Uncertainty in Artificial Intelligence 2
J.F. Lemmer and L.N. Kanal (Editors)

A NON-ITERATIVE MAXIMUM ENTROPY ALGORITHM

Sally A. Goldman and Ronald L. Rivest

Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

We present a new algorithm for computing the maximum entropy probability distribution satisfying a set of constraints. Unlike previous approaches, our method is integrated with the planning of data collection and tabulation. We show how *adding constraints* and performing the associated additional tabulations can substantially speed up computation by replacing the usual iterative techniques with a non-iterative computation. We note, however, that the constraints added may contain significantly more variables than any of the original constraints so there may not be enough data to collect meaningful statistics. These extra constraints are shown to correspond to the intermediate tables in Cheeseman's method. Furthermore, we prove that acyclic hypergraphs and decomposable models are equivalent, and discuss the similarities and differences between our algorithm and Spiegelhalter's algorithm. Finally, we compare our work to Kim and Pearl's work on singly-connected networks.

1 Introduction

Many applications require reasoning with incomplete information. For example, one may wish to develop expert systems that can answer questions based on an incomplete model of the world. Having an incomplete model means that some questions may have more than one answer consistent with the model. How can a system choose reasonable answers to these questions?

Many solutions to this problem have been proposed. While probability theory is the most widely used formalism for representing uncertainty, ad-hoc approximations to probability have been used in practice. The problem with pure probabilistic approaches is that their computational complexity seems prohibitive. One way to reduce the complexity of this problem is to make the strong assumption of conditional independence. However, this assumption is typically not valid and generates inaccurate results. A probabilistic method which shows promise for solving some of the problems with uncertain reasoning is the maximum entropy approach.

This research was supported in part by NSF Grant DCR-8006938. Sally Goldman received support from an Office of Naval Research Fellowship.

A preliminary version of this paper was presented at the 6th Annual Workshop on Maximum Entropy and Bayesian Methods in Applied Statistics; Seattle, Washington, August 1986.

This paper discusses efficient techniques, based on the principle of maximum entropy, for answering questions when given an incomplete model. The organization is as follows. Section 2 contains a formal definition of the inference problem. In sections 3 and 4, we introduce the maximum entropy principle, and review iterative methods for calculating the maximum entropy distribution. Section 5 contains a discussion of some non–iterative techniques which use a conditional independence assumption rather than maximum entropy to obtain a unique probability distribution. In sections 6 and 7, we examine Malvestuto's work on acyclic hypergraphs, and present a new technique which makes maximum entropy computations non–iterative by adding constraints. Section 8 discusses Spiegelhalter's non–iterative algorithm for estimating a probability distribution. In section 9, we present some interesting comparisons between our technique and those of Cheeseman, Spiegelhalter, and Kim and Pearl. We conclude with some open problems.

2 Formal Problem Definition

In this section, we formally define the inference problem which this paper addresses. We begin by defining some notation. Let $V = \{A, B, C, \ldots\}$ be a finite set of binary-valued variables, or attributes. (The generalization to finite-valued variables is straightforward.) Consider the event space Ω_V defined to be the set of all mappings from V to $\{0,1\}$. We call such mappings *assignments* since they assign a value to each variable in V. It is easy to see that $|\Omega_V| = 2^{|V|}$. If $E \subseteq V$, we have Ω_V is isomorphic to $\Omega_E \times \Omega_{V-E}$; we identify assignments in Ω_E with subsets of Ω_V in the natural manner.

We are interested in probability distributions defined on Ω_V. We use the following convention throughout this paper. If $E \subseteq V$, we write $P(E)$ to denote the probability of an element of $\Omega_E \subseteq \Omega_V$. In other words, we specify only the variables involved in the assignments and not their values. For example,

$$P(V) = P(A)P(B)P(C)\cdots \tag{1}$$

represents $2^{|V|}$ equations, stating that the variables are independent. (We do not assume equation (1).) By convention, all assignments in an equation must be consistent. We also write $P(A)$ instead of $P(\{A\})$, $P(AB)$ instead of $P(\{AB\})$, and so on.

We use a similar convention for summations: $\sum_E$ stands for a summation over all assignments in Ω_E, when $E \subseteq V$. Using these conventions, we see that $Y \subseteq E \subseteq V$ implies that

$$P(Y) = \sum_{E-Y} P(E) \tag{2}$$

For example, if $E = \{ABCD\}$ and $Y = \{AB\}$ then $P(Y) = \sum_{CD} P(E)$.

For conditional probabilities we use similar notation. For $Y \subseteq E \subseteq V$ the probability of E given Y is written as $P(E|Y)$ and defined to be

$$P(E|Y) = P(E - Y|Y) = \frac{P(E \wedge Y)}{P(Y)}. \tag{3}$$

We say that X and Y are *conditionally independent* given S if

$$P(X \wedge Y|S) = P(X|S)P(Y|S) \tag{4}$$

where $X, Y, S \subseteq V$.

We are interested in probability distributions on Ω_V satisfying a set of constraints. We assume that the constraints are supplied in the form of *joint marginal probabilities*. Let $E_1, \ldots, E_m$ be distinct but not necessarily disjoint subsets of V. Let us suppose that for each i we are given the $2^{|E_i|}$ constraint values $\{P(E_i)\}$. Furthermore, we assume that these values are *consistent*. By consistent we mean that there exists at least one probability distribution on Ω_V which satisfies the constraints. Note that equation (2) states that a constraint on the values $P(Y)$ is implied by a constraint on the values $P(E)$ when $Y \subseteq E$. A common way of ensuring that the constraints are consistent is to derive the constraints by computing the observed marginal probabilities from a common set of data [1].

Many techniques require that constraints are given in the form of *conditional probabilities*. Here, each element of $E_1, \ldots, E_m$ has a distinguished subset e_i, and the input is the $2^{|E_i|}$ constraint values $\{P(E_i - e_i | e_i)\}$ for each i. This approach is frequently used when obtaining information form experts because it is often easier for experts to give information in terms of conditional probabilities. In general, we find that joint marginal distributions are easier to handle. Since we plan to obtain the constraint values from raw data, we are free to use joint marginal distributions. By doing so, we avoid both the possibility of inconsistent data, and the difficulty in transforming conditional distributions to joint marginal distributions.

In general, there may be many probability distributions satisfying the constraints. There are two problems which must be addressed. First, assuming that there are many probability distributions satisfying the constraints, which one should be chosen? And second, how can one *efficiently* calculate the desired distribution? Most of our attention shall be given to the second of these two problems, but we briefly address the first in the following section.

3 The Maximum Entropy Principle

In this section, we formally define the maximum entropy principle. When faced with an underconstrained problem, a reasonable way to get a unique answer is to apply the principle of maximum entropy. The entropy function, H, is defined as follows:

$$H(P) = -\sum_V P(V) \log(P(V)). \tag{5}$$

The maximum entropy probability distribution, P^*, is the unique distribution which maximizes H while satisfying the supplied constraints. Informally, the maximum entropy principle says that when one makes inferences based on incomplete information, one should draw them from the probability distribution that has the maximum uncertainty permitted by the data. That is, the maximum entropy distribution is the unique distribution which is maximally noncommittal with regard to missing information. Motivation for this choice are discussed by Jaynes [18,19]. Arguments formally justifying the choice of the maximum entropy distribution are provided by Rissanen, Shore and Johnson, and Tikochinsky, Tishby and Levine [20,28,31,35].

Given that the distribution of choice is the one with maximum entropy, an efficient algorithm for calculating the maximum entropy distribution is desired. The remaining sections of this paper will address this goal.

One problem that is immediately apparent is that space required to store a probability distribution is exponential in $|V|$. An advantage of the maximum entropy distribution,

[1] Using "experts" to provide subjective probability estimates is a well-known way of deriving a set of *inconsistent* constraints [36].

P^*, is that it has a simple representation. For each ω in Ω_{E_i}, there is a non-negative real parameter $\alpha_{E_i}(\omega)$ (i.e., one parameter set per constraint set and $2^{|E_i|}$ parameters in the parameter set for E_i), that determine P^* as follows. Let us write $\alpha_i(\omega)$ instead of $\alpha_{E_i}(\omega)$ for brevity, and omit the argument ω when it can be deduced from context. Now we may simply write

$$P^*(V) = \alpha_1 \alpha_2 \ldots \alpha_m. \tag{6}$$

Each element of Ω_V is assigned a probability which is the product of the appropriate α's where each α determines its argument from the assignment to V. This is known as a *log-linear* representation. For example, suppose we have the variables A, B, C and constraint sets $E_1 = \{AB\}$ and $E_2 = \{BC\}$. Then we will have the parameter sets α_{AB} and α_{BC}, where the corresponding parameters are $\alpha_{AB}(00), \ldots, \alpha_{AB}(11)$ and $\alpha_{BC}(00), \ldots, \alpha_{BC}(11)$. The maximum entropy distribution is given by the log–linear model

$$P^*(V) = \alpha_{AB}\alpha_{BC}.$$

If $P_{ABC}(010)$ (i.e., the probability that $A = 0$, $B = 1$, and $C = 0$) is desired, it can be calculated by

$$P^*(010) = \alpha_{AB}(01)\alpha_{BC}(10).$$

Thus, the maximum entropy distribution can be represented in a linear amount of space in the number of constraints.

4 Iterative Maximum Entropy Methods

Most existing methods for calculating the maximum entropy distribution are iterative. They typically begin with a representation of the uniform distribution and converge towards a representation of the maximum entropy distribution. Each step adjusts the representation so that a given constraint is satisfied. To enforce a constraint $P(E_i)$, all of the elementary probabilities $P(V)$ relevant to that constraint are multiplied by a common factor. Because constraints are dependent, adjusting the representation to satisfy one constraint may cause a previously satisfied constraint to no longer hold. Thus, one must iterate repeatedly through the constraints until the desired accuracy is reached. (We note that the implicit constraint — that the probabilities sum to one — must usually be explicitly considered.) Examples of this type of algorithm are discussed in [5,6,13,17,21,23].

Representing the probability distribution explicitly as a table of $2^{|V|}$ values is usually impractical. For this problem, it is most convenient to store only $\alpha_1, \alpha_2, \ldots \alpha_m$; this is a representation as compact as the input data, which represents the current probability distribution implicitly via equation (6). To represent the uniform distribution, every α is set to 1, except for the α corresponding to the requirement that entries of the probability distribution must sum to 1 — which is set to $2^{-|V|}$. To determine if a constraint is satisfied, one must sum the appropriate elements of the probability distribution; any particular element can be computed using equation (6). If the constraint is not satisfied, the relevant α is multiplied by the ratio of the desired sum to the computed sum. Thus, in originally calculating the α's and later in evaluating queries it is necessary to evaluate a sum of terms, where each term is a product of α's. This sum is difficult to compute since it may involve an exponential number of terms.

Cheeseman [6] proposes a clever technique for rewriting such sums to evaluate them more efficiently. For example

$$\alpha \sum_{A \ldots F} \alpha_{AB}\ \alpha_{ACD}\ \alpha_{DE}\ \alpha_{AEF}$$

is rewritten as follows. First, $\sum_{A...F}$ is broken into six sums, each over one variable. Arbitrarily choosing the variable ordering $CDFEAB$, we obtain

$$\alpha \sum_B \sum_A \sum_E \sum_F \sum_D \sum_C \alpha_{AB}\alpha_{ACD}\alpha_{DE}\alpha_{AEF}.$$

Now each α is moved left as far as possible (it stops when reaching a sum over a variable on which it depends). The above sum then becomes

$$\alpha \sum_B \sum_A \alpha_{AB} \sum_E \sum_F \alpha_{AEF} \sum_D \alpha_{DE} \sum_C \alpha_{ACD}.$$

The sums are evaluated from right to left. The result of each sum is an *intermediate table* containing the value of the sum evaluated so far as a function of variables further to the left which have been referenced. The variable ordering must be chosen carefully in order to take full advantage of this technique. A poor choice of variable ordering can yield a sum which is not much better than explicitly considering all $2^{|V|}$ terms; a good choice may dramatically reduce the work required. While picking a good variable ordering is important, the success of this technique depends greatly on the interconnectedness of the constraints. If the constraints are highly connected, no ordering can significantly reduce the complexity of evaluating the summation.

Some alternative approaches to the standard iterative schemes have been proposed. One of the more interesting proposals is due to Geman [14,24]. Instead of considering one constraint at a time, this algorithm uses stochastic relaxation to simultaneously adjust the probability distribution to meet all of the constraints. In particular, a convex function, whose minimum gives the maximum entropy distribution, is calculated. Then a technique to approximate the gradient and a gradient descent algorithm are used to find this minimum.

An approach which comes immediately from the Lagrange multiplier technique is discussed by Agmon, Alhassid, and Levine [1,2]. First they calculate the "potential function" $F(\lambda^t)$. They show that F is strictly convex, and has a unique global minimum for λ^t which solves $\nabla F(\lambda^t) = 0$. Finally, they use the modified Newton-Raphson procedure to find the global minimum.

Unlike most approaches which are based on the Lagrange multiplier technique, Csiszár [9] uses I–divergence geometry to prove that a generalized version of the standard iterative technique converges. His proof is more general than those which are derived from the Lagrange multiplier technique.

5 Non–Iterative Techniques

The iterative techniques of the previous section are applicable in most situations, but computationally they are rather inefficient. We want a model that is powerful enough to handle real–world situations, yet simple enough for the maximum entropy distribution to be calculated efficiently. In this section, we discuss some non–iterative approaches which use conditional independence instead of the principle of maximum entropy to obtain a unique result.

Chow and Liu [8] consider the class of product approximations in which only second–order distributions are used. If there is a product approximation such that for some ordering of the variables $x_1 \ldots x_n$, each x_i depends on at most one variable from the set $\{x_1, \ldots x_{i-1}\}$,

then this approximation forms a *dependence tree*. They discuss how to build the best dependence tree when supplied with the complete probability distribution. They also present a method to construct an optimal dependence tree from samples.

Similar to their work is the work of Kim and Pearl [22]. They construct a *Bayesian network* where the nodes represent variables and directed links represent direct dependencies; all direct influences on a node come from its parents. We will use the following notation for stating their formula: S_x is the set of the *immediate* predecessors (parents) of node x in the network, T_x is the set of *all* predecessors of node x in the network, and $\mathcal{R}$ is the set of roots (sources). All conditional probabilities of the form $P(x|S_x)$ for all $x \notin \mathcal{R}$ and $P(x)$ for all $x \in \mathcal{R}$, along with the independence assumptions that $P(x|S_x) = P(x|T_x)$ suffice to define the following unique probability distribution:

$$P^*(V) = \left(\prod_{x \in \mathcal{R}} P(x)\right)\left(\prod_{x \notin \mathcal{R}} P(x|S_x)\right). \tag{7}$$

They define a network to be *singly-connected* if there is at most one *undirected* path between any pair of nodes. One of the most interesting results of this work is that the propagation of new evidence through a singly–connected network can be accomplished by a network of parallel processors in time proportional to the longest path in the network. Pearl [27] addresses the problem of propagating new evidence through *multiply–connected* networks.

Dalkey [10] has shown that if a Bayesian network is a tree (i.e. all nodes have at most one parent), equation (7) gives the maximum entropy distribution. So for a Bayesian network which is a tree, a non–iterative technique exists for calculating the maximum entropy distribution when the input is all conditional probabilities of the form $P(x|S_x)$ for all $x \notin \mathcal{R}$ and $P(x)$ for all $x \in \mathcal{R}$.

6 Acyclic Hypergraphs

Even with Cheeseman's summation technique, the general iterative algorithm discussed in section 4 still has a very high (exponential) computational cost since many iterations through the constraints are required before the distribution converges. The non–iterative techniques discussed in the previous section are efficient, but they assume conditional independence which is rarely present. What we want is an non–iterative maximum entropy algorithm. If we are willing to put restrictions on the supplied constraints, this goal can be achieved.

Malvestuto [25] introduced a way to model constraints which are joint probability distributions as a hypergraph and presented a non–iterative maximum entropy algorithm for hypergraphs which are acyclic. In this section we introduce his work. We begin by describing how to model a set of variables and associated constraints as a hypergraph. It is interesting to note that the work on acyclic hypergraphs first appeared in the database literature [3,4,26,34]. The variables in our problem replace the attributes of the database, and the constraint sets replace the relations. If the database schema is acyclic, many problems can be simplified.

A hypergraph is like an ordinary undirected graph, except that each edge may be an arbitrary subset of the vertices, instead of just a subset of size two. We define the hypergraph $H = (\mathcal{V}, \mathcal{E})$ to contain a vertex for each variable, and a hyperedge for each constraint. For example the hyperedge $\{ABC\}$ corresponds to the constraint set $E_i = \{A, B, C\}$. We say

that hyperedge X *subsumes* hyperedge Y if $Y \subseteq X$. It is important to observe that the constraints on a sub-hypergraph induced by restricting attention to a subset of the vertices can be inferred from the original hypergraph constraints using equation (2).

We define the graph $C(H)$ of a hypergraph H to be the graph whose vertices are those of H and whose edges are the vertex pairs $\{v, w\}$ such that v and w are in a common hyperedge of H. A hypergraph H is *conformal* if every clique of $C(H)$ is contained in a hyperedge of H. A graph, H, is *chordal* if for every cycle of length greater than three, there is an edge of H joining two non–consecutive vertices A hypergraph H is *acyclic* if H is conformal and $C(H)$ is chordal.

An equivalent definition is that a hypergraph is acyclic if repeatedly applying the following reduction steps results in the empty hypergraph (containing no hyperedges and no vertices):

1. Delete any vertices which belong to only one hyperedge.
2. Delete any hyperedges which are subsumed by another hyperedge.

Graham's algorithm is the procedure of applying reduction steps 1 and 2 until either the empty set is reached, or neither can be applied [16].

Before proceeding, we shall define some notation regarding the above reduction procedure. Let $\mathcal{E}^{(0)} = \{E_1^{(0)}, \ldots, E_m^{(0)}\}$, where $E_i^{(0)}$ is the i^{th} hyperedge of H. Let $Y_i^{(k)}$ be the set of variables which appear in at least one hyperedge other than $E_i^{(k)}$. Finally let $\mathcal{E}^{(i+1)}$ be the result of applying reduction step (1) and then (2) to $\mathcal{E}^{(i)}$. If H is acyclic then there exists an l such that $\mathcal{E}^{(l+1)} = \emptyset$.

For acyclic hypergraphs, Malvestuto [25] gave the following formula for the maximum entropy distribution, $P^*(V)$:

$$P^*(V) = \left(\prod_{k=0}^{l-1} \frac{\prod_i P\left(E_i^{(k)}\right)}{\prod_i P\left(Y_i^{(k)}\right)} \right) \left(\prod_i P\left(E_i^{(l)}\right) \right) \tag{8}$$

Note that no α's are needed; the formula depends only on joint marginal distributions of the constraints. This formula is an immediate extension of the following theorem due to Malvestuto [25].

Theorem 1 *Given the constraints $\{E_1, \ldots, E_m\}$, the maximum entropy distribution is given by the following.*

$$P^*(V) = \frac{P(E_1) \cdots P(E_m)}{P(Y_1) \cdots P(Y_m)} P^*(Y)$$

where Y_i is the set of variables which appear in at least one hyperedge other than E_i and $P^*(Y)$ is the maximum entropy distribution for the constraints $\{Y_1, \ldots, Y_m\}$.

Proof: From the marginal constraints we have the following

$$\begin{aligned} P(E_i) &= \sum_{V - E_i} \alpha_1 \cdots \alpha_m \\ &= \alpha_i \sum_{V - E_i} \prod_{j \neq i} \alpha_j \end{aligned} \tag{9}$$

Similarly we have,

$$\begin{aligned} P(Y_i) &= \sum_{Z_i} \sum_{V - E_i} \alpha_1 \cdots \alpha_m \\ &= \left(\sum_{Z_i} \alpha_i \right) \left(\sum_{V - E_i} \prod_{j \neq i} \alpha_j \right) \end{aligned} \tag{10}$$

Let $\beta_i = \sum_{Z_i} \alpha_i$. Combining equations (9) and (10) from above gives:

$$\alpha_i = \frac{P(E_i)}{P(Y_i)}\beta_i \tag{11}$$

Now writing $P^*(V)$ in its product form we get

$$\begin{aligned} P^*(V) &= \alpha_1 \cdots \alpha_m \\ &= \frac{P(E_1)\cdots P(E_m)}{P(Y_1)\cdots P(Y_m)}\beta_1 \cdots \beta_m \end{aligned} \tag{12}$$

We want to show that $\psi(Y) = \beta_1 \cdots \beta_m$ is $P^*(Y)$, the maximum entropy distribution for the constraints $P(Y_i)$. To do this, it is suffices to prove that the marginal constraints hold.

$$\begin{aligned} P^*(E_i) &= \sum_{V-E_i} \left(\prod_j \frac{P(E_j)}{P(Y_j)} \right) \psi(Y) \\ &= \sum_{V-E_i} \psi(Y) \frac{P(E_i)}{P(Y_i)} \prod_{j \neq i} \frac{P(E_j)}{P(Y_j)} \\ &= \frac{P(E_i)}{P(Y_i)} \sum_{Y-Y_i} \left(\psi(Y) \prod_{j \neq i} \frac{1}{P(Y_j)} \sum_{Z-Z_i} \left(\prod_{j \neq i} P(E_j) \right) \right) \end{aligned} \tag{13}$$

where $Z = Z_1 \cup \cdots \cup Z_m$, so that $V = Y \cup Z$. Now, since the Z_j's are disjoint,

$$\begin{aligned} \sum_{Z-Z_i} \prod_{j \neq i} P(E_j) &= \prod_{j \neq i} \sum_{Z-Z_i} P(E_j) \\ &= \prod_{j \neq i} \sum_{Z_j} P(E_j) \\ &= \prod_{j \neq i} P(Y_j) \end{aligned} \tag{14}$$

Substituting equation (14) into equation (13) gives:

$$P^*(E_i) = \frac{P(E_i)}{P(Y_i)} \sum_{Y-Y_i} \psi(Y)$$

However since $P^*(E_i) = P(E_i)$ we get $P(Y_i) = \sum_{Y-Y_i} \psi(Y)$, so $\psi(Y)$ satisfies the constraints $P(Y_i)$. ∎

7 A New Maximum Entropy Method

While Malvestuto's work provides a non–iterative maximum entropy algorithm for acyclic hypergraphs, in practice, this technique is not very useful since typically constraint sets do not form acyclic hypergraphs. While one could make a hypergraph acyclic by removing some hyperedges, this approach would lead to inaccurate results. In this section, we propose a new maximum entropy algorithm which is based on the observation that a hypergraph can be made acyclic by *adding* hyperedges. In other words, maximum entropy computations can actually be simplified by adding constraints. The main advantage of our procedure is that it avoids the iteration previously required by providing a non–iterative

formula for the desired answer. The major disadvantage is that the method cannot ordinarily be applied if the data is already tabulated and the constraints already derived; the method requires that one "plan ahead" and tabulate additional constraints when processing the data.

We begin by describing our algorithm. Equation (8) allows one to avoid iteration when calculating the maximum entropy distribution for schemas having acyclic hypergraphs. What should one do for *cyclic* hypergraphs? Our method is based on the observation that ***a hypergraph can always be made acyclic by adding hyperedges.*** (This is trivial to prove, since at worst a hypergraph can be made acyclic by adding the hyperedge containing all vertices.) For example, the hypergraph:

$$(\mathcal{V}, \mathcal{E}) = (\{ABCDEF\}, \{\{AB\}, \{ACD\}, \{DE\}, \{AEF\}\})$$

becomes acyclic when the hyperedge $\{ADE\}$ is added (see figures 1 and 2).

Thus, by adding additional constraints (edges) the maximum entropy calculation can be

$$\begin{array}{c} \{\underline{A}B, \underline{AC}\underline{D}, \underline{DE}, \underline{AE}F\} \\ \Downarrow \\ \{\underline{AD}, \underline{DE}, \underline{AE}\} \end{array}$$

Figure 1: The hypergraph consisting of the hyperedges $\{AB\}$,$\{ACD\}$,$\{DE\}$, and $\{AEF\}$ is cyclic as shown be the reduction above. (Elements of Y_i are underlined.)

$$\begin{array}{c} \{\underline{A}B, \underline{AC}\underline{D}, \underline{ADE}, \underline{AE}F\} \\ \Downarrow \\ \{ADE\} \\ \Downarrow \\ \emptyset \end{array}$$

Figure 2: The hypergraph consisting of the hyperedges $\{AB\}$, $\{ACD\}$, $\{ADE\}$, and $\{AEF\}$ is acyclic as shown be the reduction above. (Elements of Y_i are underlined.)

simplified so that no iteration is required. Here is a summary of how our method works:

1. We begin with a set of variables (attributes) and constraint sets deemed to be of interest. (Cheeseman [7] discusses a learning program which uses the raw data to find a set of significant constraints. Edwards and Kreiner [12] also discuss how to choose a good set of constraints.) Here a "constraint set" is a set of variables; the intent is that during data-gathering there will be one joint marginal distribution table created for each constraint set, and the observed events will be tabulated once in each table according to the values of the attributes in the constraint set. For example, if $\{A, B, C\}$ is a constraint set of three binary–valued attributes, then there will be a table of size 8 used to categorize the data with respect to these three attributes. This results in 8 constraints on the maximum-entropy distribution desired, one for each of the eight observed probabilities $P(ABC)$.

2. Construct the corresponding hypergraph $H = (\mathcal{V}, \mathcal{E})$, where there is one vertex for each variable and one hyperedge corresponding to each constraint group.

3. Perform Graham's algorithm on H, and let H' denote the resulting hypergraph. If H' is the empty hypergraph, then H is acyclic, and the following step is skipped.

4. Find a set $\mathcal{X}$ of *additional* hyperedges (constraint groups) which can be added to H' to make it acyclic. Note that any original edges subsumed by edges in $\mathcal{X}$ are eliminated. These additional hyperedges should be chosen to minimize the space required to store the joint marginal distributions.

5. Collect data for the expanded set $\mathcal{E} \cup \mathcal{X}$ of constraints [2].

6. Apply equation (8) to calculate individual elements of the maximum entropy distribution.

With the exception of step 4, we have completely described how to perform each of the steps. We now discuss how to find a good set of hyperedges to add which makes a hypergraph acyclic. Finding the optimal set of hyperedges to add is extremely similar to the minimum fill–in problem which has been proven to be NP–complete [37], and thus we conjecture that our problem is also NP–complete. See Rose, Tarjan, and Lueker [29,30] for a discussion of the fill–in problem. There are many heuristics which have been studied for the minimum fill–in problem. We plan on using one of these heuristics, the minimum–degree heuristic, to find a good set of hyperedges to add. Here is our proposal for performing step 4 of our algorithm. We define the *degree* of a vertex v to be the number of vertices in a common hyperedge with v. We begin by calculating the degree of each vertex in H'. Let v be the vertex with the smallest degree (break ties at random). Add to $\mathcal{X}$ the hyperedge e which contains v and any vertices in a common hyperedge with v. Next, modify H' by adding e to it and performing Graham's algorithm. If H' is now the empty hypergraph then we are done, otherwise return to the step of calculating the degree of each vertex. The reason for choosing this algorithm, is that it usually keeps the hyperedges in $\mathcal{X}$ as small as possible.

We will now demonstrate our algorithm on the example of figure 3. First we must perform Graham's algorithm on H. The result is shown below, where elements of Y_i are underlined.

$$\{\underline{AB}, \underline{AC}, \underline{BCE}, \underline{BDF}, \underline{CD}\}$$
$$\Downarrow$$
$$\{\underline{AB}, \underline{AC}, \underline{BC}, \underline{BD}, \underline{CD}\}$$

So after performing step 3 of our algorithm we have the hypergraph H' shown in figure 4. Now we must find the set $\mathcal{X}$ of hyperedges to make H' acyclic. We start by calculating the degree of the vertices in H'. We find that A and D both have a degree of two, and B and C both have a degree of three. Since A (we could have picked D) has the smallest degree we let $\mathcal{X} = \{\{ABC\}\}$. After adding $\{ABC\}$ to H' and performing Graham's algorithm, we find that $H' = (\{BCD\}, \{\{BC\}, \{BD\}, \{CD\}\})$. Since H' is a complete graph the only way to make it acyclic is to add a hyperedge which contains all vertices of H'. Thus, $\mathcal{X} = \{\{ABC\}, \{BCD\}\}$. Finally after adding the hyperedges in $\mathcal{X}$ to those in H and removing any hyperedges subsumed by another, we obtain the acyclic hypergraph

[2] Our method is unusual in that it extends the set of tables (constraints) used to tabulate the data. To fill in the entries of a new table, the raw data must still be available in step (5). So, steps 1-4 may be considered to be "planning" steps.

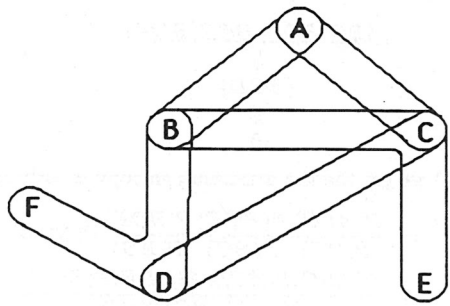

Figure 3: The original hypergraph $H = (\{ABCDEF\}, \{\{AB\}, \{AC\}, \{BCE\}, \{BDF\}, \{CD\}\})$.

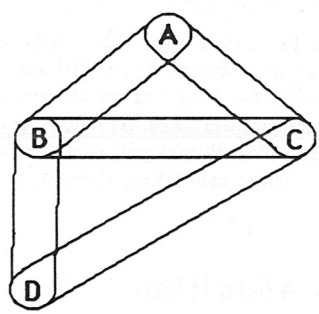

Figure 4: The hypergraph $H' = (\{ABCD\}, \{\{AB\}, \{AC\}, \{BC\}, \{BD\}, \{CD\}\})$ is obtained by performing Graham's algorithm on H.

$(\{ABCDEF\}, \{\{ABC\}, \{BCD\}, \{BCE\}, \{BDF\}\})$. So the set of constraints for which the data must be collected is $E_1 = \{ABC\}$, $E_2 = \{BCD\}$, $E_3 = \{BCE\}$, $E_4 = \{BDF\}$. Now in order to apply equation (8) we must perform Graham's algorithm on this new set of constraints.

$$\{\underline{ABC}, \underline{BCD}, \underline{BCE}, \underline{BDF}\}$$
$$\Downarrow$$
$$\{BCD\}$$
$$\Downarrow$$
$$\emptyset$$

Now applying equation (8) we get the the maximum entropy distribution is as follows:

$$\begin{aligned} P^*(V) &= \frac{P(ABC)}{P(BC)} \frac{P(BCE)}{P(BC)} \frac{P(BDF)}{P(BD)} P(BCD) \\ &= \frac{P(ABC)P(BCE)P(BDF)P(BCD)}{P(BC)P(BC)P(BD)} \end{aligned} \tag{15}$$

Finally, we consider possible inefficiencies of our method. First, it may be necessary to add "large" hyperedges containing many vertices in order to make the hypergraph acyclic. For example, to make the complete undirected graph (containing all hyperedges of size two) acyclic, one must add the "maximum" hyperedge containing all vertices. Since the size of the table corresponding to a hyperedge is an exponential function of the size of the hyperedge, adding large hyperedges creates a problem. Furthermore, the table corresponding to the maximum hyperedge is itself the probability distribution that we are estimating, so the above situation is clearly undesirable. This kind of behavior depends on the structure of the hypergraph; hypergraphs which are "highly connected" will tend to require the addition of large hyperedges. However, when the graph is highly connected other techniques seem to "blow up" as well.

Finally, because of our method's unique approach, we have a unique concern. Recall that since the data is tabulated *after* adding the additional constraints; steps 1-4 of our algorithm must be performed while the source of the constraints (i.e., the raw data) is still available. If the added hyperedges are too large, there may not be enough data to calculate meaningful statistics. Tabulating 100,000 data points in a table of size approximately 1,000 will give reasonable estimates, while tabulating them in a table of size approximately 1,000,000 will not.

8 Spiegelhalter's Algorithm

In this section, we briefly describe a method independently introduced by Spiegelhalter [32,33] for calculating the maximum entropy distribution. His work is based on *decomposable models* as discussed by Darroch, Lauritzen and Speed [11]. Spiegelhalter takes advantage of the fact that the maximum entropy probability distribution for decomposable models may be expressed as a simple function of the joint probabilities of the constraints from the model and their intersections.

A major difference between his work and our work is that he assumes that the data is given as *conditional* probability distributions rather that joint marginal distributions. From these conditional probabilities he finds a set of joint marginal distributions which form a decomposable model. To obtain the needed data, he used equation (7) to calculate the joint probability distributions. Since equation (7) only gives the maximum entropy

distributions for networks which are singly–connected, in general Spiegelhalter's technique does not produce the maximum entropy distributions.

9 Comparisons

In this section we compare our technique for estimating the probability distribution to Cheeseman's algorithm, Spiegelhalter's algorithm, and Kim and Pearl's algorithm. In this paper we will just state the results which we have obtained. For a more complete discussion and proofs of the stated results see Goldman [15].

We begin by comparing our algorithm to Cheeseman's algorithm. We have proven that for a given problem, the hyperedges (tables) added by our technique are like the intermediate tables used by Cheeseman's summation technique. The only difference between these tables is that for Cheeseman's technique they are half the size, since they are summed over one of the variables in the table.

In terms of time complexity, Cheeseman's method specifies an iterative approximation of the αs, whereas our method requires no such iteration. So, if Cheeseman's method requires 10 iterations on the average, our method should yield an average speed-up of a factor of 10. In terms of space complexity, both methods use approximately the same amount of space. However, our method adds what might be called "permanent" edges, since they correspond to tabulations of the raw data. Note, however, that new edges may subsume and eliminate original edges, so the space required by our method may not be quite as great as it first appears. In Cheeseman's method the tables exist only temporarily during the course of the computation, and not all such tables may be needed at the same time. And finally, in terms of the "precomputation" needed, both methods need to compute a vertex ordering to use. We observe that a good summation ordering is a good ordering for eliminating vertices. So the problem of choosing the hyperedges to make a graph acyclic is essentially equivalent to the problem of choosing an optimal summation ordering. Therefore, we conclude that our algorithm will be generally more efficient than Cheeseman's algorithm, where both are applicable.

Next we compare our algorithm to Spiegelhalter's algorithm. We have shown that a graphical model is decomposable if and only if the corresponding hypergraph is acyclic. We also have proven that if the decomposable model obtained by Spiegelhalter's technique is the same as the acyclic hypergraph obtained after step 4 of our technique, then these techniques produce the same formula for the estimated probability distribution.

We now consider the additional data required by our technique and the additional data required by Spiegelhalter's technique. As we have mentioned, adding hyperedges to the original hypergraph corresponds to requiring joint marginal probability distributions which were not included in the original data. Now let's look at the data requirements for Spiegelhalter's algorithm. In order to get a decomposable model, Spiegelhalter must add edges to the original directed graph. In doing so, he may form cliques in the corresponding undirected graph which are larger than any of the original maximal cliques. These large cliques correspond to joint marginal probability distributions which are not contained in the original data. Finally, let's compare how we propose getting the additional data to how Spiegelhalter proposes doing so. We propose that this additional data is collected with the original data by having the first four steps of our algorithm be "planning" steps. On the other hand, Spiegelhalter's technique assumes conditional independence to get the additional data from the given constraints.

Finally, we will compare our technique to Kim and Pearl's technique which uses conditional independence instead of maximum entropy to obtain a unique distribution. We show that acyclic hypergraphs generalize singly–connected networks. We know that in the case where the Bayesian network is a tree, equation (7) gives the maximum entropy distribution, and when the network is not a tree, equation (7) does not give the maximum entropy distribution. However, if the independence constraints assumed by Kim and Pearl are supplied as additional constraints to a maximum entropy algorithm, then the two techniques give equivalent results. This is because the conditional independence constraints uniquely define a probability distribution. So, in some sense one could argue that a maximum entropy algorithm is more general that the one used by Kim and Pearl.

10 Conclusions and Open Problems

We have presented an efficient algorithm for calculating the maximum entropy distribution for a given set of attributes and constraints. Using a hypergraph to model the attributes and constraints, we have shown the benefits of making the corresponding hypergraph *acyclic*. We also have shown how to make a hypergraph acyclic by adding hyperedges (constraints). Finally, we have compared this new technique to Cheeseman's technique, Spiegelhalter's technique and Kim and Pearl's technique.

An open problem is to determine whether or not the problem of choosing the best set of hyperedges which will make a hypergraph acyclic is an NP–complete problem. We conjecture that this is the case, but have not yet been able to exhibit a proof. We would like either to find an NP–completeness proof, or to find a polynomial time algorithm to solve the problem.

Another direction of future research is to determine how well our new algorithm works on real–life problems. We intend to try our technique on some realistic examples. Our goal is to determine if the size of the hyperedges will remain within reasonable limits for realistic examples. We expect that in practice our new method will give substantial improvements in running time. Since our method adds tables which may be larger than the original ones, it may be interesting to explore how these larger tables impact data accuracy.

Another interesting problem is to find a condition which ensures the existence of a non–iterative algorithm for approximating the maximum entropy distribution when the input can consist of joint marginal probability distributions, conditional probability distributions, and some clearly defined independence constraints.

References

[1] Agmon, N., Y. Alhassid, R.D. Levine, "An Algorithm for Determining the Lagrange Parameters in the Maximal Entropy Formalism," In Levine and Tribune, editors, *The Maximum Entropy Formalism*, M.I.T. Press, (1979).

[2] Agmon, N., Y. Alhassid, R.D. Levine, "An Algorithm for Finding the Distribution of Maximal Entropy," *Journal of Computational Physics* **30**,2 (February 1979), 250–259.

[3] Berri, C., R. Fagin, D. Maier, A. Mendelzon, J.D. Ullman and M. Yannakakis, "Properties of Acyclic Database Schemas," in *Proc.* 13th *Annual ACM STOC* (1981), 355–362.

[4] Berri, C., R. Fagin, D. Maier and M. Yannakakis, "On the Desirability of Acyclic Database Schemas," *J. ACM*, **30**,3 (1983), 355-362.

[5] Brown, D.T., "A Note on Approximations to Discrete Probability Distributions," *Information and Control*, **2** (1959), 386–392.

[6] Cheeseman, P.C., "A Method For Computing Generalized Bayesian Probability Values For Expert Systems," in *Proc. Eighth International Conference on Artificial Intelligence* (August 1983), 198–202.

[7] Cheeseman, P.C., "Learning of Expert Systems From Data," in *Proc. IEEE Workshop on Principles of Knowledge Based Systems* (1984), 115–122.

[8] Chow, C.K. and C.N. Liu, "Approximating Discrete Probability Distributions With Dependence Trees," *IEEE Trans. on Info. Theory*, **IT-14**,3 (May 1968), 462–467.

[9] Csiszár, I., "I-Divergence geometry of probability distributions and minimization problems," *Annals of Probability*, **3**,1 (1975), 146–158.

[10] Dalkey, N.C., "Min-Score Inference on Probability Systems," *University Of California, Los Angeles Dept. of Computer Science Technical Report UCLA-ENG-CSL-8112*, (June 1981).

[11] Darroch, J.N., S.L. Lauritzen, and T.P. Speed, "Markov Fields and Log–Linear Models for Contingency Tables," *Annals of Statistics*, **8** (1980), 522–539.

[12] Edwards, D., and S. Kreiner, "Analysis of Contingency Tables by Graphical Models," *Biometrika* **70**,3 (1983), 553–565.

[13] Fienberg, S.E., "An Iterative Procedure For Estimation In Contingency Tables," *The Annals of Mathematical Statistics*, **41**,3 (1970), 907–917.

[14] Geman, S., "Stochastic Relaxation Methods For Image Restoration and Expert Systems," In Cooper, D.B., R.L. Launer, and E. McClure, editors, *Automated Image Analysis: Theory and Experiments*, New York: Academic Press, (to appear).

[15] Goldman, S.A., "Efficient Methods for Calculating Maximum Entropy Distributions," S.M. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, (May 1987).

[16] Graham, M.H., "On the Universal Relation," *University of Toronto Technical Report* (1979).

[17] Ireland, C.T., and S. Kullback, "Contingency tables with given marginals," *Biometrika* **55**,1 (1968), 179–188.

[18] Jaynes, E.T., "Where Do We Stand On Maximum Entropy," In Levine and Tribune, editors, *The Maximum Entropy Formalism*, M.I.T. Press, (1979).

[19] Jaynes, E.T., "On the Rationale of Maximum-Entropy Methods," *Proceedings of the IEEE*, **70**,9 (September 1982), 939–952.

[20] Johnson, R.W., and J.E. Shore, "Comments and corrections to 'Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy'," *IEEE Trans. Inform. Theory* **IT-29**, 6 (Nov. 1983), 942–943.

[21] Ku, H.H. and S. Kullback, "Approximating Discrete Probability Distributions," *IEEE Trans. on Info. Theory*, **IT-15**,4 (July 1969), 444–447.

[22] Kim, J.H. and J. Pearl, "A Computational Model for Causal and Diagnostic Reasoning in Inference Systems," *Proc. Eighth International Conference on Artificial Intelligence* (August 1983), 190–193.

[23] Lewis, P.M., "Approximating Probability Distributions to Reduce Storage Requirements," *Information and Control*, **2** (1959), 214–225.

[24] Lippman, A.F., "A Maximum Entropy Method for Expert System Construction," Ph.D. thesis, Brown University, Division of Applied Mathematics, (May 1986).

[25] Malvestuto, F.M., "Decomposing Complex Contingency Tables to Reduce Storage Requirements," *Proceedings of 3rd International Workshop on Statistical and Scientific Database Management* (July 1986), 66–71.

[26] Malvestuto, F.M., "Modeling Large Bases of Categorical Data with Acyclic Schemes," *Proceedings of the International Conference on Database Theory* (September 1986).

[27] Pearl, J., "Fusion, Propagation and Structuring in Bayesian Networks," *University Of California, Los Angeles Dept. of Computer Science Technical Report CSD-850022 R-42*, (April 1985).

[28] Rissanen, J., "A Universal Prior for Integers and Estimation by Minimum Description Length," *The Annals of Statistics*, **11**,2 (1983), 416–431.

[29] Rose, D.J. and R.E. Tarjan, "Algorithmic Aspects of Vertex Elimination in Directed Graphs," *SIAM Journal Applied Math*, **24** (1978), 176–197.

[30] Rose, D.J., R.E. Tarjan, and G.S. Lueker, "Algorithmic Aspects of Vertex Elimination on Graphs," *SIAM Journal Comput.*, **5**,2 (June 1976), 266–283.

[31] Shore, J.E., and R.W. Johnson, "Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross-Entropy," *IEEE Trans. Inform. Theory*, **IT-26**,1 (Jan. 1980), 26–37.

[32] Spiegelhalter, D.J., "Probabilistic Reasoning in Predictive Expert Systems," in *Uncertainty in Artificial Intelligence*, (eds. Kanal, L.N. and Lemmer, J.) North Holland Amsterdam, 47–68.

[33] Spiegelhalter, D.J., "Coherent Evidence Propagation in Expert Systems," to appear in the *The Statistician.*

[34] Tarjan, R.E. and M. Yannakakis, "Simple Linear-Time Algorithms to Test Chordality of Graphs, Test Acyclicity of Hypergraphs, and Selectively Reduce Acyclic Hypergraphs," *SIAM J. Comp.*, **13**,3 (August 1984), 566–579.

[35] Tikochinsky, Y., N.Z. Tishby, and R.D. Levine, "Consistent Inference of Probabilities for Reproducible Experiments," *Physical Rev. Letters* **52**, 16 (16 April 1984), 1357–1360.

[36] Tversky, A. and Kahneman, D., "Judgment Under Uncertainty: Heuristics and Biases," *Science*, **185**, (September 1974), 1124–1131.

[37] Yannakakis, M., "Computing the Minimum Fill-in is NP-Complete," *SIAM Journal Alg. Disc. Meth.*, **2**,1 (March 1981), 77–79.