Approximate Factorizations of Distributions and the Minimum Relative Entropy Principle

Heinz Mühlenbein
heinz.muehlenbein@ais.fraunhofer.de
Fraunhofer Institute for Autonomous Intelligent Systems
53754 Sankt Augustin, Germany

Robin Höns
robin.hoens@ais.fraunhofer.de
Fraunhofer Institute for Autonomous Intelligent Systems
53754 Sankt Augustin, Germany

ABSTRACT

Estimation of Distribution Algorithms (EDA) have been proposed as an extension of genetic algorithms. In this paper the major design issues of EDA's are discussed within a general interdisciplinary framework, the *maximum entropy* approximation. Our EDA algorithm *FDA* assumes that the function to be optimized is additively decomposed (ADF). The interaction graph G_{ADF} is used to create exact or approximate factorizations of the Boltzmann distribution. The relation between *FDA* factorizations and the $MaxEnt$ solution is shown. We introduce a second algorithm, derived from the *Bethe-Kikuchi* approach developed in statistical physics. It tries to minimize the Kullback-Leibler divergence $KLD(q|p_\beta)$ to the Boltzmann distribution p_β by solving a difficult constrained optimization problem. We present in detail the concave-convex minimization algorithm *CCCP* to solve the optimization problem. The two algorithms are compared using popular benchmark problems (2-d grid problems, 2-d Ising spin glasses, Kaufman's $n-k$ function.) We use instances up to 900 variables.

Categories and Subject Descriptors

F.2 [**Analysis of Algorithms and Problem Complexity**]: [Estimation of Distribution Algorithm]

General Terms

Theory, Algorithms

Keywords

Estimation of distributions, Boltzmann distribution, factorization of distributions, maximum entropy principle, minimum relative entropy, minimum log-likelihood ratio, Bayesian information criterion, Bethe-Kikuchi approximation.

GECCO'05, June 25–29, 2005, Washington, DC, USA.

1. INTRODUCTION

The *Estimation of Distribution* (EDA) family of population based search algorithms was introduced in [20] as an an extension of genetic algorithms.[1] The following observations lead to this proposal. First, genetic algorithm have difficulties to optimize deceptive and non-separable functions, and second, the search distributions implicitly generated by recombination and crossover can be extended to include the correlation of the variables in samples of high fitness values.

EDA uses probability distributions derived from the function to be optimized to generate search points instead of crossover and mutation as done by genetic algorithms. The other parts of the algorithms are identical. In both cases a population of points is used and points with good fitness are selected either to estimate a search distribution or to be used for crossover and mutation.

In [20] the distribution has been estimated by computationally intensive Monte Carlo methods. The distribution was restricted to tree-like structures. It has been shown by [19] that simpler and more effective methods exist which use a general factorization of the distribution.

The family of EDA algorithms can be understood and further developed without the background of genetic algorithms. The problem to estimate empirical distributions has been investigated independently in several scientific disciplines. In this paper we will show how results in statistics, belief networks and statistical physics can be used to understand and further develop EDA. In fact, an interdisciplinary research effort is well under way which cross-fertilizes the different disciplines.

Unfortunately each discipline uses a different language, has a slightly different application, and has developed different algorithms. In EDA we have to sample from a distribution, in belief networks one computes a single marginal distribution $p(\mathbf{y}|\mathbf{z})$ for new evidence $\mathbf{z}$, and statistical physicists want to compute the free energy of a Boltzmann distribution. Thus the algorithms developed for belief networks concentrate on computing a single marginal distribution, whereas for EDA we want to sample $p(\mathbf{x})$ in areas of high fitness values, i.e. we are interested in a sampling method which generates points with a high value of $p(\mathbf{x})$. All disciplines are interested in developing fast algorithms to compute marginal distributions. The foundation of the theory is the same for all disciplines. It is based on graphical models and their decomposition. We hope that the readers

[1] In [20] they have been named *conditional distribution algorithms.*

are interested to accompany us on our journey through the different disciplines.

Today two major branches of EDA can be distinguished. In the first branch the factorization of the distribution is computed from the structure of the function to be optimized, in the second one the structure is computed from the correlations of the data generated. The second branch has been derived from the theory of belief networks [10]. The underlying theory is the same for both branches. For large real life applications often a hybrid between these two approaches is most successful [16]. In this paper we investigate the first branch only.

The problem of computing approximations of distributions using factorization is investigated using the framework of *maximum entropy*. We distinguish exact factorizations and approximate factorizations. We shortly summarize the results for our well-known algorithm *FDA*. We present in detail a new algorithm *CCCP*. It is derived from an approach used in statistical physics to approximate the Boltzmann distribution. It is called the *Bethe-Kikuchi* approximation. In this approach the marginals from the unknown Boltzmann distribution are not computed from data, but from a difficult constrained optimization problem. This paper extends the theory first described in [14].

The different EDA algorithms are shortly numerically compared, using large benchmark problems from 2-D Ising spin glasses, and Kaufman's $n-k$ function. We investigate problems with up to 900 variables, continuing the work in [16], where graph bipartitioning problems of 1000 nodes are solved.

2. FACTORIZATION OF THE SEARCH DISTRIBUTION

EDA has been derived from a search distribution point of view. We just recapitulate the major steps published in [19, 16]. We will use in this paper the following notation. Capital letters denote variables, lower cases instances of variables. If the distinction between variables and instances is not necessary, we will use lower case letters. Vectors are denoted by $\mathbf{x}$, a single variable by x_i.
Let a function $f : \mathbf{X} \to \mathbb{R}_{\geq 0}$ be given. We consider the optimization problem

$$\mathbf{x}_{opt} = \operatorname{argmax} f(\mathbf{x}) \tag{1}$$

A good candidate for optimization using a search distribution is the Boltzmann distribution.

DEFINITION 1. *For $\beta \geq 0$ define the* Boltzmann distribution[2] *of a function $f(\mathbf{x})$ as*

$$p_\beta(\mathbf{x}) := \frac{e^{\beta f(\mathbf{x})}}{\sum_{\mathbf{y}} e^{\beta f(\mathbf{y})}} =: \frac{e^{\beta f(\mathbf{x})}}{Z_f(\beta)} \tag{2}$$

where $Z_f(\beta)$ is the partition function. To simplify the notation β and/or f might be omitted.

The Boltzmann distribution concentrates with increasing β around the global optima of the function. Obviously, the distribution converges for $\beta \to \infty$ to a distribution where only the optima have a probability greater than 0 [17]. Therefore, if it were possible to sample efficiently from this distribution for arbitrary β, optimization would be an easy task. But the computation of the partition function needs an exponential effort for a problem of n variables. We have therefore proposed an algorithm which incrementally computes the Boltzmann distribution from empirical data using Boltzmann selection.

DEFINITION 2. *Given a distribution p and a selection parameter $\Delta\beta$,* Boltzmann selection *calculates the distribution for selecting points according to*

$$p^s(\mathbf{x}) = \frac{p(\mathbf{x})e^{\Delta\beta f(\mathbf{x})}}{\sum_y p(\mathbf{y})e^{\Delta\beta f(\mathbf{y})}} \tag{3}$$

The following theorem is easy to prove.

THEOREM 3. *If $p_\beta(\mathbf{x})$ is a Boltzmann distribution, then $p^s(\mathbf{x})$ is a Boltzmann distribution with inverse temperature $\beta(t+1) = \beta(t) + \Delta\beta(t)$.*

Algorithm 1 describes *BEDA*, the Boltzmann Estimated Distribution Algorithm.

***BEDA* – Boltzmann Estimated Distribution**

1 $t \Leftarrow 1$. Generate N points according to the uniform distribution $p(\mathbf{x}, 0)$ with $\beta(0) = 0$.
2 **do** {
3 With a given $\Delta\beta(t) > 0$, let
$$p^s(\mathbf{x}, t) = \frac{p(\mathbf{x}, t)e^{\Delta\beta(t)f(\mathbf{x})}}{\sum_{\mathbf{y}} p(\mathbf{y}, t)e^{\Delta\beta(t)f(\mathbf{y})}}.$$
4 Generate N new points according to the distribution $p(\mathbf{x}, t+1) = p^s(\mathbf{x}, t)$.
5 $t \Leftarrow t + 1$.
6 } **until** (stopping criterion reached)

BEDA is a conceptional algorithm, because the calculation of the distribution requires a sum over exponentially many terms. In the next section we transform *BEDA* into a practical numerical algorithm.

2.1 Factorization of the distribution

In this section an efficient numerical algorithm is derived if the fitness function is additively decomposed.

DEFINITION 4. *Let $s_1, \ldots, s_m$ be index sets, $s_i \subseteq \{1, \ldots, n\}$. Let f_i be functions depending only on the variables x_j with $j \in s_i$. Then*

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i(\mathbf{x}_{s_i}) \tag{4}$$

is an additive decomposition *of the fitness function (ADF).*

DEFINITION 5. *Let an ADF be given. Then the interaction graph G_{ADF}[3] is defined as follows: The vertices represent the variables of the ADF . Two vertices are connected by an arc iff the corresponding variables are contained in a common sub-function.*

Given an ADF we want to estimate the Boltzmann distribution (2) using a product of marginals of low order. The approximation has to fulfill two conditions

[2]The Boltzmann distribution is usually defined as $e^{-\frac{E(\mathbf{x})}{T}}/Z$. The term $E(x)$ is called the energy and $T = 1/\beta$ the temperature. We use the inverse temperature β instead of the temperature.

[3][24] call it a decomposable Markov graph.

- The approximation should use marginals of low order.
- Sampling from the approximation should be easy.

A class of distributions fulfilling these conditions are the acyclic Bayesian network (acBN).

$$q(\mathbf{x}) = \prod_{i=1}^{n} q(x_i|\pi_i) \tag{5}$$

where π_i are called the parents of x_i. For acyclic Bayesian networks sampling can easily be done starting with the root x_1. Cyclic Bayesian networks need a complex sampling procedure.

Note that any distribution can be written in the form of an acyclic Bayesian network because of

$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)p(x_3|x_1,x_2)\cdots p(x_n|x_1,\ldots,x_{n-1}) \tag{6}$$

But this factorization uses marginal distributions of size $O(n)$, thus sampling from the distribution is exponential in n. Therefore we are looking for factorizations where the size of the marginals is bounded, hopefully independent of n.

For *ADF*'s the following procedure can be used to create factorizations. We need the following sets:

DEFINITION 6. *Given $s_1,\ldots,s_m$, we define for $i = 1,\ldots,m$ the sets d_i, b_i and c_i:*

$$d_i := \bigcup_{j=1}^{i} s_j, \qquad b_i := s_i \setminus d_{i-1}, \qquad c_i := s_i \cap d_{i-1} \tag{7}$$

We demand $d_m = \{1,\ldots,n\}$ and set $d_0 = \emptyset$. In the theory of decomposable graphs, d_i are called histories, *b_i* residuals *and c_i* separators *[11].*

The next definition is stated a bit informally.

DEFINITION 7. *A set of marginal distributions $\tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})$ is called* consistent *if the marginal distributions fulfill the laws of probability, e.g.*

$$\sum_{\mathbf{x}_{b_i},\mathbf{x}_{c_i}} \tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}) = 1 \tag{8}$$

$$\sum_{\mathbf{x}_{b_i}} \tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}) = \tilde{q}(\mathbf{x}_{c_i}) \tag{9}$$

PROPOSITION 8. *Let a consistent set of marginal distributions $\tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})$ be given. If $b_i \neq \emptyset$ then*

$$q(\mathbf{x}) = \prod_{i=1}^{m} \tilde{q}(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}) \tag{10}$$

defines a valid distribution ($\sum q(\mathbf{x}) = 1$). Furthermore

$$q(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}) = \tilde{q}(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}), \quad i = 1,\ldots m \tag{11}$$

whereas in general

$$q(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}) \neq \tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}), \quad i = 1,\ldots m \tag{12}$$

The proof follows from the definition of marginal probabilities. The proof of equation (11) is somewhat technical, but straightforward. The inequality (12) is often overlooked. It means that sampling from the factorization does not reproduce the given marginals.

DEFINITION 9. *Equation* (10) *defines an FDA factorization for a given ADF.*

Remark: *Any FDA factorization can easily be transformed into an acyclic Bayesian network. Therefore the class of FDA factorizations is identicak to the class of acyclic Bayesian networks.*

The next theorem was proven in [19]. It shows when $q(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}) = \tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})$ is true for a *FDA* factorization.

THEOREM 10 (FACTORIZATION THEOREM). *Let $f(\mathbf{x}) = \sum_{i=1}^{m} f_{s_i}(\mathbf{x})$ be an additive decomposition. If*

$$\forall i = 1,\ldots,m; \quad b_i \neq \emptyset \tag{13}$$

$$\forall i \geq 2\, \exists j < i \;\; \textit{such that } c_i \subseteq s_j \tag{14}$$

then

$$p_\beta(\mathbf{x}) = \prod_{i=1}^{m} p_\beta(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}) = \frac{\prod_{i=1}^{m} p_\beta(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})}{\prod_{i=2}^{m} p_\beta(\mathbf{x}_{c_i})} \tag{15}$$

The Factorization Theorem shows that under certain conditions the Boltzmann distribution can exactly be represented by a product of conditional marginals.

DEFINITION 11. *The constraint defined by equation* (14) *is called the* running intersection property (*RIP*). *The factorization is* polynomially bounded *(PBF) if the size of the sets $\{b_i, c_i\}$ is bounded by a constant independent of n.*

The above theorem does not answer the question how to compute a good or even an exact factorization. The construction defined by equation 7 depends on the sequence $s_1,\ldots,s_m$. If the sequence is permutated, it might be possible that the *RIP* will be fulfilled, even if it was not fulfilled before. Furthermore, we can join two or more sub-functions, resulting in larger sets $\tilde{s}_i$. It might be that using these larger sets, the factorization becomes exact.

Testing all these combinations is prohibitive. Actually, it turns out that the computation of exact factorization is done better by investigating the corresponding interaction graph of the *ADF*. A well-known algorithm algorithm computes *junction tress* [9]. It obtains an exact factorization with marginals of small size and fulfilling the *RIP* given an arbitrary graph. A short description can be found in [14]. The largest clique of the junction tree gives the largest marginal of the factorization.

The space complexity of factorizations has been investigated in [5]. Many applications are defined on grids. This means that the interaction graph is a grid. Unfortunately, exact factorizations of 2-D grids are not bounded polynomially. Thus for larger problems, FDA has to use approximate factorizations. In the next section some 2-D grid factorizations are discussed.

2.2 Factorizations of 2-D grids

Let there be a 2-D grid of variables $x_{i,j}$, $i,j = 1,\ldots,n$. Let the fitness function be composed of the sub-functions of pairs of neighboring variables, $x_{i,j}, x_{i+1,j}$ and $x_{i,j}, x_{i,j+1}$. The goal is to compute a factorized distribution which is a good approximation to the true distribution.

An exact factorization can be found with a junction tree. The difficulty of the computation lies in the triangulation of the graphical model. One valid triangulation uses the rows of the grid. Each variable is connected with all variables in the same row and the neighboring rows. This adds $O(n)$ edges to the graph. The cliques in the junction tree consist

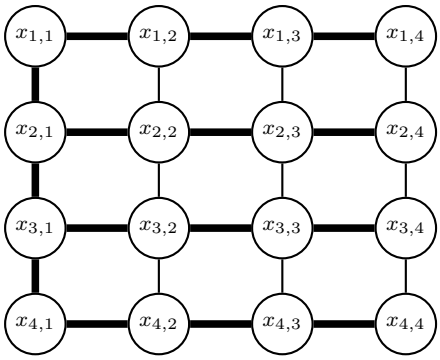

Figure 1: Graph model for a 2-D grid. The thick lines give a possible spanning tree

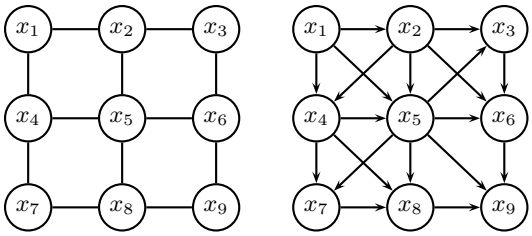

Figure 2: A 3×3 grid and its factorization using (17).

of pairs of neighboring rows and have size $2n$. Thus the exact factorization is not polynomially bounded.

Therefore it is advisable to look for approximations. A very straightforward approximation is to leave out some of the marginals and build a spanning tree of the grid. This could be the vertical edges in the first column and all the horizontal edges, forming a big "E" (see thick lines in figure 1).

Given this subset of the edges and disregarding the rest, we can define the following distribution:

$$q(\mathbf{x}) = p(x_{1,1}, x_{2,1}) \prod_{i=2}^{n-1} p(x_{i+1,1}|x_{i,1}) \prod_{i=1}^{n} \prod_{j=1}^{n-1} p(x_{i,j+1}|x_{i,j}) \tag{16}$$

This is a valid probability distribution insofar as it sums up to 1 and complies with the regarded marginals. But obviously the choice of some marginals, while forgetting the rest, retains the stain of arbitrariness. Another possibility which regards all the given marginals, consists of combining blocks of four variables $(x_{i,j}, x_{i+1,j}, x_{i,j+1}, x_{i+1,j+1})$. The complete distribution can then be built up by:

$$\begin{aligned} q(\mathbf{x}) &= p(x_{1,1}, x_{2,1}, x_{1,2}, x_{2,2}) \\ &\prod_{i=2}^{n-1} p(x_{i+1,1}, x_{i+1,2}|x_{i,1}, x_{i,2}) \prod_{j=2}^{n-1} p(x_{1,j+1}, x_{2,j+1}|x_{1,j}, x_{2,j}) \\ &\prod_{i=2}^{n-1} \prod_{j=2}^{n-1} p(x_{i+1,j+1}|x_{i,j}, x_{i+1,j}, x_{i,j+1}) \end{aligned} \tag{17}$$

However, the factorization (17) violates the running intersection property (14). It reproduces the given marginals only in the first tetra-variate row and column, but not in the areas where the running intersection property is violated. We call the factorization (17) $G4$. An extension is the factorization $G5$. It uses marginals up to order 5. For the 3*3 grid it is given by

$$\begin{aligned} q(\mathbf{x}) = p(x_1, x_2, x_4, x_5) p(x_3, x_6|x_4, x_5) \\ p(x_7, x_8)|x_4, x_5, x_6) p(x_9|x_5, x_6, x_8) \end{aligned} \tag{18}$$

The optimal decomposition of a grid has already been investigated for non-serial dynamic programming by [12]. Any decomposition fulfilling the *RIP* needs marginals of order $O(n)$. Thus the exact decomposition can be used for small grids only.

We next describe our factorized distribution algorithm *FDA* which also works with approximate factorizations.

2.3 The Factorized Distribution Algorithm

If the factorization violates the assumption of the factorization theorem, then non-serial dynamic programming does not work. But an algorithm which estimates the marginals from samples might still find the optimum. One only has to compute a good approximate factorization given the graph G_{ADF}. We first describe our algorithm *FDA*.

***FDA* – Factorized Distribution Algorithm**

1. Calculate b_i and c_i by the Sub-function Merger Algorithm.
2. $t \Leftarrow 1$. Generate an initial population with N individuals from the uniform distribution.
3. **do** {
4. Select $M \leq N$ individuals using Boltzmann selection[a] (see definition 2).
5. Estimate the conditional probabilities $p(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}, t)$ from the selected points.
6. Generate new points according to $p(\mathbf{x}, t+1) = \prod_{i=1}^{m} p(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}, t)$.
7. $t \Leftarrow t+1$.
8. } **until** (stopping criterion reached)

[a] The algorithm works with any selection method

We next describe the sub-function merger algorithm which computes the *FDA* factorization. It is a simple heuristic, trying to cover many edges of the interaction graph by merging of sub-functions. Let us first discuss the assumption $b_i \neq \emptyset$ of the factorization theorem. This assumption is violated already for the loop

$$s_1 = \{1,2\}, s_2 = \{2,3\}, s_3 = \{1,3\}$$

All possible sequences end in $b_3 = \emptyset$ because the variables of the sub-function left are already contained in the two previous sets. One possibility to solve this problem is to choose only a subset of the s_i and disregard the others; in our example, we can use the factorization $q(\mathbf{x}) = p(x_1, x_2)p(x_3|x_2)$ using s_1 and s_2. An exact factorization is

$$p(\mathbf{x}) = p(x_1, x_2)p(x_3|x_2, x_1)$$

This factorization will be generated if the two sub-functions s_2 and s_3 are merged. This observation leads to the idea to compute approximate factorizations by merging of sub-functions[4].

[4] [3] have called it fusion.

Sub-function Merger

```
1  S ⇐ {s_1, ..., s_m}
2  j ⇐ 1
3  while d̃_j ≠ {1, ..., n} do {
4     Chose an s_i ∈ S to be added
5     S ⇐ S \ {s_i}
6     Let the indices of the new variables in s_i be b_i =
      {k_1, ..., k_l}
7     for λ = 1 to l do {
8        δ_λ ⇐ {k ∈ d̃_{j-1} | (x_k, x_{k_λ}) ∈ G_ADF}
9     }
10    for λ = 1 to l do {
11       if exists λ' ≠ λ with δ_λ ⊆ δ_λ' and k_λ' not
         marked superfluous
12          δ_λ' ⇐ δ_λ' ∪ {k_λ}
13          Mark k_λ superfluous
14    }
15    for λ = 1 to l do {
16       if not k_λ superfluous
17          s̃_j ⇐ δ_λ ∪ {k_1, ..., k_λ}
18          j ⇐ j + 1
19    }
20 }
```

A good merging heuristic tries to minimize the number of mergers but simultaneously to use all dependencies in G_{ADF}. Thus the heuristic generates graphs with $b_i \neq \emptyset$ which keep the number of dependencies as low as possible.

Algorithm 3 describes our heuristic. The idea of the sub-function merger algorithm is that each new variable is included in a set together with the previous variables on which it depends. However, if another variable depends on a superset of variables, the two sets are merged. After completing the merge phase, the algorithm calculates $\tilde{c}_j$, $\tilde{b}_j$ and $\tilde{d}_j$ analogous to the construction given by (7).

This sub-function merger algorithm might still compute too large cliques. Therefore a cut parameter k is needed which bounds the clique size. If the size of a clique becomes larger than k our implementation will randomly leave out arcs from G_{ADF}. For 2-D grid problems where the ADF consists of functions of two variables only, the sub-function merger algorithm uses marginals up to order 3. The factorization covers the interaction G_{ADF}. For the 3*3 grid shown in figure 2 the sub-function merger constructs the following factorization:

$$p(\mathbf{x}) = p(x_5, x_6)p(x_4|x_5)p(x_3|x_6)p(x_2|x_3, x_5) \\ p(x_8|x_5)p(x_1|x_2, x_4)(x_9|x_6, x_8)(x_7|x_4, x_8) \quad (19)$$

Our presentation of the sub-function merger algorithm has been very short. The interested reader is referred to [3] for an in depth discussion of different fusion and folding heuristics. In the area of Bayesian networks, the problem has been investigated by [2].

If the conditions of the factorization theorem are fulfilled, the convergence proof of *BEDA* is valid for *FDA*, too. Since *FDA* uses finite samples of points to estimate the conditional probabilities, convergence to the optimum will depend on the size of the sample. For small sample sizes the convergence rate is higher if a number of steps with low selection is used instead of just one step using strong selection. Thus this method is numerically more efficient than to use a very large sample size and strong selection.

FDA has experimentally proven to be very successful on a number of functions where standard genetic algorithms fail to find the global optimum. In [15] the scaling behavior for various test functions has been studied. For recent surveys the reader is referred to [16, 18, 14].

We next want to put the *FDA* factorizations in a broader perspective, especially the approximate factorizations.

3. THE MAXIMUM ENTROPY PRINCIPLE

In this section we investigate the problem of approximating an unknown distribution given some information in a theoretical framework.

Let $\mathbf{x} = (x_1, \ldots, x_n)$, $B = \{0, 1\}^n$. Let $\phi_j : B \to \{0, 1\}, j = 1, k$ be binary functions, often called features. Let a sample S be given, $\tilde{p}(\mathbf{x})$ the observed distribution. Let

$$E_{\tilde{p}}(\phi_j) = \sum_{x \in B} \tilde{p}(\mathbf{x})\phi_j(\mathbf{x}) \quad (20)$$

Note that ϕ_j can specify any marginal distribution, but also more general expectations.

Problem *We are looking for a distribution which fulfills the constraints*

$$E_p(\phi_j) = E_{\tilde{p}}(\phi_j) \quad (21)$$

and is in some sense plausible.

If only a small number of features is given the problem is under-specified. Consequently, for incomplete specifications the missing information must be added by some automatic completion procedure. This is achieved by the *maximum entropy principle*. Let us recall

DEFINITION 12. *The entropy [4] of a distribution is defined by*

$$H(p) = -\sum_x p(\mathbf{x}) \ln(p(\mathbf{x})) \quad (22)$$

Maximum entropy principle (MaxEnt): *Let*

$$P = \{p | E_p(\phi_j) = E_{\tilde{p}}(\phi_j), \quad j = 1, \ldots, k\} \quad (23)$$

Then the MaxEnt solution is given by

$$p^* = \text{argmax}_{p \in P} H(p) \quad (24)$$

The maximum entropy principle formulates the *principle of indifference.* If no constraints are specified, the uniform random distribution is assumed. MaxEnt has a long history in physics and probabilistic logic. The interested reader is referred to [7, 8]. MaxEnt is especially attractive because there exists a constructive way to obtain the solution.

The MaxEnt solution is obtained from the constrained optimization problem

$$p^* = \text{argmax}_{p \in P} H(p) \quad (25)$$

$$\sum_x p(x) = 1 \quad (26)$$

$$\sum_x p(\mathbf{x})\phi_j(\mathbf{x}) = E_{\tilde{p}}(\phi_j) \quad (27)$$

This is a convex optimization problem with linear constraints. It can be solved by introducing Lagrange multipliers.

$$L(p,\Lambda,\gamma) = -\sum_x p(\mathbf{x})\ln(p(\mathbf{x}) + \gamma(\sum_x p(x) - 1) \quad (28)$$

$$+ \sum_{i=1}^{k} \lambda_i (E_p(\phi_j) - E_{\tilde{p}}(\phi_j)) \quad (29)$$

where $\Lambda = (\lambda_1, \ldots, \lambda_k)$.

The maxima of L can be obtained by setting the derivatives of L to zero. We obtain

$$\frac{\partial L}{\partial p(\mathbf{x})} = -\ln p(\mathbf{x})) - 1 - \sum_{j=1}^{k} \lambda_j \phi_j(\mathbf{x}) + \gamma \quad (30)$$

Setting the derivative to zero gives the parametric form of the solution

$$p^*(\mathbf{x}) = \exp(1-\gamma) \exp \sum_{j=1}^{k} \lambda_j \phi_j(\mathbf{x}) \quad (31)$$

DEFINITION 13. *Let Q be the space of distributions of the parametric form*

$$Q = \{q | q(\mathbf{x}) = \frac{1}{Z} \exp \sum_{j=1}^{k} \lambda_j \phi_j(\mathbf{x}) \quad (32)$$

In order to characterize the MaxEnt solution, the relative entropy between distributions has to be introduced.

DEFINITION 14. *The relative entropy or Kullback-Leibler divergence between two distributions p and q is defined as*

$$KLD(p,q) = \sum_x p(\mathbf{x}) \ln \frac{p(\mathbf{x})}{q(\mathbf{x})} \quad (33)$$

Note that $KLD(p,q) \neq KLD(q,p)$, i.e. KLD is not symmetric! If $q(\mathbf{x}) = 0$ and $p(\mathbf{x}) > 0$ we have $KLD(p,q) = \infty$. This means that KLD gives large weights to values near zero. In all other aspects KLD is a distance measure. The following lemma holds ([4]).

LEMMA 15. *For any two probability distributions p and q, $KLD(p,q) \geq 0$ and $KLD(p,q) = 0$ iff $p = q$.*

In our application KLD fulfills the Pythagorean property.

LEMMA 16 (PYTHAGOREAN PROPERTY). *Let $p \in P$, $q \in Q$, and $p^* \in P \cap Q$, then*

$$KLD(p,q) = KLD(p,p^*) + KLD(p^*,q) \quad (34)$$

PROOF. Let $r, s \in P$, $q \in Q$. Then

$$\sum_x r(\mathbf{x}) \ln q(\mathbf{x}) = \sum_x s(\mathbf{x}) \ln q(\mathbf{x})$$

Now let $p \in P$, $qinQ$, and $p^* \in P \cap Q$. Then

$$\begin{aligned} & KLD(p,p^*) + KLD(p^*,q) \\ = \; & KLD(p,p^*) + \sum_x p^*(\mathbf{x}) \ln p^*(\mathbf{x}) - \sum_x p^*(\mathbf{x}) \ln q(\mathbf{x}) \\ = \; & KLD(p,p^*) + \sum_x p(\mathbf{x}) \ln p^*(\mathbf{x}) - \sum_x p(\mathbf{x}) \ln q(\mathbf{x}) \\ = \; & KLD(p,q) \end{aligned}$$

□

The following theorem follows easily from the lemma:

THEOREM 17 (MAXIMUM ENTROPY SOLUTION). *If $p^* \in P \cap Q$, then*

$$p^*(\mathbf{x}) = \text{argmax}_{p \in P} H(p) \quad (35)$$

Furthermore, p^* is unique.

The constrained optimization problem can be solved by standard mathematical algorithms. But also specialized algorithms have been invented, a popular one is *Iterative Proportional Fitting* (*IPF*). It is used if the features define marginals. IPF iteratively computes a distribution $q_\tau(\mathbf{x})$ from the given marginals $p_k(\mathbf{x}_k)$, $k = 1, \ldots, K$, where $\mathbf{x}_k$ is a sub-vector of $\mathbf{x}$ and $\tau = 0, 1, 2, \ldots$ is the iteration index. Let n be the dimension of $\mathbf{x}$ and d_k be the dimension of $\mathbf{x}_k$. $q_{\tau=0}$ is the uniform distribution. The update formula is

$$\forall \mathbf{x} \quad q_{\tau+1}(\mathbf{x}) = q_\tau(\mathbf{x}) \frac{p_k(\mathbf{x}_k)}{\sum_{y \in \{0,1\}^{n-d_k}} q_\tau(\mathbf{x}_k, \mathbf{y})} \quad (36)$$

with $k = ((\tau - 1) \mod K) + 1$.

Since the distribution q, which has to be stored and updated in every time step, has exponential size, this implementation takes exponential time and space.

There exist another justification of the MaxEnt solution, it is given by the *Maximum Log-Likelihood* principle.

DEFINITION 18. *Let $S = \{X_1, \ldots, X_N\}$ be an empirical sample, $\tilde{p}(\mathbf{x})$ the empirical distribution. let $q(\mathbf{x})$ be a distribution. Then the likelihood that q generates the data is given by*

$$LH(q) = \prod_{i=1}^{N} q(X_i) = \prod_{x \in B} q(\mathbf{x})^{N\tilde{p}(\mathbf{x})} \quad (37)$$

The log-likelihood is defined as

$$LogLH(q) = \sum_{x \in B} N\tilde{p}(\mathbf{x}) \ln q(\mathbf{x}) \quad (38)$$

THEOREM 19 (MAXIMUM LOG-LIKELIHOOD SOLUTION). *If $p^* \in P \cap Q$, then*

$$p^*(\mathbf{x}) = \text{argmax}_{q \in Q} LogLH(q) \quad (39)$$

Furthermore, p^* is unique.

PROOF. Let $\tilde{p}(\mathbf{x})$ be the observed distribution. Clearly $\tilde{p} \in P$. Suppose $q \in Q$ and $p^* \in P \cap Q$. We show that $L(q) \leq L(p^*)$. The Pythagorean property gives

$$KLD(\tilde{p},q) = KLD(\tilde{p},p^*) + KLD(p^*,q)$$

Therefore

$$\begin{aligned} KLD(\tilde{p},q) &\geq KLD(\tilde{p},p^*) \\ -H(\tilde{p}) - LogLH(q) &\geq -H(\tilde{p}) - LogLH(p^*) \\ LogLH(q) &\leq LogLH(p^*) \end{aligned}$$

□

Thus the MaxEnt solution can be viewed under both the maximum entropy framework as well as the maximum log-likelihood framework. This means that p^* will fit the data as closely as possible while as the maximum entropy solution will not assume facts beyond those given by the constraints.

We next investigate the relation of *FDA* factorizations and the MaxEnt solution.

DEFINITION 20. *The MaxEnt problem is called complete marginal if all marginal distributions $p(x_{s_k})$ are given. The FDA factorization is called complete, if the corresponding graphical model contains the interaction graph.*

THEOREM 21. *The MaxEnt solution of a complete marginal MaxEnt problem is the exact distribution. The MaxEnt solution of any complete FDA factorization is the exact distribution.*

PROOF. Let a complete marginal MaxEnt problem be given. Then the features $\phi(\mathbf{x}_{s_i})$ are defined by $E_{\tilde{p}}\phi(\mathbf{x}_{s_i}) = \tilde{p}(\mathbf{x}_{s_i})$. We abbreviate the parameters in equation 32 by $\lambda(\mathbf{x}_{s_i})$. Now set $\lambda(\mathbf{x}_{s_i})\tilde{p}(\mathbf{x}_{s_i}) = \beta f(\mathbf{x}_{s_i})$. Thus the exact distribution is in the set Q. Obviously the exact distribution fulfills the marginalization constraints. Therefore the exact distribution is the MaxEnt solution. The proof for complete *FDA* factorizations works accordingly. □

This theorem is another justification of the MaxEnt principle. If all relevant information is given, then the unique MaxEnt solution is the exact distribution. We obtain the following corollary.

COROLLARY 22. *The MaxEnt solution of any complete FDA factorization fulfilling the RIP is identical to the distribution specified by the FDA factorization.*

Thus the best approximation seems to be the MaxEnt solution. But the computation of the MaxEnt solution for complete *FDA* factorizations which do not fulfill the *RIP* is exponential. Furthermore, sampling from the MaxEnt solution is computationally expensive. Therefore we use the *FDA* factorization as the approximation. Sampling from a *FDA* factorization is easy, but even for complete factorizations the generated distribution is different from the exact distribution, if the *RIP* is violated.

We next turn to another approach to approximate the Boltzmann distribution. In this case the Kullback-Leibler divergence is minimized without computing the marginal distributions from samples. Instead the best values are computed from a constrained minimization problem.

4. APPROXIMATING THE BOLTZMANN DISTRIBUTION

The Boltzmann distribution plays an important role in statistical physics. Therefore a number of approximation techniques have been tried. The idea is the compute an approximation q using marginals of low order

$$q(\mathbf{x}) = \frac{1}{Z}\prod_{i=1}^{k}\tilde{q}(\mathbf{x}_k) \tag{40}$$

which minimizes some distance to the Boltzmann distribution. The approach using the Kullback-Leibler distance is described in [14] using the terminology of physics. We give here a short mathematical derivation.

$$\begin{aligned} KLD(q|p_\beta) &= \sum_x q(\mathbf{x})\ln q(\mathbf{x}) - \sum_x q(\mathbf{x})\ln p_\beta(\mathbf{x}) \\ &= -H(q) + \ln Z - \beta E_q(f) \end{aligned}$$

We now assume that the function is defined by an *ADF*. Then we easily obtain

$$E_q(f) = \sum_{i=1}^{m} q(\mathbf{x}_{s_i}) f_i(x_{s_i}) \tag{41}$$

The expected average of the function can be computed using the marginals. The difficult problem is the computation of $H(q)$. We will restrict our discussion to *FDA* factorizations. In the simplest case we use

$$q(\mathbf{x}) = \frac{\tilde{q}(\mathbf{x}_{s_i})}{\tilde{q}(\mathbf{x}_{c_i})} \tag{42}$$

For this factorization one computes

$$H(q) = -\sum_{i=1}^{m} q(\mathbf{x}_{s_i}\ln\tilde{q}(\mathbf{x}_{s_i}) + q(\mathbf{x}_{c_i})\ln\tilde{q}(\mathbf{x}_{c_i}) \tag{43}$$

Thus we arrive at the following constraint optimization problem.

DEFINITION 23 (BETHE-KIKUCHI APPROXIMATION).

$$\begin{aligned} \boldsymbol{argmin}_q KLD(q|p_\beta) = \sum_{i=1}^{m}(q(\mathbf{x}_{s_i})\ln q(\mathbf{x}_{s_i}) \\ - q(\mathbf{x}_{c_i})\ln q(\mathbf{x}_{c_i})) - \beta\sum_{i=1}^{m} q(\mathbf{x}_{s_i}) f_i(x_{s_i}) \end{aligned} \tag{44}$$

subject to the constraints for all s_j with $c_i \subset s_j$

$$\sum_{x_{s_i}} q(\mathbf{x}_{s_i}) = 1 \tag{45}$$

$$\sum_{x_{s_j}\setminus x_{c_i}} q(\mathbf{x}_{s_j}) = q(\mathbf{x}_{c_i}) \tag{46}$$

Remark: Before we discuss how to solve the minimization problem, we want to mention that the *minimization problem is not convex! There might exist many local minima.* This leads to the following important difference between the MaxEnt solution and the Kikuchi approximation. Theorem 21 shows that the MaxEnt solution is exact if a set of marginals is given which covers the interaction graph. In contrast, the Bethe-Kikuchi approximation gives not the unknown distribution if the *RIP* is not fulfilled.

The constraints make the the solution problem difficult. As already done for the MaxEnt solution we introduce the Lagrange function

$$\begin{aligned} L(p,\Lambda,\Gamma) = KLD(q|p_\beta) + \sum_{i=1}^{m}\gamma_i\sum_{x_{s_i}}(q(\mathbf{x}_{s_i}) - 1) \\ + \sum_{i=1}^{m}\sum_{x_{c_i}}(\lambda(s_j,c_i)\sum_{x_{s_j}\setminus x_{c_i}}(q(\mathbf{x}_{s_j}) - q(\mathbf{x}_{c_i})) \end{aligned} \tag{47}$$

The minima of L are determined be setting the derivatives of L zero. The independent variables are $(q(\mathbf{x}_{s_i}),q(\mathbf{x}_{c_i}),\gamma_i$,and $\lambda(s_j,c_i)$. We obtain

$$\frac{\partial L}{\partial q(\mathbf{x}_{s_i})} = \ln q(\mathbf{x}_{s_i}) + 1 - \beta q(\mathbf{x}_{s_i}) f_(\mathbf{x}_{s_i}) + \gamma_i + r(\Lambda) \tag{48}$$

Setting the derivative to zero, we obtain the parametric form

$$q(\mathbf{x}_{s_i}) = e^{-1-\gamma_i} e^{-r(\Lambda)} e^{\beta f(x_{s_i})} \tag{49}$$

Note that the parametric form is again exponential. The Lagrange factors Γ are easily computed from $\sum_{x_{s_i}} q(x_{s_i}) = 1$. The factors Λ have to be determined from a non-linear system of equation. Before we describe an algorithm for solving this equation, we describe a simple special case, the mean-field approximation.

4.1 The mean-field approximation

In the mean-field approximation uni-variate marginals only are determined.

$$q(\mathbf{x}) = \prod_{i=1}^{n} q(x_i) \tag{50}$$

Then we can compute its entropy and $E_q(f)$.

$$\begin{aligned} H(q) &= -\sum_x \prod_{i=1}^n q(x_i) \sum_{j=1}^n \ln q(x_j) \\ &= -\sum_{x_1} q(x_1) \ln q(x_1) - \sum_{x_2,..x_n} \prod_{i=2} q(x_i) \sum_{j=2} \ln q(x_j) \\ &= -\sum_{i=1}^n \sum_{x_i} q(x_i) \ln q(x_i) \\ E_q(f) &= \sum_x \prod_{i=1}^n q(x_i) f(\mathbf{x}) \\ &= \sum_{i=1}^m \prod_{j \in s_i}^n q(x_j) f(\mathbf{x}_{s_i}) \end{aligned}$$

We can now try to find a minimum by setting the derivative of KLD equal to zero, using the uni-variates as variables. We abbreviate $q_i = q(x_i = 1)$. For the mean-field approximation the minimization problem is convex, therefore the minimum exists and is unique.

THEOREM 24. *The mean-field approximation minimizes the Kullback-Leibler divergence to the Boltzmann distribution. The local minima of the divergence are given by the nonlinear equation*

$$q_i^* = \frac{1}{1 + e^{\frac{\partial E_q}{\partial q_i}}} \tag{51}$$

PROOF. We compute the derivative

$$\frac{\partial KLD}{\partial q_i} = \ln \frac{q_i}{1 - q_i} + \frac{\partial E_q}{\partial q_i} = 0 \tag{52}$$

The solution gives (51). □

Equation 51 can be solved by an iteration scheme.

Remark: In the mean-field approximation the univariate marginals are considered to be variables. The minimization problem is convex, therefore a unique solution exist.

If higher-order marginals are used in the approximation, then the minimization problem might have many local minima.

5. LOOPY BELIEF MODELS AND REGION GRAPHS

The solution of the minimization problem is difficult. We decided not to use a general mathematical minimization algorithm, but to modify a specialized algorithm, recently proposed in[26]. It is based on the concept of a region graph. A region graph is a loopy graphical model. It is strongly related to partially ordered sets (posets) or Hasse diagrams. Similar or identical structures have been presented in [1, 13, 23]. This section follows largely the notation of [26].

The original Kikuchi factorization is a loopy model, therefore different from the *FDA* factorization. Therefore sampling from the Kikuchi factorization is difficult. This is the reason that the *FDA* factorization has no loops. The Kikuchi factorization and the concept of region graph has also been used for an EDA algorithm by [22]. But the marginals are not determined from minimization of the Kullback-Leibler divergence, they are estimated from samples.

5.1 Regions

The region graph is introduced in [26] using another graphical model, the *factor graph*. The factor graph is a more detailed way to describe an additive decomposition. The factor graph is not introduced here. Therefore some expressions will be more clumsy.

DEFINITION 25. *Let $S = \{s_1, \ldots, s_m\}$ be the index set of an additive decomposition for a fitness function f, such that*

$$f(\mathbf{x}) = \sum_{s_i \in S} f_i(\mathbf{x}_{s_i}) \tag{53}$$

*A **region** $R = (s_R, I_R)$ is a set of variable indices $s_R \subseteq \{1, \ldots, n\}$ and a set of sub-function indices $I_R \subseteq \{1, \ldots, m\}$, such that*

$$\forall i \in I_R : s_i \subseteq s_R \tag{54}$$

The variables contained in the region are indexed by s_R, whereas I_R contains the indices of the sub-functions which are contained in the region. It is asserted by (54) *that all variables needed for the contained sub-functions are in s_R.*

We keep in mind our goal to approximate the Boltzmann distribution with the energy $E(\mathbf{x}) = -f(\mathbf{x})$. For a region, we can define a local energy.

DEFINITION 26. *For a region R, define the **region energy***

$$E_R(\mathbf{x}_{s_R}) := -\sum_{i \in I_R} f_i(\mathbf{x}_{s_i}) \tag{55}$$

Region energies are defined only for those regions which contain the variables of at least one sub-functions. Similar to the mean-field approach, we define a local approximation of the objective Boltzmann distribution on a region, q_R. In [26] this local distribution is called the *belief* on R.

5.2 Region Graph

DEFINITION 27. *A **region graph** is a graph $G = (\mathcal{R}, E_\mathcal{R})$, where $\mathcal{R}$ is a set of regions and $E_\mathcal{R}$ is a set of directed edges. An edge $(R_p, R_c) \in E_\mathcal{R}$ is only allowed if $s_{R_c} \subset s_{R_p}$. If $(R_p, R_c) \in E_\mathcal{R}$, we call R_p a parent of R_c and R_c child of R_p.*

Since $E_\mathcal{R}$ imposes a partial ordering on the set of regions, in [13] the same structure was called a partially ordered set or *poset*.

LEMMA 28. *A region graph is directed acyclic.*

PROOF. This follows immediately from the requirement that edges are only allowed from supersets to subsets. □

A junction tree can be turned into a region graph by creating a region for every cluster and every separator and adding edges from each node to each neighboring separator.

The global distribution of a junction tree is the product of all distributions on the clusters ([14]), divided by the distributions of all the separators. We generalize this concept too, by introducing counting numbers of the regions.

DEFINITION 29. *The **counting number** c_R of a region R is defined recursively as*

$$c_R = 1 - \sum_{R' \in A(R)} c_{R'} \tag{56}$$

where $A(R)$ is the set of all ancestors of R.

This is well-defined, because the region graph is cycle-free. The maximal regions (without ancestors) have counting number 1. From there, the counting numbers can be calculated from the top to the bottom of the graph.

5.3 Region Graph and Junction Tree

If the region graph is derived from a junction tree, with the q_R being the local distributions on the clusters and separators, $k(\mathbf{x})$ is a valid distribution, since its definition coincides with the junction tree distribution.

The junction property also has an equivalent on the region graph.

DEFINITION 30. *We call a region graph **valid** if it fulfills the **region graph condition**, which states that*

1. *For all variable indices $i \in \{1, \dots, n\}$ the set $\mathcal{R}_{X,i} := \{R \in \mathcal{R} | i \in s_R\}$ of all regions R that contain X_i form a connected subgraph with*

$$\sum_{R \in \mathcal{R}_{X,i}} c_R = 1 \ , \tag{57}$$

   and

2. *For all sub function indices $i \in \{1, \dots, m\}$ the set $\mathcal{R}_{f,i} := \{R \in \mathcal{R} | i \in I_R\}$ of all regions R that contain f_i form a connected subgraph with*

$$\sum_{R \in \mathcal{R}_{f,i}} c_R = 1 \ . \tag{58}$$

The connectivity of the subgraph, like the junction property, prevents that in different parts of the graph contradictory beliefs evolve. The condition on the counting numbers makes sure that every variable and every sub-function is counted exactly once.

In a junction tree it is often the case that several separators contain the same variables. In [9] it was proposed to replace these by a single separator that is connected to all clusters in which it is contained. Then care must be taken that the local distribution p_S of such a separator S is counted the appropriate number of times. This can also be done in the region graph. The definition of counting numbers and the Kikuchi approximation ensure that the distribution is divided by p_S the appropriate number of times.

Furthermore, [9] proposes separators not only between clusters, but also between other separators. They call the resulting tree an *Almond tree*. This is another step of generalizing the junction tree towards the region graph, and it is straightforward to do this with our region graph definition, too.

In fact, it has been proven that cycle-free region graphs give exact results. In the following an adaptation of the proof for our notation is presented. For this we prove some lemmata.

LEMMA 31. *In a cycle-free region graph, the counting numbers are*

$$c_R = 1 - |\Pi_R| \tag{59}$$

where $|\Pi_R|$ is the number of parents of the region R.

PROOF. The proof exploits the fact that the sets of ancestors for all parents of a node R are disjunct.

$$\begin{aligned}
c_R &= 1 - \sum_{R' \in A(R)} c_{R'} && (60)\\
&= 1 - \sum_{R' \in \Pi_R} \left(c_{R'} + \sum_{R'' \in A(R')} c_{R''} \right) && (61)\\
&= 1 - \sum_{R' \in \Pi_R} \left(1 - \sum_{R'' \in A(R')} c_{R''} + \sum_{R'' \in A(R')} c_{R''} \right) && (62)\\
&= 1 - \sum_{R' \in \Pi_R} 1 && (63)\\
&= 1 - |\Pi_R| && (64)
\end{aligned}$$

□

LEMMA 32. *Cycle-free region graphs originating from junction trees are valid.*

PROOF. It follows immediately from the junction property that the subgraph $\mathcal{R}_{X,i}$ of the region graph that contains a variable X_i is connected. For every region R that the subgraph contains, it contains also all its parents Π_R. Since it is cycle-free, it is also a tree.

We only have to prove (57) (the sum of the counting numbers is 1). We use Lemma 31:

$$\begin{aligned}
\sum_{R \in \mathcal{R}_{X,i}} c_R &= \sum_{R \in \mathcal{R}_{X,i}} 1 - |\Pi_R| && (65)\\
&= |\mathcal{R}_{X,i}| - \sum_{R \in \mathcal{R}_{X,i}} |\Pi_R| && (66)
\end{aligned}$$

This is equal to the number of nodes in the tree $\mathcal{R}_{X,i}$ minus the number of edges in the tree, and it is an obvious property of trees that this difference is 1.

The second part with the sub-functions can be proven analogously. □

The Kikuchi factorization is defined as follows ([22]).

DEFINITION 33. *The **Kikuchi approximation** of a probability distribution for a region graph is*

$$k(\mathbf{x}) = \prod_{R \in \mathcal{R}} q_R(\mathbf{x}_{s_R})^{c_R} \tag{67}$$

In general, it is not normalized and therefore no probability distribution. The ***normalized Kikuchi approximation***

$$p_k(\mathbf{x}) = \frac{k(\mathbf{x})}{\sum_{\mathbf{y}} k(\mathbf{y})} \tag{68}$$

is a probability distribution.

The computation of $p_k(\mathbf{x})$ is in general exponential.

THEOREM 34. *For a valid region graph without cycles, the Kikuchi approximation is exact.*

PROOF. We prove that the Kikuchi approximation (67) gives exact marginals on the regions:

$$k(\mathbf{x}_{s_R}) = q_R(\mathbf{x}_{s_R}) \tag{69}$$

We choose leaf regions – regions that are connected to only one other region – and eliminate those from the graph, until only the region R remains.

Choose a leaf region $S \neq R$. Since the graph is cycle-free, such a leaf must exist. For the single connection of S, there exist two possibilities:

- S is the child of another region. But since it has only one parent, from Lemma 31 follows that it has counting number $c_S = 0$. So, its local belief $q_S(\mathbf{x}_{s_S})^{c_S}$ has no effect in (67), and we can remove this region.

- S is parent of another region T. Remember that $s_T \subset s_S$. From local consistence of the beliefs we have

$$\sum_{\mathbf{x}_{s_S \setminus s_T}} q(\mathbf{x}_{s_S}) = q(\mathbf{x}_{s_T}) \tag{70}$$

From this, it follows that

$$q(\mathbf{x}_{s_T})^{c_T} \sum_{\mathbf{x}_{s_S \setminus s_T}} q(\mathbf{x}_{s_S}) = q(\mathbf{x}_{s_T})^{c_T+1} \tag{71}$$

We see that it has no effect to eliminate the region S and increase the counting number of T by one, since it has then one parent less.

□

For cycle-free region graphs derived from a junction tree, the Kikuchi approximation is equal to the junction tree distribution. But in general, k is not even a distribution. And the normalized Kikuchi approximation p_k cannot be computed in polynomial time. What we wish for is an approximative distribution which can be computed polynomially and from which points can be sampled.

We now describe a local iteration algorithm, based on the region graph and message passing between regions.

6. THE CONCAVE CONVEX PROCEDURE

The Concave Convex Procedure (CCCP) [27] is a variant of Generalized Belief Propagation GBP proposed in [25]. It is based on the observation that the Lagrangian consists of a convex and a negative convex (concave) term. The *CCCP* algorithm alternates between updates of the convex and the concave term.

6.1 Convex and Concave Lagrangian

We now derive the CCCP update procedure, following [27]. The algorithm is fairly complex. A detailed description can be found in the dissertation [6].

The Lagrangian to be minimized is given by equation (47)

$$\begin{aligned} L = \sum_{R \in \mathcal{R}} c_R \Big(& \sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) \beta E(\mathbf{x}_{s_R}) \\ & + \sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) \log q_R(\mathbf{x}_{s_R}) \Big) \\ & + \sum_{R \in \mathcal{R}} \gamma_R \left(1 - \sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) \right) \\ + \sum_{(P,R) \in E_{\mathcal{R}}} \sum_{\mathbf{x}_{s_R}} & \lambda_{PR}(\mathbf{x}_{s_R}) \left(\sum_{\mathbf{x}_{s_P \setminus s_R}} q_P(\mathbf{x}_{s_P}) - q_R(\mathbf{x}_{s_R}) \right) \end{aligned} \tag{72}$$

The basic idea of CCCP is now to split up L in a convex and a concave part. The problematical part is the entropy term: For regions with $c_R > 0$, the entropy term is convex, for regions with $c_R < 0$ it is concave. The average energy and the constraints are linear in the q_R, so it does not matter where we put them.

To avoid an awkward case separation into convex and concave regions, we set

$$c_{\max} = \max_R c_R \tag{73}$$

and use this definition to split up L into a convex part

$$\begin{aligned} L_{\text{vex}} = \sum_{R \in \mathcal{R}} c_{\max} \Big(& \sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) \beta E_R(\mathbf{x}_{s_R}) \\ & + \sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) \log q_R(\mathbf{x}_{s_R}) \Big) \\ & + \sum_{R \in \mathcal{R}} \gamma_R \left(1 - \sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) \right) \\ + \sum_{(P,R) \in E_{\mathcal{R}}} \sum_{\mathbf{x}_{s_R}} & \lambda_{PR}(\mathbf{x}_{s_R}) \left(\sum_{\mathbf{x}_{s_P \setminus s_R}} q_P(\mathbf{x}_{s_P}) - q_R(\mathbf{x}_{s_R}) \right) \end{aligned} \tag{74}$$

and a concave part

$$\begin{aligned} L_{\text{ave}} = \sum_{R \in \mathcal{R}} (c_R - c_{\max}) \Big(& \sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) E_R(\mathbf{x}_{s_R}) \\ & + \sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) \log q_R(\mathbf{x}_{s_R}) \Big) \end{aligned} \tag{75}$$

It is easy to see that $L = L_{\text{vex}} + L_{\text{ave}}$.

6.2 Outer and Inner Loop

CCCP updates the beliefs and messages in turn. It consists of an *inner loop* in which the messages are updated until convergence, and an *outer loop* in which the current estimates of the beliefs are updated. The inner loop uses the iteration index τ (like GBP), and the outer loop uses the iteration index ξ.

6.2.1 The Outer Loop

For the outer loop iteration the ansatz is

$$\nabla L_{\text{vex}}^{\xi+1} + \nabla L_{\text{ave}}^{\xi} = 0 \tag{76}$$

where ∇L denotes the vector of the partial derivatives of L with respect to the beliefs $q_R(\mathbf{x}_{s_R})$. These derivatives are

$$\frac{\partial L_{\text{vex}}}{\partial q_R(\mathbf{x}_{s_R})} = c_{\max}\left(\beta E_R(\mathbf{x}_{s_R}) + \log q_R(\mathbf{x}_{s_R}) + 1\right) - \gamma_R - \sum_{P|(P,R)\in E_{\mathcal{R}}} \lambda_{PR}(\mathbf{x}_{s_R}) + \sum_{C|(R,C)\in E_{\mathcal{R}}} \lambda_{RC}(\mathbf{x}_{s_C}) \tag{77}$$

and

$$\frac{\partial L_{\text{ave}}}{\partial q_R(\mathbf{x}_{s_R})} = (c_R - c_{\max})\left(\beta E_R(\mathbf{x}_{s_R}) + \log q_R(\mathbf{x}_{s_R}) + 1\right) \ . \tag{78}$$

Inserting (77) and (78) into (76) yields

$$c_{\max}\left(\beta E_R(\mathbf{x}_{s_R}) + \log q_R^{\xi+1}(\mathbf{x}_{s_R}) + 1\right) - \gamma_R - \sum_{P|(P,R)\in E_{\mathcal{R}}} \lambda_{PR}(\mathbf{x}_{s_R}) + \sum_{C|(R,C)\in E_{\mathcal{R}}} \lambda_{RC}(\mathbf{x}_{s_C}) + (c_R - c_{\max})\left(\beta E_R(\mathbf{x}_{s_R}) + \log q_R^{\xi}(\mathbf{x}_{s_R}) + 1\right) = 0 \ . \tag{79}$$

Solving this for $q_R^{\xi+1}(\mathbf{x}_{s_R})$ gives the update equations for the beliefs in the outer loop:

$$q_R^{\xi+1}(\mathbf{x}_{s_R}) = q_R^{\xi}(\mathbf{x}_{s_R})^{\frac{c_{\max}-c_R}{c_{\max}}} \exp\left[-\frac{c_R}{c_{\max}}\beta E_R(\mathbf{x}_{s_R})\right] \exp\left[\frac{\gamma_R - c_R}{c_{\max}} + \frac{1}{c_{\max}}\Big(\sum_{P|(P,R)\in E_{\mathcal{R}}} \lambda_{PR}(\mathbf{x}_{s_R})\Big)\right] \exp -\frac{1}{c_{\max}}\left[\sum_{C|(R,C)\in E_{\mathcal{R}}} \lambda_{RC}(\mathbf{x}_{s_C})\right] \tag{80}$$

For the regions with $c_R = c_{\max}$ the previous belief $q_R^{\xi}(\mathbf{x}_{s_R})$ disappears in this equation.

We next introduce messages ([25])

$$m_{PC}(\mathbf{x}_{s_C}) := e^{\frac{1}{c_{\max}}\lambda_{PC}(\mathbf{x}_{s_C})} \tag{81}$$

and choose γ_R appropriately for normalization, which changes the update equation to

$$q_R^{\xi+1}(\mathbf{x}_{s_R}) \propto q_R^{\xi}(\mathbf{x}_{s_R})^{\frac{c_{\max}-c_R}{c_{\max}}} e^{-\frac{c_R}{c_{\max}}\beta E_R(\mathbf{x}_{s_R})} \frac{\prod_{P|(P,R)\in E_{\mathcal{R}}} m_{PR}(\mathbf{x}_{s_R})}{\prod_{C|(R,C)\in E_{\mathcal{R}}} m_{RC}(\mathbf{x}_{s_C})} \tag{82}$$

6.2.2 The Inner Loop

The inner loop update equation for the messages can be derived by inserting (82) into the consistency equation

$$\sum_{\mathbf{x}_{s_P\setminus s_R}} q_P(\mathbf{x}_{s_P}) = q_R(\mathbf{x}_{s_R}) \tag{83}$$

This gives

$$\sum_{\mathbf{x}_{s_P\setminus s_R}} q_P^{\xi}(\mathbf{x}_{s_P})^{\frac{c_{\max}-c_P}{c_{\max}}} e^{-\frac{c_P}{c_{\max}}\beta E_P(\mathbf{x}_{s_P})} \frac{\prod_{Q|(Q,P)\in E_{\mathcal{R}}} m_{QP}(\mathbf{x}_{s_P})}{\prod_{C|(P,C)\in E_{\mathcal{R}}} m_{PC}(\mathbf{x}_{s_C})} = q_R^{\xi}(\mathbf{x}_{s_R})^{\frac{c_{\max}-c_R}{c_{\max}}} e^{-\frac{c_R}{c_{\max}}\beta E_R(\mathbf{x}_{s_R})} \frac{\prod_{Q|(Q,R)\in E_{\mathcal{R}}} m_{QR}(\mathbf{x}_{s_R})}{\prod_{C|(R,C)\in E_{\mathcal{R}}} m_{RC}(\mathbf{x}_{s_C})} \tag{84}$$

The message $m_{PR}(\mathbf{x}_{s_R})$ is independent of the summation variables $\mathbf{x}_{s_P\setminus s_R}$, so it can be extracted from the sum. It appears in the denominator on the left side of (84) and in the numerator on the right side. This allows to solve the equation for this message.

With the abbreviations

$$g_R(\mathbf{x}_{s_R}) := q_R^{\xi}(\mathbf{x}_{s_R})^{\frac{c_{\max}-c_R}{c_{\max}}} e^{-\frac{c_R}{c_{\max}}\beta E_R(\mathbf{x}_{s_R})} \tag{85}$$

$$h_R(\mathbf{x}_{s_R}) := \frac{\prod_{Q|(Q,R)\in E_{\mathcal{R}}} m_{QR}^{\tau}(\mathbf{x}_{s_R})}{\prod_{C|(R,C)\in E_{\mathcal{R}}} m_{RC}^{\tau}(\mathbf{x}_{s_C})} \tag{86}$$

we arrive at the inner loop update equation

$$m_{PR}^{\tau,\text{upd}}(\mathbf{x}_{s_R}) = m_{PR}^{\tau}(\mathbf{x}_{s_R}) \sqrt{\frac{\sum_{\mathbf{x}_{s_P\setminus s_R}} g_P(\mathbf{x}_{s_P}) h_P(\mathbf{x}_{s_P})}{g_R(\mathbf{x}_{s_R}) h_R(\mathbf{x}_{s_R})}} \tag{87}$$

In order to make the iteration more robust, damping is applied

- Linear damping [26, 27] calculates the messages as a linear combination between the old and update messages:

$$m_{P\to R}^{\tau}(\mathbf{x}_R) = (1-\alpha) m_{P\to R}^{\tau-1}(\mathbf{x}_R) + \alpha m_{P\to R}^{\tau,\text{upd}}(\mathbf{x}_R) \tag{88}$$

In [27], linear damping with $\alpha = 0.1$ was used.

6.3 FDA factorization and region graphs

The marginals proposed by Kikuchi cannot be expressed as an *FDA* factorization. Therefore sampling from the Kikuchi approximation is computationally expensive. In [22] Gibbs sampling has been used. We decided to consider *FDA* factorizations only.

Given an arbitrary *FDA* factorization, we use the specified marginals to create a region graph. This is always possible. Then the Kikuchi approximation is computed using this region graph. After the computation of the approximation the *FDA* factorization is used for sampling.

7. NUMERICAL RESULTS

The EDA family of algorithms seems to be mature, at least for binary problems. It is time to demonstrate the state-of-the-art with large instances of popular benchmark problems. In [16] large graph-bi-partitioning problems have been solved. Large problems have been also solved in [21]. We will continue this work here. We will use Kaufman's $n-k$ function and problems on 2-D grids problems. The number of variables will be up to 900. Kaufman's function is an example of an *ADF* with random connections, the 2-D grid problems are important problems with regular connections.

Size	pro.	entr.	best value	optimum
7(1)	0.184	2.65	34.6649	34.6649
7(2)	0.158	1.26	35.0256	35.0256
7(3)	0.249	1.60	35.4166	35.4166
10(3)	0.077	3.54	72.6994	72.6994
10(1)	0.014	5.26	65.9970	65.9970
15($\beta = 10$)	0.000	5.81	165.4260	165.4260
15($\beta = 30$)	0.000	1.86	165.4260	165.4260
20(1)	0.000	1.96	297.259	297.259
20(2)($\beta = 10$)	0.000	10.88	299.697	300.94
20(2)($\beta = 30$)	0.000	2.05	299.994	300.94
25 ($\beta = 20$)	0.000	5.59	462.743	466.87
25 ($\beta = 30$)	0.000	3.10	463.622	466.87

Table 1: Results of *CCCP* on Ising spin glasses ($\beta = 10$). Best value by *CCCP*, known optimum. $\beta = 10$.

Spin glass:

$$f(\mathbf{x}) = \sum_{i,j} f_{i,j}(s_i, s_j) \tag{89}$$

s_j is one of the 4 neighbors of s_i, $s_i \in \{-1, 1\}$. The function values are randomly drawn uniformly in the interval $[-1, +1]$.

4-grid:

$$f(\mathbf{x}) = \sum_{i,j,k,l} f_{i,j}(x_i, x_j, x_k, x_l) \tag{90}$$

The indices define a plaquette on the 2-D grid. The function values are randomly drawn uniformly in the interval $[-1, +1]$.

Kaufmann random $n - 3$:

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i, x_j, x_k) \tag{91}$$

The indices i, j, k are randomly chosen. The function values are drawn uniformly in $[0, 1]$.

7.1 Numerical results of the Bethe-Kikuchi Approximation

The Bethe-Kikuchi approximation is obtained by minimizing the relative entropy to the Boltzmann distribution. It leads to a convex-concave constraint minimization problem. Thus the minimization problem might have many local minima. This can be confirmed by small instances of the Ising problem, where an exact factorization by the junction tree method is still possible.

The Convex-Concave algorithm *CCCP* is surprisingly fast. Overall, the results are astonishingly good. For small problems (7*7) the Kikuchi approximation can be exact. This can be seen for the problem instances (1) and (3) by looking at the results produced by an exact factorization derived from a junction tree. For 10*10 problems, the Kikuchi approximation is not exact, nevertheless the optimum is generated with a reasonable probability. From problems of size 15*15 on the optimum is generated with a probability less than 10^{-4}. Nevertheless the best values in a sample of size 10000 are nearby the global optimum. For the largest problems (size $25 * 25$) the results become bad.

Altogether, the *CCCP* algorithm is very fast. The quality of the generated solutions is good, but decreases with the size of the problem instance.

problem	size	alg.	sample.	β	best value
Ising	400	*FDA*	30000	-	297.259
Ising	400	*CCCP*	10000	30	297.259
Ising	625	*FDA*	30000	-	466.460
Ising	625	*CCCP*	10000	30	463.622
4-grid	400	*FDA*	10000	-	207.565
4-grid	400	*CCCP*	10000	30	207.565
4-grid	625	*FDA*	30000	-	320.069
4-grid	625	*CCCP*	10000	30	320.132
4-grid	900	*FDA*	30000	-	459.274
4-grid	900	*CCCP*	10000	30	454.237
$n - 3$	400	*FDA*	10000	-	0.7535
$n - 3$	400	*CCCP*	10000	12000	0.7520
$n - 3$	625	*FDA*	30000	-	0.7501
$n - 3$	625	*CCCP*	10000	15000	0.7436

Table 2: Comparison of *FDA* and *CCCP* on large problems.

7.2 A comparison of *FDA* and *CCCP*

In this section we make a first comparison. We consider the 2-D grid problems Ising spin glasses, Random-4, and Kaufman's random $n - k$ function. Random-4 is an *ADF* with sub-functions of four variables $f(x_1, x_2, x_3, x_4)$ defined on contiguous plaquettes of the 2-D grid. We set $k = 3$ for Kaufman's function. The following table is just a proof of concept.

The standard *FDA* algorithm with large population sizes ($N = 30000$) performs very good on all instances. It should be no surprise that the population size has to be large. The factorization of 2-D grid problems uses marginals of size 5, the sub-function merger algorithms creates marginals up to size 8. It needs a large sample size to compute good estimates of these marginals. We remind the reader that for the *CCCP* algorithm the samples are computed only once, after computing the marginals. Note that the values of β has to be extremely large for the Kaufman function.

8. CONCLUSION AND OUTLOOK

The efficient estimation and sampling of distributions is a common problem in several scientific disciplines. Unfortunately each discipline uses a different language to formulate its algorithms. We have identified two principles used for the approximation – minimizing the Kullback-Leibler divergence $KLD(p||q)$ or $KLD(q||p)$. p is the distribution to be estimated and q its approximation. $KLD(p||q)$ is used by the maximum entropy principle and the maximum loglikelihood principle, $KLD(q||p)$ is used by the Bethe-Kikuchi approximation developed in statistical physics.

We have shown that the basic theory is the same for the two algorithms. This theory deals with the decomposition of graphical models and the computation of approximate factorizations. If the unknown distribution allows an exact factorization, then both methods lead to $KLD = 0$, thus they compute the exact distribution.

We have discussed two EDA algorithms in detail. The standard *FDA* algorithm computes a factorization from the graph representing the structure. If the corresponding graphical model does not fulfill the assumptions of the factoriza-

tion theorem the exact distribution is only approximated. Factorizations which cover as much as possible from the interaction graph G_{ADF} are obtained by merging of subfunctions. The marginals are computed from sampling the *FDA* factorization.

The Bethe-Kikuchi approach computes the marginals from a difficult constrained minimization problem. We have proposed an extension of the original approach which uses the *FDA* factorization. The results show that for binary problems the EDA algorithms perform as good or even better than other heuristics for optimization. At this stage our algorithm is not yet optimized from a numerical point of view, nevertheless is is already competitive. In our opinion too many researchers still investigate 1-D problems. Our theory and practice shows that these problems can be solved exactly in polynomial time if the junction tree factorization is used.

The interested reader can download our software from the WWW site http://www.ais.fraunhofer.de/~muehlen/.

9. REFERENCES

[1] S. M. Aji and R. J. McEliece. The generalized distributive law and free energy minimization. In *Proceedings of the 39th Annual Allerton Conference on Communication, Control, and Computing*, pages 672–681, 2001.

[2] R. G. Almond. *Graphical Belief Modelling*. Chapman & Hall, London, 1995.

[3] U. Bertelè and F. Brioschi. *Nonserial Dynamic Programming*. Academic Press, New York, 1972.

[4] T. M. Cover and J. Thomas. *Elements of Information Theory*. Wiley, New York, 1989.

[5] Y. Gao and J. Culberson. Space complexity of estimation of distribution algorithms. *Evolutionary Computation*, 13(1):125–143, 2005.

[6] R. Höns. *Estimation of Distribution Algorithms and Minimum Relative Entropy*. PhD thesis, University of Bonn, 2005.

[7] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev*, 6:620–643, 1957.

[8] E. T. Jaynes. Where do we stand on maximum entropy? In R. D. Levine and M. Tribus, editors, *The Maximum Entropy Formalism*. MIT Press, Cambridge, 1978.

[9] F. V. Jensen and F. Jensen. Optimal junction trees. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 360–366, Seattle, 1994.

[10] M. I. Jordan, editor. *Learning in Graphical Models*. MIT Press, Cambrigde, 1999.

[11] S. L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, 1996.

[12] A. Martelli and U. Montanari. Nonserial dynamic programming: On the optimal strategy of variable elimination for the rectangular lattice. *J. Math. Anal. Appl.*, 40:226–242, 1972.

[13] R. J. McEliece and M. Yildirim. Belief propagation on partially ordered sets. In *Proceedings of the 15th Internatonal Symposium on Mathematical Theory of Networks and Systems (MTNS 2002)*, 2002.

[14] H. Mühlenbein and R. Höns. The estimation of distributions and the minimum relative entropy principle. *Evolutionary Computation*, 13(1):1–27, 2005.

[15] H. Mühlenbein and T. Mahnig. FDA - a scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7(4):353–376, 1999.

[16] H. Mühlenbein and T. Mahnig. Evolutionary optimization and the estimation of search distributions with applications to graph bipartitioning. *Journal of Approximate Reasoning*, 31(3):157–192, 2002.

[17] H. Mühlenbein and T. Mahnig. Mathematical analysis of evolutionary algorithms. In C. C. Ribeiro and P. Hansen, editors, *Essays and Surveys in Metaheuristics*, Operations Research/Computer Science Interface Series, pages 525–556. Kluwer Academic Publisher, Norwell, 2002.

[18] H. Mühlenbein and T. Mahnig. Evolutionary algorithms and the Boltzmann distribution. In K. D. Jong, R. Poli, and J. C. Rowe, editors, *Foundations of Genetic Algorithms 7*, pages 525–556. Morgan Kaufmann Publishers, San Francisco, 2003.

[19] H. Mühlenbein, T. Mahnig, and A. Ochoa. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):213–247, 1999.

[20] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. binary parameters. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Lecture Notes in Computer Science 1141: Parallel Problem Solving from Nature - PPSN IV)*, pages 178–187, Berlin, 1996. Springer-Verlag.

[21] M. Pelikan and D. Goldberg. Hierarchical BOA solves Ising spin glasses and MAXSAT. In *Genetic and Evolutionary Computation Conference 2003*, volume 2724 of *Lecture Notes in Computer Science*, pages 1271–1282. Springer, 2003. Also IlliGAL Report No. 2003001.

[22] R. Santana. Estimation of distribution algorithms with Kikuchi approximations. *Evolutionary Computation*, 13(1):67–97, 2005.

[23] Y. W. Teh and M. Welling. On improving the efficiency of the iterative proportional fitting procedure. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, volume 9, 2003.

[24] Y. Xiang, S. K. M. Wong, and N. Cercone. A 'microscopic' study of minimum entropy search in learning decomposable Markov networks. *Machine Learning*, 26:65–92, 1997.

[25] C. Yanover and Y. Weiss. Finding the M most probable configurations using loopy belief propagation. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[26] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Technical Report 2004-040, Mitsubishi Electric Research Laboratories, May 2004.

[27] A. L. Yuille. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14(7):1691–1722, 2002.