $I_\chi(x_i, x_j) \geq 0$
2. $I_\chi(x_i, x_j) = 0$ iff $Pr(x_i, x_j) = Pr(x_i)Pr(x_j)$.

Observe that the latter equation represents the case when the variables x_i, x_j are *statistically independent.*

At this juncture, it is worth recapitulating that Section 2 concluded with the very important result due to Chow *et. al.* which is that the I^* metric defined in (2) naturally appears in the process of selecting the maximum likelihood estimate of the dependence tree. This is true, since we are constrained by the laws of probabilities, to choose a product form approximation for the joint distribution $P(\mathbf{X})$, and the only metric which minimizes the closeness measure defined in (1) is the EMIM given in (2).

In this light, it is clear that the metric defined in (7), cannot be related to the closeness of approximation defined by (1). Furthermore, it does not seem to be possible to relate this metric to an analogously defined closeness of approximation metric obtained if (7) was generalized for the case of distributions, as shown below:-

$$I_\chi(P, P_a) = \sum_{\mathbf{X}} \frac{(P(\mathbf{X}) - P_a(\mathbf{X}))^2}{P_a(\mathbf{X})}$$

Although $I_\chi(x_i, x_j)$ cannot be related to $I(P, P_a)$ defined in (1), it is not entirely futile to pursue it. Observe that for the first part, it is far easier to compute it rather than to compute I^* because as pointed out earlier, the latter involves the evaluation of numerous logarithms. Thus although we cannot prove the optimality of I_χ, for all distributions, this metric is not entirely void of an underlying theoretical framework. Indeed we can prove the following properties of I_χ:

1. In the case when the features x_i, x_j are binary valued, the metric $I_\chi(x_i, x_j)$ defined in (7), increases or decreases monotonically with $I^*(x_i, x_j)$.

2. For a restricted class of probability distributions, i.e. for a specific type of dependence between the individual features, the quantities $I_\chi(x_i, x_j)$ and $I^*(x_i, x_j)$ are shown to be equivalent. Thus within this family, the I_χ metric indeed yields the Maximum likelihood estimate of the underlying tree, even though the computation is achieved without computing the matrix $I^*(\cdot, \cdot)$.

Although Property 1 has been proved for the case of binary-valued features, we believe that the same result holds even if the restriction of having binary-valued features is relaxed. We now prove the above results formally.

Theorem 1

If x_i, x_j are binary-valued distinct features, $I_\chi(x_i, x_j)$ increases(decreases) iff $I^*(x_i, x_j)$ increases (decreases).

Proof:

We note that both I_χ and I^* are functions of the four joint probabilities $Pr(x_i = \alpha, x_j = \beta)$, for $\alpha, \beta = 0, 1$. For the sake of brevity, we denote $p_{\alpha\beta} = Pr(x_i = \alpha, x_j = \beta)$. Since $p_{00}+p_{01}+p_{10}+p_{11} = 1$, all the four $p_{\alpha\beta}$'s cannot be independently specified. Indeed, since any three of them automatically determine the fourth, in the following derivations, we assume p_{00}, p_{01}, p_{10} to be the *linearly independent* parameters, and that p_{11} is implicitly specified in terms of their values. We also note that the first order marginals for x_i and x_j can be easily expressed in terms of these second-order marginals.

To do this, we introduce the following notation: $a_\alpha = Pr(x_i = \alpha)$ and $b_\alpha = Pr(x_j = \alpha)$. It is easy to see that:

$$\begin{aligned} a_0 &= p_{00} + p_{01} \\ a_1 &= p_{10} + p_{11}, \\ b_0 &= p_{00} + p_{10}, \\ b_1 &= p_{01} + p_{11}. \end{aligned} \tag{8}$$

Now, by the definition of I^* in (2),

$$\begin{aligned} I^*(x_i, x_j) &= \sum_{i,j} p_{ij} \log \frac{p_{ij}}{a_i b_j} \\ &= p_{00} \log p_{00} - p_{00} \log a_0 b_0 + p_{01} \log p_{01} - p_{01} \log a_0 b_1 \\ &\quad + p_{10} \log p_{10} - p_{10} \log a_1 b_0 + p_{11} \log p_{11} - p_{11} \log a_1 b_1. \end{aligned}$$

Observe that as a consequence of (8), $\frac{\partial a_0}{\partial p_{00}} = \frac{\partial b_0}{\partial p_{00}} = 1$ and $\frac{\partial a_1}{\partial p_{00}} = \frac{\partial b_1}{\partial p_{00}} = -1$. Thus, taking partial derivatives with respect to p_{00}, we get,

$$\begin{aligned} \frac{\partial I^*}{\partial p_{00}} &= (\log p_{00} + 1) - (\log(a_0 b_0) + \frac{p_{00}}{a_0 b_0}(a_0 + b_0)) - \frac{p_{01}}{a_0 b_1}(b_1 - a_0) - \frac{p_{10}}{a_1 b_0}(a_1 - b_0) \\ &\quad -(\log p_{11} + 1) - (-\log(a_1 b_1) + \frac{p_{11}}{a_1 b_1}(-a_1 - b_1)). \end{aligned}$$

Simplifying the above equation,

$$\begin{aligned} \frac{\partial I^*}{\partial p_{00}} &= \log \frac{p_{00}/a_0 b_0}{p_{11}/a_1 b_1} + \frac{p_{10} + p_{11}}{a_1} + \frac{p_{01} + p_{11}}{b_1} - \frac{p_{00} + p_{01}}{a_0} - \frac{p_{00} + p_{10}}{b_0} \\ &= \log \frac{p_{00}/a_0 b_0}{p_{11}/a_1 b_1}. \end{aligned} \tag{9}$$

We now derive an analogous expression for the derivative for I_χ. Consider,

$$\begin{aligned} I_\chi(x_i, x_j) &= \sum_{i,j} \frac{(p_{ij} - a_i b_j)^2}{a_i b_j} \\ &= \sum_{i,j} \frac{p_{ij}^2}{a_i b_j} - \sum_{i,j} 2p_{ij} + \sum_{i,j} a_i b_j \\ &= \sum_{i,j} \frac{p_{ij}^2}{a_i b_j} - 1. \end{aligned} \tag{10}$$

Using the simplified form of I_χ in (10), and taking the partial derivative w.r.t. p_{00}, we get,

$$\begin{aligned}\frac{\partial I_\chi}{\partial p_{00}} &= 2\frac{p_{00}}{a_0b_0} - \left(\frac{p_{00}}{a_0b_0}\right)^2(a_0+b_0) \\ &\quad - \left(\frac{p_{01}}{a_0b_1}\right)^2(b_1-a_0) - \left(\frac{p_{10}}{a_1b_0}\right)^2(a_1-b_0) \\ &\quad -2\frac{p_{11}}{a_1b_1} - \left(\frac{p_{11}}{a_1b_1}\right)^2(-a_1-b_1).\end{aligned}$$

Grouping terms and performing some elementary algebraic simplifications, we obtain,

$$\begin{aligned}\frac{\partial I_\chi}{\partial p_{00}} &= 2\left(\frac{p_{00}}{a_0b_0} - \frac{p_{11}}{a_1b_1}\right) - (a_1-b_0)\left\{\left(\frac{p_{01}}{a_0b_1}\right)^2 + \left(\frac{p_{10}}{a_1b_0}\right)^2\right\} \\ &\quad +2\left(\frac{p_{11}}{a_1b_1}\right)^2 - (a_0+b_0)\left\{\left(\frac{p_{00}}{a_0b_0}\right)^2 + \left(\frac{p_{11}}{a_1b_1}\right)^2\right\}.\end{aligned}$$

To simplify the above expression, we define r_{ij} as:

$$r_{ij} = \frac{p_{ij}}{a_ib_j}.$$

Thus the above equation (for the derivative) can be conveniently rewritten as:

$$\frac{\partial I_\chi}{\partial p_{00}} = 2(r_{00}-r_{11}) - (a_1-b_0)(r_{01}^2+r_{10}^2) - (a_0+b_0)(r_{00}^2+r_{11}^2) + 2r_{11}^2. \tag{11}$$

We shall now derive certain auxiliary equations, relating the $\{r_{ij}\}$'s. Consider the difference $r_{00}-r_{11}$. Indeed, this has the value,

$$\begin{aligned}r_{00}-r_{11} &= \frac{p_{00}}{a_0b_0} - \frac{p_{11}}{a_1b_1} = \frac{p_{00}a_1b_1 - p_{11}a_0b_0}{a_0a_1b_0b_1} \\ &= \frac{p_{00}\{p_{11}^2 + (p_{01}+p_{10})p_{11} + p_{01}p_{10}\} - p_{11}\{p_{00}^2 + (p_{01}+p_{10})p_{00} + p_{01}p_{10}\}}{a_0a_1b_0b_1} \\ &= \frac{(p_{11}-p_{00})D}{a_0a_1b_0b_1} \quad \text{where } D = p_{00}p_{11} - p_{01}p_{10}.\end{aligned} \tag{12}$$

In a similar manner, we can obtain the following equations for $r_{00}-r_{01}$ as:

$$\begin{aligned}r_{00}-r_{01} &= \frac{p_{00}}{a_0b_0} - \frac{p_{01}}{a_0b_1} = \frac{p_{00}b_1 - p_{01}b_0}{a_0b_0b_1} \\ &= \frac{p_{00}(p_{11}+p_{01}) - p_{01}(p_{10}+p_{00})}{a_0b_0b_1} \\ &= D/a_0b_0b_1\end{aligned}$$

Using the symmetry of the situation, the rest of the auxiliary relations between $\{r_{ij}\}$'s can be written down as:

$$\begin{aligned} r_{00} - r_{10} &= D/(a_0 a_1 b_0) \\ r_{11} - r_{01} &= D/(a_0 a_1 b_1) \\ r_{11} - r_{10} &= D/(a_1 b_0 b_1) \end{aligned} \tag{13}$$

We note that $a_0 + b_0 = 1 - (a_1 - b_0)$, and hence we can rewrite the expression in (11) for $\partial I_\chi / \partial p_{00}$ as:-

$$\begin{aligned} \frac{\partial I_\chi}{\partial p_{00}} &= 2(r_{00} - r_{11}) - (a_1 - b_0)(r_{01}^2 + r_{10}^2) + (a_1 - b_0)(r_{00}^2 + r_{11}^2) - (r_{00}^2 + r_{11}^2) + 2r_{11}^2 \\ &= (r_{00} - r_{11})(2 - r_{00} - r_{11}) + (a_1 - b_0)(r_{00}^2 + r_{11}^2 - r_{01}^2 - r_{10}^2) \\ &= (r_{00} - r_{11})(2 - r_{00} - r_{11}) + (p_{11} - p_{00}) \left\{ \frac{D}{a_0 b_0 b_1}(r_{00} + r_{01}) + \frac{D}{a_1 b_0 b_1}(r_{11} + r_{10}) \right\} \\ &= (r_{00} - r_{11})(2 - r_{00} - r_{11}) + \frac{D(p_{11} - p_{00})}{a_0 a_1 b_0 b_1} \{a_1(r_{00} + r_{01}) + a_0(r_{11} + r_{10})\} \\ &= (r_{00} - r_{11}) \{2 - r_{00} - r_{11} + a_1(r_{00} + r_{01}) + a_0(r_{11} + r_{10})\} \\ &= (r_{00} - r_{11}) \{2 + a_1(r_{01} - r_{11}) + a_0(r_{10} - r_{00})\}. \end{aligned} \tag{14}$$

Consider the second term of the RHS of (14). We can simplify this using the values of $\{r_{ij}\}$ and the relation between the $\{r_{ij}\}$ (13) to yield:

$$\begin{aligned} 2 + a_1(r_{01} - r_{11}) + a_0(r_{10} - r_{00}) &= 2 - \frac{a_1 D}{a_0 a_1 b_1} - \frac{a_0 D}{a_0 a_1 b_0} \\ &= 2 - \frac{D}{a_0 b_1} - \frac{D}{a_1 b_0} \\ &= (1 - \frac{D}{a_0 b_1}) + (1 - \frac{D}{a_1 b_0}). \end{aligned} \tag{15}$$

Also since,

$$\begin{aligned} a_0 b_1 &= (p_{00} + p_{01})(p_{01} + p_{11}) \\ &= p_{01}^2 + (p_{00} + p_{11}) p_{01} + p_{00} p_{11} \end{aligned}$$

it is clear that $a_0 b_1 - D = p_{01}^2 + (p_{00} + p_{11}) p_{01} + p_{01} p_{10} = p_{01}$.

Hence, combining (11) and (15), we get:

$$\frac{\partial I_\chi}{\partial p_{00}} = (r_{00} - r_{11}) \left\{ (1 - \frac{D}{a_0 b_1}) + (1 - \frac{D}{a_1 b_0}) \right\}$$

$$= (r_{00} - r_{11}) \left\{ \frac{p_{01}}{a_0 b_1} + \frac{p_{10}}{a_1 b_0} \right\}$$
$$= (r_{00} - r_{11})(r_{01} + r_{10}). \tag{16}$$

Comparing equations (9) and (16), we see that for all distinct random variables, x_i, x_j, $\partial I_\chi / \partial p_{00}$ follows $\partial I^* / \partial p_{00}$ in sign everywhere and the theorem is proved with regard to the variation of the quantities w.r.t. p_{00}. Similar expressions can be derived for the partial derivatives of I_χ and I^*, w.r.t. the variables p_{01} and p_{10}. In the interest of brevity, these are not derived here, but the results are stated below:-

$$\begin{aligned}
\frac{\partial I_\chi}{\partial p_{01}} &= (r_{01} - r_{11})(r_{10} + r_{00}) \\
\frac{\partial I_\chi}{\partial p_{10}} &= (r_{10} - r_{11})(r_{01} + r_{00}) \\
\frac{\partial I^*}{\partial p_{01}} &= \log \frac{r_{01}}{r_{11}} \\
\frac{\partial I^*}{\partial p_{10}} &= \log \frac{r_{10}}{r_{11}}.
\end{aligned} \tag{17}$$

These expressions can be derived, by following the above derivation almost synchronously. An inspection of the equations in (17) reveals that the partial derivatives of I_χ and I^* w.r.t. p_{01} and p_{10} have the same sign. Hence the Theorem. □

Remark:

A comparison of equations (9) and (16), indicates the equality of sign, only if r_{01} and r_{10} are *both* non-zero. It is interesting to note the implication of this condition. Observe that if r_{01} and r_{10} are both zero, the definition of r_{ij} implies that $p_{01} = p_{10} = 0$, and that the values of x_i, x_j are closely related (indeed, they are *never* different). In this case the condition $r_{01} = r_{10} = 0$, is equivalent to the condition that x_i is *identically equal* to x_j and thus x_i and x_j are non-distinct.

Furthermore, an observation of (17) indicates that the derivatives of I_χ and I^* w.r.t. p_{01} agree in sign everywhere, except when $r_{10} = r_{00} = 0$. This condition implies that $p_{10} = p_{00} = 0$, or that $b_0 = 0$. The condition $b_0 = 0$ implies that the feature x_j is always 1, and hence is not informative. Such features can therefore be assumed to be absent. Similar arguments can be made with regard to the partial derivative w.r.t. p_{10}. In summary, except in trivial cases, the partial derivates of I_χ and I^* agree in sign.

The result established in Theorem 1 implies that if, as a result of slightly perturbing the second-order marginals of the variables x_i and x_j, the measure $I^*(x_i, x_j)$ increases or decreases, a similar effect will be observed in the measure $I_\chi(x_i, x_j)$. Though the result is all-encompassing and powerful, it is unfortunately quite a local result. Thus, although the theorem guarantees how the weight assigned to an edge will vary depending on the metric used, it does not guarantee how the weight of the MST will vary as a function of the metric used. In other words, this property does not ensure that these two metrics are equivalent when it concerns computing the MST. In order to guarantee that both measures of dependencies find the same dependency tree, the relative ordering of the weights for all the edges meeting at a particular node must be preserved, irrespective of the metric used (i.e. I_χ or I^*).

Although I^* and I_χ only locally follow one another for all distributions, there is a subset of distributions for which there is a global relationship between the two. For the first part note the I^* itself yields the optimal tree only if we restrict ourselves to second-order marginals. If additionally we constrain ourselves to distributions in which the second-order marginals satisfy certain elementary constraints, we can show that I^* and I_χ globally follow each other. Thus working with a restricted set of distributions, it becomes evident that the MST chosen using the I_χ metric is **exactly** the Maximum Likelihood estimate for the underlying tree. The following sequence of Lemmas prove the result.

Lemma 1

Consider the pair of variables x_i, x_j and the joint probabilities defined as follows:

$$\begin{aligned} p_{00} &= a - \lambda \\ p_{01} &= b + \lambda \\ p_{10} &= c + \lambda \\ p_{11} &= d - \lambda \end{aligned} \tag{18}$$

where a, b, c, d and λ satisfy:

$$a + b + c + d = 1$$

$$\begin{aligned} ad &= bc \\ 0 &\leq \lambda \leq \min(a, d) \end{aligned} \tag{19}$$

Then $dI^*/d\lambda$ follows $dI_\chi/d\lambda$ in sign.

Proof:

We note that from (10):

$$I_\chi(x_i, x_j) = \sum_{i,j} \frac{p_{ij}^2}{a_i b_j} - 1.$$

Now, the total derivative of I_χ can be written as:

$$\begin{aligned} \frac{dI_\chi}{d\lambda} &= \frac{2p_{00}(-1)}{a_0 b_0} + \frac{2p_{01}}{a_0 b_1} + \frac{2p_{10}}{a_1 b_0} + \frac{2p_{11}(-1)}{a_1 b_1} \\ &= 2\frac{(\lambda - a)b_1 + (b + \lambda)b_0}{a_0 b_0 b_1} + 2\frac{(c + \lambda)b_1 + (\lambda - d)b_0}{a_1 b_0 b_1} \end{aligned} \tag{20}$$

The numerator of the first term of the RHS of (20) is:

$$\begin{aligned} (\lambda - a)b_1 + (b + \lambda)b_0 &= \lambda(b_0 + b_1) - a(b + d) + b(c + a) \\ &= \lambda - (ad - bc) = \lambda, \end{aligned}$$

the last equality being a consequence of (19).

Similarly, the numerator of the second term of the RHS of (20) is:

$$(\lambda - d)b_0 + (c + \lambda)b_1 = \lambda - d(c + a) + c(b + d) = \lambda$$

Thus,

$$\begin{aligned} \frac{dI_\chi}{d\lambda} &= \frac{2\lambda}{a_0 b_0 b_1} + \frac{2\lambda}{a_1 b_0 b_1} = \frac{2\lambda(a_0 + a_1)}{a_0 a_1 b_0 b_1} \\ &= \frac{2\lambda}{a_0 a_1 b_0 b_1}, \end{aligned} \tag{21}$$

where again the last equality follows since $a_0 + a_1 = 1$.

We now proceed to evaluate the derivative of I^* w.r.t. λ. Since,

$$\begin{aligned} I^*(x_i, x_j) &= \sum_{i,j} p_{ij} \log \frac{p_{ij}}{a_i b_j} \\ &= \sum p_{ij} \log p_{ij} - \sum p_{ij} \log K_{ij}, \quad \text{where } K_{ij} = a_i b_j, \end{aligned}$$

we have

$$\begin{aligned}\frac{dI^*}{d\lambda} &= (1+\log p_{00})(-1)+(1+\log p_{01})+(1+\log p_{10})+(1+\log p_{11})(-1)\\ &\quad +\log K_{00}+\log K_{11}-\log K_{01}-\log K_{10}\\ &= \log\frac{p_{01}p_{10}}{p_{00}p_{11}}+\log\frac{K_{00}K_{11}}{K_{01}K_{10}}.\end{aligned}$$

Now it is easy to verify that $K_{00}K_{11}/K_{01}K_{10} = 1$. Hence the second term of the equation for $dI^*/d\lambda$ vanishes. Thus,

$$\begin{aligned}\frac{dI^*}{d\lambda} &= \log\frac{p_{01}p_{10}}{p_{00}p_{11}}\\ &= \log\frac{(b+\lambda)(c+\lambda)}{(a-\lambda)(d-\lambda)}\\ &= \log\frac{\lambda^2+(b+c)\lambda+bc}{\lambda^2-(a+d)\lambda+ad}\\ &= \log\frac{\lambda^2+(1-\alpha)\lambda+\beta}{\lambda^2-\alpha\lambda+\beta} \quad \text{where} \quad \alpha=a+d, \quad \text{and}\,\beta=ad \qquad (22)\end{aligned}$$

Comparing (21) and (22), we see that both $dI_\chi/d\lambda$ and $dI^*/d\lambda$ are positive, negative or zero, according as λ is positive, negative, or zero respectively. Hence we have proved Lemma 1. □

Lemma 2

Let x_i be any feature. Then for all $j \neq i$, let the joint distribution of the pair of features x_i, x_j satisfy the conditions (18) and (19) of Lemma 1 with the additional constraint that all distributions are based on common values of the parameters a, b, c, d but differ only in the value of the parameter λ. Then the index k which maximizes $I^*(x_i, x_j)$ also maximizes $I_\chi(x_i, x_j)$.

Proof:

Since all the edges have common values of the parameters a, b, c, d, but differ only in the value of λ, the edge weights (under I^* and I_χ metrics) are functions of the same single parameter λ. The stated result simply follows from Lemma 1 as per the observation that both I^* and I_χ are both increasing functions of λ in the interval that λ is constrained to belong to. This implies that the index of the edge with maximum weight remains unchanged under both metrics. □

Theorem 2

If the joint distributions for every pair of variables x_i, x_j have the same parametric form defined by (18) and (19) with identical values of a, b, c, d, but the distributions differ only in the parameter λ, then the MST chosen by I_χ will be **exactly identical** to that chosen by I^* or is just as good an approximation as the latter.

Proof:

Lemma 2 has established the result that of the edges meeting at a particular node, the index of edge with maximum weight is invariant under both the metrics I^* and I_χ. Given the additional information that **all edges** are specified using the same parametric constants (i.e. a, b, c, d), a straightforward extension of Lemma 1 yields the result that the **relative ordering** among the $\binom{N}{2}$ $I^*(\cdot,\cdot)$ weights and $I_\chi(\cdot,\cdot)$ weights are identical.

Several algorithms to solve the MST problem are known. We shall show the equivalence of the trees produced by I^* and I_χ metrics, by simulating Kruskal's algorithm [1]. This algorithm essentially sorts the edges in the descending order of weights and selects the (N-1) edges that do not form cycles. It is apparent that since the relative ordering of the edge weights under the two metrics are identical, the MST computed will be identical.

It must be noted however that the MST is unique only if all the edge weights are distinct. If two edges have identical weights under I^* metric, they can be easily shown to have identical weights under the I_χ metric. Thus in these cases, the MST's produced by the two metrics can only differ in the selection of an edge from the set of edges with equal weight. The sum of the weights of the edges included in the MST will indeed be identical and hence both the MSTs will be equally good approximations (see (6)). □

It is interesting to note that the decision to specify all the $\binom{N}{2}$ distributions identically, has other implications. These conditions are indicated below:

Lemma 3

If all the $\binom{N}{2}$ pairs of variables in the graph G use the same parameters a, b, c, d, then $b = c$ and the values of a, d are given by: $\{(1-2b) \pm \sqrt{1-4b}\}/2$.

Proof:

Let us consider a node i and its neighbour j. Then $Pr(x_i) = 0$ and $Pr(x_j) = 0$ are given by $a+b$ and $a+c$ respectively. Because of the symmetry, we must have $a+b = a+c$, and hence $b = c$. Given the additional conditions $a + b + c + d = 1$ and $ad = bc$, we can solve for the two unknowns (i.e. a and d) from these two equations. The above equation indicates that $b \leq 0.25$. When $b = 0.25$, all the constants a, b, c, d are equal (to 0.25). □

4 Experimental Results

In order to test the validity of the theoretical results presented in this paper, numerous simulations were carried out. The experiments conducted were of two categories. In the first set of experiments the intention was primarily to test the accuracy of the new metric I_χ. That is, the aim was to determine the number of times the MST τ_χ (obtained by using I_χ) is identical to the corresponding tree τ^* (derived by I^*). In the second set of experiments, the aim was to observe how quickly the τ_χ and τ^* converged to the "real" underlying tree.

As a consequence of Theorems 1 and 2, it is easy to see that whenever the conditions of (18) and (19) are satisfied and the feature distributions have identical parameters, both the I_χ and the I^* metric will be equally accurate. Of course this has been experimentally verified and in **every** single case, when the conditions were satisfied, τ_χ and τ^* were either identical or equally efficient w.r.t. the EMIM metric I^* where by "equally efficient" we mean that the sum of the I^* weights for all the edges in τ_χ and τ^* are identical. Thus any further simulations done for this scenario can only further justify the assertion.

The quality of the estimate τ_χ can be evaluated for the case when the conditions of Theorem 2 are not satisfied. Let us suppose that some (or all) pairs of features x_i, x_j has a second-order marginal distribution $Pr(x_i, x_j)$, which does not satisfy these conditions. Then the estimate τ_χ need not be identical to (or equally efficient as) τ^*. However we observe that for a good proportion of time, τ_χ is **identical** to τ^* and in the cases when they are not identical, the relative difference between the weights of τ_χ and τ^* (i.e. sum of the weights of all the edges) as per the EMIM metric I^* is extremely small — being of

the order of 0.5 %.

These experiments were conducted for various dimensions of the feature vector, as follows. First the number of features was selected and each second-order marginal distribution was randomly generated using the procedure explained below.

The procedure is easy to follow, if we consider the following sub-problem. Let us limit ourselves to dealing with two events A and B, which occur with probabilities $Pr(A)$ and $Pr(B)$ respectively. The problem which we wish to address is one of randomly assigning a probability to the joint event $A \cap B$.

In order to assign probabilities to the event $A \cap B$, we have to first establish some bounds for this quantity. Clearly $0 \leq Pr(A \cap B) \leq \min(Pr(A), Pr(B))$. Furthermore, since,

$$Pr(A \cup B) = Pr(A) + Pr(B) - Pr(A \cap B)$$

and $Pr(A \cup B) \leq 1$,

we easily get the following inequalities:

$$\begin{aligned} Pr(A \cap B) &= Pr(A) + Pr(B) - Pr(A \cup B) \\ &\geq Pr(A) + Pr(B) - 1. \end{aligned}$$

Combining the above results, we arrive at the following relation:

$$\max(0, Pr(A) + Pr(B) - 1) \leq Pr(A \cap B) \leq \min(Pr(A), Pr(B)). \tag{23}$$

Given the lower and upper bounds established by this equation, it is easy to randomly assign a value to $Pr(A \cap B)$.

We now illustrate the use of the above result, in assigning the second-order marginal distributions for the pair of features x_i and x_j. For simplicity, we assume that these features are binary valued. The first step in this process is to assign the first-order marginals to the features x_i and x_j. In other words we first randomly assign values to $Pr(x_i = 0)$ and $Pr(x_j = 0)$, and thus implicitly assign values to $Pr(x_i = 1)$ and $Pr(x_j = 1)$. Using the bounds specified by (23), we can now randomly assign a value to the joint probability

$Pr(x_i = 0, x_j = 0)$. Once this choice has been made, the remaining second-order marginals can be easily written down as follows:

$$\begin{aligned} Pr(x_i = 0, x_j = 1) &= Pr(x_i = 0) - Pr(x_i = 0, x_j = 0), \\ Pr(x_i = 1, x_j = 0) &= Pr(x_j = 0) - Pr(x_i = 0, x_j = 0), \\ Pr(x_i = 1, x_j = 1) &= Pr(x_i = 1) - Pr(x_i = 0, x_j = 1). \end{aligned}$$

To get the joint distribution for all the features, the above procedure was repeated for all pairwise combinations of features.

We are now ready to present the results obtained for the above mentioned case. The results represent the ensemble average, computed over 5000 random distributions, for each value of the dimension D of the feature vector. In the following discussion EMIM(τ) stands for $\sum_{e \in \tau} I^*(e)$. The two parameters we report are (i) the the number of times τ^* and τ_χ are different, and (ii) the Relative Error between EMIM(τ^*) and EMIM(τ_χ). This quantity is computed as the percentage average of $(EMIM(\tau^*) - EMIM(\tau_\chi))/EMIM(\tau^*)$. Table 1 presents our results. The results are significant. For example, when D the number of dimensions is 6, the percentage error between the EMIM values of τ^* and τ_χ is as low as 0.277%. This value increases to 0.49 %, when the dimension D increases to 12.

Similar experiments were conducted in the case when the features were not binary valued, but were ternary valued (3-valued). Table 2 presents the results for this case. Notice that although the number of mismatches is relatively large, the percentage error between the weights of two trees τ^* and τ_χ is small.

In the second set of experiments, instead of generating the probability distributions directly, we used an underlying tree to generate the samples. Given the dependence tree, the program must generate the various features of the vector in such a way that the "parent" feature is assigned a value before the "child" (or dependent) feature. It is easy to see that this condition is satisfied if the features are assigned values in the order that a breadth-first traversal of the dependence tree would visit them. The estimates of the

Dimension of Feature Vector (D)	Proportion of Mismatches (in %)	Average Relative Error (in %)
6	22.12	0.2771
8	39.02	0.3799
10	52.28	0.4470
12	62.58	0.4912

Table 1: This table summarizes the results of comparing the trees produced by the I_χ and I^* metrics, when random distributions were used. The reported results are the number of mismatches and the accuracy of the tree weights. In each case, the features are binary valued.

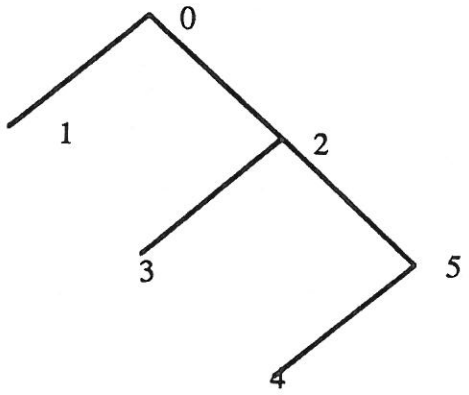
Dimension of Feature Vector (D)	Proportion of Mismatches (in %)	Average Relative Error (in %)
6	46.34	1.2941
8	65.62	1.6487
10	79.54	1.8448

Table 2: This table summarizes the results of comparing the trees produced by the I_χ and I^* metrics, when random distributions were used. The reported results are the number of mismatches and the accuracy of the tree weights. In each case (i.e. for each value of D), each of the features take one of 3 values.

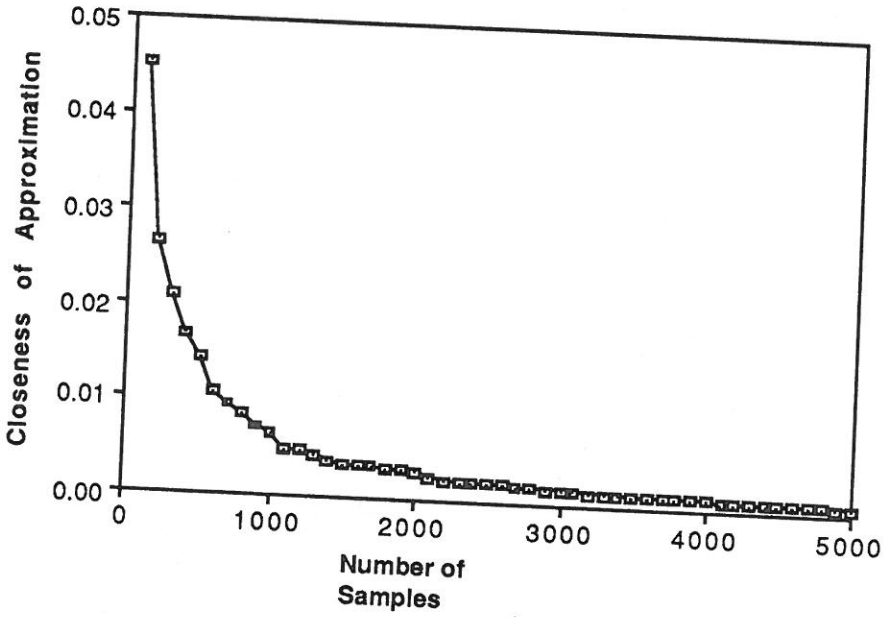
second-order (and the first-order) marginals were updated with every sample. Subject to an initial training phase, the trees τ_χ and τ^* were computed as the samples arrived. As before, we maintain the number of times the trees τ^* and τ_χ were different. Since the dependence in this case comes from an **unknown underlying tree** τ_r, it is useful to compare the convergence of τ^* and τ_χ to this tree τ_r. One measure of the closeness of these trees is the closeness of the distributions generated by them and this is precisely what we chose to monitor. If P_r, P^* and P_χ are the distributions derived from the "real" underlying tree τ_r, τ^* and τ_χ respectively, the quantities which we measured are the closeness measures $I(P_r, P_\chi)$ and $I(P_r, P^*)$. Note that these quantities can be easily computed from the definition given in (1).

We now present the results for the case when D, the number of dimensions, is 6. The underlying dependence tree used is shown in Figure 2a. In our experiments, the agreement between τ^* and τ_χ was so close that the plots of $I(P_r, P_\chi)$ and $I(P_r, P^*)$ vs. n, the number of samples, can hardly be differentiated visually. For this reason, in Figure 2b we have shown $I(P_r, P_\chi)$ as a function of n. The closeness of approximation in Figure 2b, was obtained as the ensemble average over 50 experiments, each consisting of 5000 samples. Besides the fact that $I(P_r, P_\chi)$ and $I(P_r, P^*)$ are extremely close, two other observations are also worth mentioning. When the underlying dependence is one of tree-type dependence, the agreement between τ^* and τ_χ is indeed very close. This is evidenced by a very small number of mismatches, when compared to the case of random distributions. The other observation is that in terms of the speed of convergence, both metrics seem to be comparable. For the case illustrated, the number of samples required to reduce the "initial distance" between the real tree and the estimate to 10% of its value, is 1000. It must however be kept in mind that even though the "real dependency" tree may have been found earlier, the closeness measure between these distributions is not exactly zero, due to **inaccurate estimates** of the conditional probabilities. Figure 3 reports similar results for the case when the dimensionality of the feature vector, D, is 10.

In order to quantify the ease of computing τ_χ over that of τ^*, we have performed some measurements on the execution times. The experiment consisted of generating a random

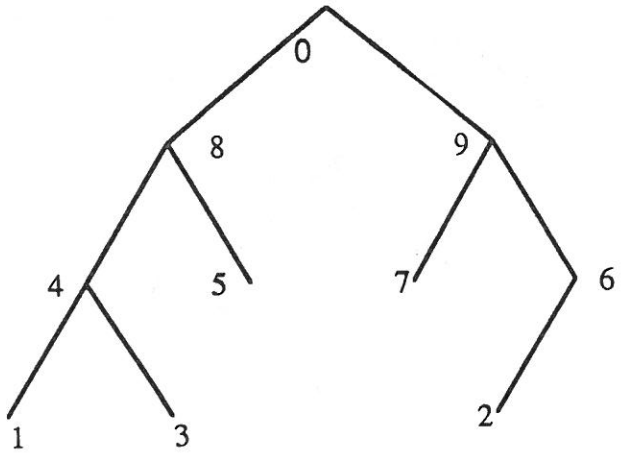


(a) Underlying dependence used for generating the samples to measure the rate of convergence of τ_χ. In this case, D, the dimensionality of the feature vector is 6, which is equal to the number of nodes in this tree.

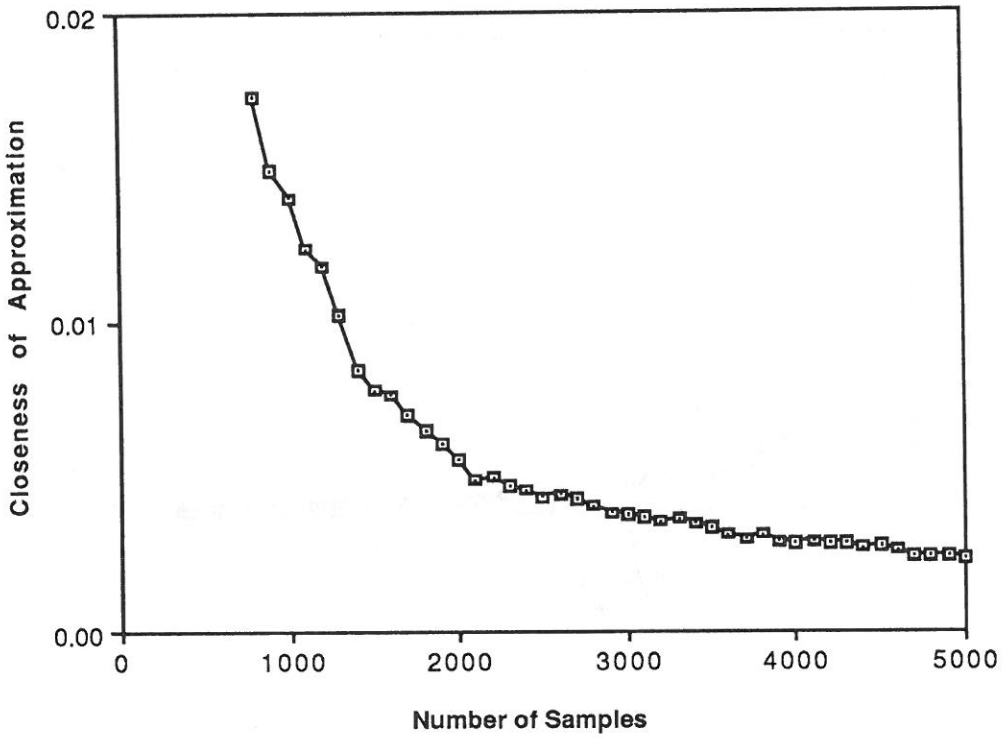


(b) Variation of the ensemble average of the closeness of approximation $I(P, P_a)$ between the actual distribution P and the estimated distribution P_a derived from the estimated dependence tree. The metric used to derive this tree is the I_χ metric. The underlying dependence tree describing P is given in Figure 2a.

Figure 2: Rate of Convergence (D=6)



(a) Underlying dependence used for generating the samples to measure the rate of convergence of τ_χ. In this case, D, the dimensionality of the feature vector is 10, which is equal to the number of nodes in this tree.



(b) Variation of the ensemble average of the closeness of approximation $I(P, P_a)$ between the actual distribution P and the estimated distribution P_a derived from the estimated dependence tree. The metric used to derive this tree is the I_χ metric. The underlying dependence tree describing P is given in Figure 3a.

Figure 3: Rate of Convergence (D=10)

probability distribution for the case of binary valued features and then computing each of τ^* and τ_χ 5000 times each. The figures on CPU time usage suggest that the computation of τ_χ takes approximately 22% of the time required to compute τ^*. This is by no means a small reduction in the demand for CPU time.

5 Conclusion

In this paper, we have considered the problem of approximating an unknown underlying discrete probability distribution, by one derived from a dependence tree. The best dependence tree τ^* is known to be the MST of a complete graph, with $I^*(i,j)$ as the edge weight between the pair of nodes i and j. This paper proposes a chi-squared based metric, I_χ to capture the dependence information between pairs of features, and the tree generated using I_χ is referred to as τ_χ.

The quantities I^* and I_χ are shown to follow one another locally, i.e. they increase or decrease together, if we restrict ourselves to binary features only. We believe that this results holds in general, but have not obtained proofs for this conjecture. By suitably restricting the domain from which the joint probabilities can be assigned to pairs of features, we have shown both these metrics to be equivalent (and hence optimal) in this restricted world.

Experimental results clearly demonstrate that when the underlying unknown distribution is derived from a dependence tree, both metrics I^* and I_χ succeed in finding it. When the undelying dependence is not actually based on a tree, both τ^* and τ_χ are estimates for the best dependence tree. Even in this case, whenever the two estimates of the best dependence tree do not always match, their total weights are almost indistinguishably close.

The main advantage with our metric I_χ is that it is computationally far superior to the metric I^*; this is a direct consequence of not requiring the evaluation of numerous logarithms. The experimental results suggest that τ_χ can be computed in a small fraction of the time required to compute τ^*. The savings in terms of CPU time, will be even more significant, if the number of values that each feature can assume, increases.

References

[1] A.V. Aho, J.E. Hopcroft, J.D. Ullman, *The Design and Analysis of Algorithms*, Addison-Wesley, 1974.

[2] C.K. Chow and C.L. Liu, "Approximating Discrete Probabililty Distributions Using Dependence Trees", *IEEE Trans. on Information Theory*, Vol. IT-14, 1968, pp. 462-467.

[3] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley Interscience, 1973.

[4] H. Ku and S. Kullback, "Approximating Discrete Probability Distributions", *IEEE Trans. on Information Theory*, Vol. IT-14, 1968, pp. 462-467.

[5] R.S. Valiveti, Ph.D. thesis, Carleton University, in preparation.

[6] C.J. Van Rijsbergen, "A Theoretical Basis For the Use of Co-occurrence Data in Information Retrieval", *Journal Of Documentation*, Vol. 33, No. 2, June 1977, pp. 106-119.

[7] S.K.M. Wong, and F.C.S. Poon, "Comments on approximating Discrete Probability Distribution with Dependence Trees", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 3, March 1989, pp. 333-335.

Appendix

In Section 2, it was mentioned that the maximum likelihood estimate of the dependence tree can be obtained by **Algorithm Chow**. The result is indeed powerful and not obvious at first sight. This part of the paper is an attempt to expand on the (outline of) the proof that appeared in [2]. To simplify the details we will be proving Theorem 0, only for the case of binary valued features, but the generalization will become obvious, once the reader follows this proof.

Proof Of Theorem 0

Let $X^1, X^2, \ldots, X^s$ be the s *independent* observations (given the underlying dependence tree t). The likelihood function is then:

$$L_t(X^1, X^2, \ldots, X^s) = \prod_{k=1}^{s} P_t(X^k)$$

where $P_t(X^k)$ is the joint probability that the vector X^k is derived from the particular tree t. If we use the product rule for the distribution generated by this tree (5), we get the following equation for the Likelihood function:

$$L_t(X^1, X^2, \ldots, X^s) = \prod_{k=1}^{s} \prod_{i=1}^{N} Pr(x^k_{m_i} \mid x^k_{m_{j(i)}})$$

Instead of maximizing $L_t(X^1, X^2, \ldots, X^s)$, we attempt to maximize its logarithm, and denote it by $l_t(X^1, X^2, \ldots, X^s)$. Therefore,

$$\begin{aligned} l_t(X^1, X^2, \ldots, X^s) &= \log L_t(X^1, X^2, \ldots, X^s) \\ &= \sum_{k=1}^{s} \sum_{i=1}^{N} \log\{Pr(x^k_{m_i} \mid x^k_{m_{j(i)}})\} \\ &= \sum_{i=1}^{N} \sum_{k=1}^{s} \log\{Pr(x^k_{m_i} \mid x^k_{m_{j(i)}})\} \end{aligned} \tag{24}$$

the final expression being obtained by interchanging summations.

The maximum likelihood tree is the value $t = \tau$, for which the above sum is maximized. Now consider the two features x_{m_i} and $x_{m_{j(i)}}$. From the definition of the dependence tree, we realize that in the dependence tree t, there is an edge between these nodes. In order to compute the likelihood function as given in (24), we need to know the parameters

$Pr(x_{m_i} = b1 \mid x_{m_{j(i)}} = b2)$ for $b1 \in \text{Domain}(x_{m_i})$ and $b2 \in \text{Domain}(x_{m_{j(i)}})$. If all the features of interest are binary valued, $b1, b2 \in \{0, 1\}$. In this case, essentially, we have two independent parameters, these being $Pr(x_{m_i} = 0 \mid x_{m_{j(i)}} = 0)$ and $Pr(x_{m_i} = 0 \mid x_{m_{j(i)}} = 1)$. Conceptually, these paramters can be thought as being associated with a branch of the dependence tree. For the (N-1) branches of the tree, we have $2(N-1)$ parameters all of which can be independently chosen.

The reader will now realize that in essence, we are faced with two decisions. The first one involves selecting the best values for the above mentioned parameters — two for each edge of the tree. The second problem concerns choosing the best estimate τ from the set of all spanning trees. To obtain the Maximum likelihood estimate, we should study the problem of maximizing the sum in (24), *for a particular tree.* That is, given a *particular* tree, we are trying to find the best estimates for the conditional probabilities such that the sum is maximized. Note that in (24), the term $\sum_{k=1}^{s} \log\{Pr(x^k_{m_i} \mid x^k_{m_{j(i)}})\}$ corresponds to adding the logarithms of the parameters associated with the branch $(x_{m_i}, x_{m_{j(i)}})$ over all the samples. Since the parameters for the individual branches are independent, we can maximize the terms corresponding to each branch and then add the corresponding maximums that have been obtained. That is, for a given tree t,

$$\begin{aligned} \max l_t(X^1, X^2, \ldots, X^s) &= \max \left\{ \sum_{i=1}^{N} \sum_{k=1}^{s} \log\{Pr(x^k_{m_i} \mid x^k_{m_{j(i)}})\} \right\} \\ &= \sum_{i=1}^{N} \left\{ \max \sum_{k=1}^{s} \log\{Pr(x^k_{m_i} \mid x^k_{m_{j(i)}})\} \right\} \end{aligned}$$

For simplicity, let $\alpha = m_i$ and $\beta = m_{j(i)}$. The above sum is composed of terms of the form $Pr(x_\alpha = b1 \mid x_\beta = b2)$, where $b1, b2 \in \{0, 1\}$. Therefore,

$$\begin{aligned} \sigma &\triangleq \sum_{k=1}^{s} \log\{Pr(x^k_{m_i} \mid x^k_{m_{j(i)}})\} \\ &= \sum_{k=1}^{s} \log Pr(x^k_\alpha \mid x^k_\beta) \\ &= \sum_{i,j=0,1} n_{ij} \log Pr(x_\alpha = i \mid x_\beta = j) \end{aligned}$$

where, n_{ij} is the number of samples in which $x_\alpha = i$ and $x_\beta = j$. Also note that $\sum n_{ij}$ is equal to s, the total number of samples.

Now, if we introduce the notation that, $\delta_{ij} = Pr(x_\alpha = i \mid x_\beta = j)$, we get,

$$\sigma = n_{00}\log\delta_{00} + n_{01}\log\delta_{01} + n_{10}\log\delta_{10} + n_{11}\log\delta_{11} \tag{25}$$

Our aim is to maximize σ under the constraints $\delta_{00} + \delta_{10} = 1$ and $\delta_{01} + \delta_{11} = 1$; the latter conditions being a consequence of the conditional distributions. Simplifying (25), subject to the above mentioned constraints, we have,

$$\begin{aligned}\sigma &= \log\left\{\prod_{i,j}\delta_{ij}^{n_{ij}}\right\} \\ &= \log\left[\underbrace{\{\delta_{00}^{n_{00}}(1-\delta_{00})^{n_{10}}\}}\underbrace{\{\delta_{01}^{n_{01}}(1-\delta_{01})^{n_{11}}\}}\right]\end{aligned}$$

To maximize σ, we must individually maximize the terms within the square brackets. By using the methods of elementary calculus, We can easily establish that $\{\delta_{00}^{n_{00}}(1-\delta_{00})^{n_{10}}\}$ and $\{\delta_{01}^{n_{01}}(1-\delta_{01})^{n_{11}}\}$ attain their maximum values when

$$\begin{aligned}\delta_{00} &= \frac{n_{00}}{n_{00}+n_{10}} \\ \delta_{01} &= \frac{n_{01}}{n_{01}+n_{11}}\end{aligned}$$

respectively. Therefore σ is maximized when the individual probabilities are chosen to be their own Maximum likelihood estimates which are indeed their intuitive frequency based estimates. Now, from (24) we have,

$$\begin{aligned}l_t(X^1, X^2, \ldots, X^s) &= \sum_{i=1}^{N}\sum_{k=1}^{s}\log\{Pr(x_{m_i}^k \mid x_{m_{j(i)}}^k)\} \\ &= \sum_{i=1}^{N}\sum_{k=1}^{s}\log\{\frac{Pr(x_{m_i}^k, x_{m_{j(i)}}^k)}{Pr(x_{m_i}^k)Pr(x_{m_{j(i)}}^k)}\} + \sum_{i=1}^{N}\sum_{k=1}^{s}\log(Pr(x_{m_i}^k)) \\ &= s\sum_{i=1}^{N}\hat{I}(x_{m_i}, x_{m_{j(i)}}) + K.\end{aligned}$$

where K is independent of the tree but is only a function of the samples. Hence to maximize $l_t(X^1, X^2, \ldots, X^s)$, over all possible trees, we only need to compute the MST of the graph with $\hat{I}(x_i, x_j)$ as edge weights. Hence the result.

□