Universität Ulm
Fakultät für Informatik

Consistency in Stochastic Networks

HERMANN VON HASSELN
LAURA MARTIGNON
Universität Ulm

Nr. 92-09

Ulmer Informatik-Berichte

November 1992

Consistency in Stochastic Networks

Hermann von Hasseln and Laura Martignon

University of Ulm
Department of Neural Information Processing
Oberer Eselsberg,W-7900 Ulm(Donau)

Abstract

Stochastic networks are given by graphs, whose vertices (nodes, neurons or spins) can take one out of a finite number of states at any given time. The way each vertex changes its state over time is determined by the edges connecting it with other vertices. The edges represent probabilistic dependencies. Updating is usually performed in a parallel or sequential way.

Stochastic networks are used to model expert sytems; in this context confidence numbers are given as dependencies between vertices and the problem is to see, to what extent they are stochastically consistent. Given a graph and a set (or subset) of confidence numbers, we give a procedure that, starting from these confidence numbers, leads to local characteristics which are consistent with the graph.

KEYWORDS: Stochastic networks, Markov networks, updating, Gibbs distribution, confidence numbers, local characteristics.

0 INTRODUCTION

Stochastic networks have been proposed as exact inference machines in expert systems dealing with uncertain reasoning by Pearl and his school [4, 16, 17]. When modelling a real life situation into a Markov or Bayes network (see [17]) one of the problems is that the confidence numbers describing the dependencies between facts are obtained statistically from experiments and may have little to do with exact probabilistic dependencies. The dilemma of establishing criteria for the consistency of confidence numbers is of epistemological nature. Choosing stochastic networks as models of reality means that we are (explicitly or implicitly) assuming that every phenomenon we are treating can be described in terms of a certain "expected value" associated with it. This means that our systems are governed by probability distributions (and therefore allow no "dissipation" or "contraction") and that consistency of data has to be defined as stochastic coherence or compatibility with those probability distributions. In this paper we propose a natural definition of stochastic consistency of confidence numbers and introduce an algorithm that corrects inconsistent data, while producing coherent stochastic models.
Section 1 is an overview of sequential and parallel updating in Markov networks. We present a modification of the Gibbs sampler introduced by Geman and Geman [5] and give a new, short proof of their theorem on stochastic relaxation. We also prove that parallel and sequential updating are equivalent in Markov networks. Section 2 is devoted to the question of consistency of confidence numbers. Consistency is defined in terms of local characteristics and the Confidence Correcting Algorithm we introduce, is a strongly ergodic inhomogenous Markov chain, whose convergence is guaranteed by theorems on positive matrices. In the case that a given set of confidence numbers is already consistent, the algorithm coincides with the Gibbs sampler. Section 3 is an excursion into another field of application. We propose Markov networks as models of (biological) neural networks and the Confidence Correcting Algorithm as a global learning process, where the task is the adaptation to a new edge structure of the underlying graph.

1 Updating in Markov networks

Let $\Lambda = \{1, \ldots, N\}$ and assume that each $i \in \Lambda$ represents a vertex in a graph G. The graph G is determined by the vertices numbered 1 through N and a set of edges connecting some pairs of vertices. G is **fully interconnected** if each pair of vertices of G is connected by an edge.[1]

Throughout this article, we will assume that G is simply connected (or path connected). This means that for each pair i, j in Λ there is a **path** connecting i with j, where a path is a union of edges $\overline{ik_1}, \overline{k_1k_2}, \ldots, \overline{k_{n-1}j}$.

We assume that each vertex can be in one out of a finite number of states. To simplify matters we will assume that there are only two possible states, namely 0 and 1 (others prefer to call them +1 and −1, or *"on"* and *"off"*). Each arrangement of 0's and 1's on the vertices of the graph will be called a **configuration** and Ω will denote the set of all configurations on G. By x_i we denote the outcome function from Ω in $\{0, 1\}$, mapping configuration ω onto its value ω_i at the vertex i.

Definition 1 *We say that a probability measure*

$$\mu : \mathcal{P}(\Omega) \rightarrow \mathbb{R}^+$$

defines a stochastic network on G if:

1. $\mu(\omega) > 0$ *for every* $\omega \in \Omega$

We say that μ *defines a Markov network with respect to the edges of G if*

2. $Pr(x_i = \omega_i \mid x_j = \omega_j, j \neq i) =$
$Pr(x_i = \omega_i \mid x_j = \omega_j,$ *for all* $j \in \Lambda$ *such that there is an edge connecting* i *with* j*),*
for $x = (x_1, \ldots, x_N)$ *and* $i \in \Lambda$.

As usual, $Pr(A \mid B)$ denotes the conditional probability of A given B. Observe that if μ satisfies Condition 1 only, it has no numerical connection to the graph. On the other hand, if the graph is fully interconnected, Condition 2 is trivially satisfied by any μ satifying Condition 1.

In the theory of Markov chains knowledge on the dependence of a given time

[1] If we make no further specification an edge will always be an undirected edge.

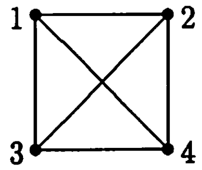

Figure 1: A fully interconnected graph of four vertices

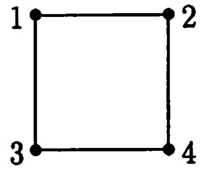

Figure 2: The symmetric cycle of four vertices

instance upon the whole past is equivalent to knowledge on its dependence on the immediate predecessor. This equivalence is called the one-sided Markov property. Clearly (see for instance [8], Prop. 12.4.) every Markov chain with a strictly positive invariant distribution satisfies the two-sided Markov property: knowledge on the dependence of a time instance upon a set of time instances containing it, is equivalent to knowledge on its dependence upon the preceding and following time instances. In stochastic networks which are not fully interconnected (i.e., strict Markov networks) Condition 2 is a generalization of the two-sided Markov property.

1.1 The characterization of Markov Networks

Consider a graph G consisting of four vertices $1, \ldots, 4$. We maintain the assumption that these vertices can take only two possible states, namely 0 and 1. The set of configurations Ω can be viewed as the set $\{0, \ldots, 15\}$, written in binary form. We assume now that G is fully interconnected as in Figure 1. Then, of course, any strictly positive normalized vector $\mu = (\mu_0, \ldots, \mu_{15})$ determines a Markov network on G since Condition 2 of Def. 1 is trivially satisfied.

If we omit the two diagonal edges and consider G as represented by Figure 2 then, for instance:

$$Pr(x_1 = 0 \mid x_2 = 0, x_3 = 0, x_4 = 0) = Pr(x_1 = 0 \mid x_2 = 0, x_3 = 0)$$

and so μ has to satisfy

$$\frac{\mu(0000)}{\mu(0000)+\mu(1000)} = \frac{\mu(0000)+\mu(0001)}{\mu(0000)+\mu(1000)+\mu(0001)+\mu(1001)}$$

A computation shows that μ given by

$$\mu = \\ (0.02721, 0.03161, 0.09483, 0.05537, 0.09499, 0.11035, 0.09033, 0.05274, \\ 0.04296, 0.05741, 0.14970, 0.10054, 0.02342, 0.03131, 0.02228, 0.01496)$$

respects the edge structure of this graph and therefore defines a Markov network. (Remark: These numbers were obtained using the procedure described in Section 2.)

Apparently, it was Dobrushin [3] who first conjectured that probability measures defining Markov networks are given by Gibbs distributions. In a famous and yet unpublished paper of 1968 [9] Hammersley and Clifford proved that Markov networks are characterized as Gibbs distributions determined by neighbor potentials (see for example [8]). In what follows, we present a sketch of this characterization.

The edges of the graph define a system $\mathcal{N}$ of **neighborhoods** in Λ given by $N_i = \{j \mid \text{there is an edge connecting } i \text{ with } j\}$.

These neighborhoods are symmetric in the sense that $i \in N_j$ iff $j \in N_i$. A subset C of Λ is called a **clique** if every pair $\{i,j\}$ contained in C satisfies $i \in N_j$ and $j \in N_i$.

A **potential** U on Λ is a family $\{U_A : A \subset \Lambda\}$ of functions $U_A : \Omega \rightarrow I\!R$ with the property that $U_A(\omega) = U_A(\omega\prime)$ if $\omega_i = \omega\prime_i$ for all $i \in A$. The **energy** H_U of the potential U is given by $H_U = \sum_{A \subset \Lambda} U_A$. The potential U is said to be normalized if $\omega_i = 0$ for some $i \in A$ implies $U_A(\omega) = 0$. If $\mathcal{N}$ is a system of neighborhoods on Λ and $\mathcal{C}$ denotes the set of all cliques in Λ, a potential U is called a **neighbor potential** if $U_A = 0$ whenever $A \notin \mathcal{C}$.

In the terminology of neighborhoods Condition 2 in Definition 1 now reads

$$Pr(x_i = \omega_i \mid x_j = \omega_j, j \neq i) = Pr(x_i = \omega_i \mid x_j = \omega_j, j \in N_i).$$

The number $Pr(x_i = \omega_i \mid x_j = \omega_j, j \neq i)$ is called the **local characteristic** of the vertex $i \in \Lambda$ evaluated at ω and is denoted by $\mu_i^\Lambda(\omega)$. The functions

μ_i^Λ are the local characteristics of the graph. Using these symbols, Condition 2 is expressed by $\mu_i^\Lambda = \mu_i^{N_i}$.

The theorem of Hammersley and Clifford states that if G is a graph on Λ with a neighborhood system $\mathcal{N}$ then μ defines a Markov network on G iff μ is given by

$$\mu(\omega) = \frac{1}{Z} e^{H_U(\omega)} \tag{1}$$

where $Z = \sum_\omega exp(H_U(\omega))$ and H_U is the energy of a neighbor potential U. If one makes the requirement that U is a normalized potential then U is uniquely determined by μ and it is called the **canonical potential** associated to μ. Observe that if the graph is fully interconnected, the edges loose their significance (constraining μ to satisfy Condition 2 of Definition 1) but it still makes sense to speak of energy and potentials.

In accordance with the terminology stemming from Statistical Mechanics a measure given by the expression in (1) as the normalized exponential of an energy is called a **Gibbs measure** or **Gibbs distribution**, H_U is also called a **Hamiltonian** and Z is the **partition function.**

1.2 Updating

Given a Markov network $(\Lambda, \mathcal{N}, \Omega, \mu)$, we begin by describing a stochastic sequential updating on Ω. Our treatment is inspired by the beautiful contribution to the theory of the Boltzmann machines by P.Mazaika in [14]. Starting with any probability distribution on the space Ω we introduce a Markov chain over the sequence of time instances, whose "states" are the configurations of Ω and whose entries are the transition probabilities between pairs of configurations. This is the analogon to the random sequential updating of the Boltzmann machine [1] and corresponds to a modified version of the Gibbs sampler introduced by Geman & Geman in [5]. We begin with a strictly positive probability distribution on Λ, which we denote py p. This probability distribution represents the frequency with which we will visit each vertex of Λ during the updating process.

We are now ready to define the following transition probabilities on the space Ω of all configurations: The probability of going from a configuration ω to another configuration $\omega\prime$ is zero if they differ in more than one component.

If they only differ, say, in the i'th component then

$$\begin{aligned} Pr(\omega \rightarrow \omega') &= p(i)Pr(x_i = \omega_i' \mid x_j = \omega_j' = \omega_j, j \in N_i) \\ &= p(i)\mu_i^{N_i}(\omega') \end{aligned} \tag{2}$$

Finally

$$Pr(\omega \rightarrow \omega) = 1 - \sum_{i=1, i \neq j}^{N} p(i)Pr(x_i = \omega_i' \mid x_j = \omega_j' = \omega_j, j \in N_i) \tag{3}$$

Observe that if ω and ω' differ only in the i'th component, we have

$$Pr(\omega \rightarrow \omega') = p(i)\frac{\mu(\omega_1, \ldots, \overset{i\text{'th}}{\omega_i'}, \ldots, \omega_N)}{\mu(\omega_1, \ldots, \overset{i\text{'th}}{\omega_i'}, \ldots, \omega_N) + \mu(\omega_1, \ldots, \overset{i\text{'th}}{\omega_i}, \ldots, \omega_N)}. \tag{4}$$

The second term in this equality is

$$\begin{aligned} p(i)\frac{e^{H(\omega)}}{e^{H(\omega')} + e^{H(\omega)}} &= p(i)\frac{1}{1 + e^{H(\omega) - H(\omega')}} \\ &= p(i)\frac{1}{1 + e^{\Delta H(\omega, \omega')}}. \end{aligned} \tag{5}$$

The last expression is given by the selection probability $p(i)$ multiplied by the Fermi function calculated in $\Delta H(\omega, \omega')$. Thus, in the case of configurations differing in one component we recognize in eq.(5) the expression for the Glauber dynamics [6]. A word should be said about **detailed balance** in this context. The property of detailed balance is expressed by the equation

$$Pr(\omega \rightarrow \omega')\mu(\omega) = \mu(\omega')Pr(\omega' \rightarrow \omega) \tag{6}$$

which is always satisfied in this network, as an easy computation shows. It is interesting to stress that this is in fact true for any Markov network, regardless of the properties of H. If H is given for example by $H(\omega) = \sum \bar{\omega} J \omega$, where J is a matrix, any possible asymmetry of J does not alter detailed balance, as the energy is always symmetric. This point is discussed further in Section 3.

We denote by $\mathcal{U}$ the updating $2^N \times 2^N$ matrix determined by the entries $(\mathcal{U}_{\omega', \omega}) = Pr(\omega' \rightarrow \omega)$. This transition matrix is positive, stochastic, irreducible and even primitive. It is irreducible because we can go from any ω

to any other $\omega\prime$ in a finite number of time steps. It is primitive because, by construction, we have strictly positive entries on the diagonal [13, 19]. Thus the only eigenvalue of $\mathcal{U}$ on the unit circle is 1. The Jordan block of $\mathcal{U}$ corresponding to 1 is one-dimensional since $\| \mathcal{U} \|_1 = 1$ and the Jordan form of $\mathcal{U}$ is

$$\mathcal{J}(\mathcal{U}) = \begin{pmatrix} 1 & & & & \\ & \mathcal{J}_1 & & & \\ & & \mathcal{J}_2 & & \\ & & & \cdots & \\ & & & & \mathcal{J}_k \end{pmatrix} \tag{7}$$

where $\mathcal{J}_i, i = 1, \ldots, k$, are the Jordan blocks corresponding to the eigenvalues of modulus less then one. The positive powers of $\mathcal{J}(\mathcal{U})$ converge to

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \ldots \\ 0 & 0 & & & \\ 0 & & . & & \\ 0 & & & . & \\ . & & & & . \end{pmatrix} \tag{8}$$

Since $\mathcal{U}$ and $\mathcal{J}(\mathcal{U})$ are similar the positive powers of $\mathcal{U}$ converge to a projection, which is the projection on the one-dimensional space spanned by the strictly positive Perron eigenvector of $\mathcal{U}^T$ (the transpose of $\mathcal{U}$) associated to the eigenvalue 1. On the other hand, our Gibbs distribution is clearly invariant under the action of $\mathcal{U}$ from the left, in symbols $\mu\mathcal{U} = \mu$ or $\mathcal{U}^T\mu = \mu$. Since μ is strictly positive, μ has to coincide with the normalized strictly positive Perron eigenvector of $\mathcal{U}^T$ associated with the eigenvalue 1 [13, 19]. A moment's reflection shows that all we have said is valid for any choice of the strictly positive selection probability on Λ that describes the frequency visiting each vertex. These simple reflections furnish a new transparent proof of the Theorem on stochastic relaxation by Geman and Geman.

Theorem 1 *(Geman & Geman [5]) Let $(\Lambda, \mathcal{N}, \Omega, \mu)$ be a Markov network and $\mathcal{U}$ be the updating matrix determined by any choice of the selection probability p on Λ. The updating process described by action of the positive powers of $\mathcal{U}$ converges to the equilibrium distribution μ.*

A word should be said about parallel updating in Markov networks (see, for instance [2]). The parallel updating transition matrix is unambiguously defined by

$$Pr(\omega \rightarrow \omega\prime) = \prod_{i=1}^{N} Pr(x_i = \omega_i\prime \mid x_j = \omega_j, j \in \mathcal{N}_i) \tag{9}$$

and, of course, requires no selection probability since each transition from one configuration to another happens in one step. We easily deduce the following result:

Theorem 2 *Parallel updating and sequential updating are equivalent in Markov networks.*

PROOF: What we want to prove is that parallel updating also converges towards the invariant equilibrium distribution on the Markov network. To this end we observe that the transition matrix whose entries are given equation (9) is again positive, stochastic, irreducible and primitive. Therefore it has a unique, normalized right invariant vector to which the updating process converges. But since the defining distribution π is obviously right invariant we must have that subsequent application of parrallel updating also converges to π. Thus, the same argument used in the case of sequential updating applies for parallel updating as well.$_{\square}$

2 CONSISTENCY IN MARKOV NETWORKS AND THE CONFIDENCE CORRECTING ALGORITHM

In the plethora of applications of Markov networks one promises to be particularly relevant in the realm of artificial intelligence, namely the modelling of stochastic expert systems. Expert systems consist of data bases and inference machines that use the "knowledge" of the data bases in intelligent way, in order to formulate reasonable answers to the queries posed by the user. Judea Pearl and his school have promoted probabilistic intelligent systems in [4, 16, 17]

Bayes networks have a consolidated tradition as inference machines, whereas Markov networks have only recently ([11, 20]) made their appearence in this context. Both kinds of networks are given by stochastic graphs. The fundamental difference is of semantic nature and is made clear by the use of (undirected) edges in the Markov and of arrows in the Bayes model. In

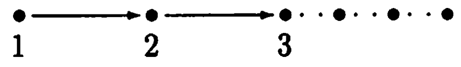

Figure 3: A Markov chain

a Markov network an edge between two vertices (nodes, statements, data) A and B signifies conditional dependence of A upon B and, necessarily vice versa. In a Bayes network an arrow from A to B means B is conditionally dependent upon A and we **deliberately concentrate our attention on this dependence** while neglecting the dependence of A upon B. Thus, the term **causal dependence** becomes appropriate here and the directed "tree" that the set of arrows determine, represents situations ruled by causality, where this causality is defined probabilistically. In both types of networks if A is connected with B and B with C (but C is **not** connected with A) then A is independent of C over B. In Symbols

$$Pr(A \mid B \wedge C) = Pr(A \mid B).$$

Markov chains are usually represented by graphs like that in Figure 3, which are causal graphs, in the sense we have just described. If the invariant distribution is strictly positive a Markov chain is both a Bayes and a Markov network.

Clearly, Bayes networks admit embeddings into Markov networks [11, 20] and therefore allow similar treatments.

In the setting of expert systems one of the main problems is the fact that confidence numbers are given from statistical observations and may have little to do with an underlying graph of conditional dependence. Once we look for the adequate graph to model the data base, we are confronted with the problem of "converting" confidence numbers into local characteristics.

The following definition is natural:

Definition 2 *Let $G = (\Lambda, \mathcal{N}, \Omega)$ be a graph and let $\kappa_i^{N_i}(\omega) := \kappa(x_i = \omega_i \mid x_j = \omega_j, j \in N_i)$ represent the "confidence" we have that $x_i = \omega_i$ if we know that $x_j = \omega_j$ for $j \in N_i$. We say that the confidences $\{\kappa_i^{N_i} \mid i \in \Lambda\}$ are stochastically consistent if they are the are local characteristics of a Markov network on G.*

Assume now that we are given a **complete** set of confidence numbers on a graph $G = (\Lambda, \mathcal{N}, \Omega)$. By complete we mean that $\kappa_i^{N_i}(\omega)$ is known, for every $i \in \Lambda$, N_i neighborhood of i and ω a configuration in Ω. Of course, the least confidence numbers are supposed to satisfy is that the confidence of the union of mutually exclusive and complementary events is equal to one. Consistency with the neighborhood structure is a further requirement, which forces the implicit existence of a probability distribution μ on Ω such that

$$\mu_i^{\Lambda}(\omega) = \mu_i^{N_i}(\omega) = \kappa_i^{N_i}(\omega).$$

As we know, **if such a μ exists, then it is the normalized left invariant vector of $\mathcal{U}$, where $\mathcal{U}$ is the sequential updating matrix induced by the numbers**

$$\mu_i^{\Lambda}(\omega) = \mu_i^{N_i}(\omega) = \kappa_i^{N_i}(\omega), i \in \Lambda, N_i \in \mathcal{N}, \omega \in \Omega$$

(see equations (2) and (3) in Section 1.2) . Thus, an easy test for the consistency of $\{\kappa_i^{N_i}, i \in \Lambda, N_i \in \mathcal{N}\}$ is the following:

1. Construct a sequential updating matrix $\mathcal{S} = \mathcal{S}_0$ by setting

   $\mathcal{S}(\omega, \omega') = 0$, if ω and ω' differ in more than one component,

   $\mathcal{S}(\omega, \omega') = p(i)Pr(x_i = \omega_i' \mid x_j = \omega_j' = \omega_j, j \in N_i) = p(i)\kappa_i^{N_i}(\omega')$

   $\mathcal{S}(\omega, \omega) = 1 - \sum_{i=1, i \neq j}^{N} p(i)Pr(x_i = \omega_i' \mid x_j = \omega_j' = \omega_j, j \in N_i)$.

2. Calculate its left invariant vector $\mu = \mu_0$ and check whether

   $\mu_i^{\Lambda}(\omega) = \mu_i^{N_i}(\omega) = \kappa_i^{N_i}(\omega), i \in \Lambda, N_i \in \mathcal{N}, \omega \in \Omega$

   for all $i \in \Lambda, N_i \in \mathcal{N}, \omega \in \Omega$.

If this equation is satisfied the confidence numbers $\kappa_i^{N_i}(\omega)$ were already consistent. It may happen that $\mu_i^{\Lambda}(\omega) = \mu_i^{N_i}(\omega)$ for all for all $i \in \Lambda, N_i \in \mathcal{N}, \omega \in \Omega$, but $\mu_i^{N_i}(\omega) \neq \kappa_i^{N_i}(\omega)$ for some $i \in \Lambda$. In this case μ defines a Markov network on $G = (\Lambda, \mathcal{N}, \Omega)$. But it may also happen that nothing fits, in other words, that μ is neither a Markov network nor consistent with the given confidence numbers. In the former case, we declare ourselves satisfied and stop our test, since we have found a Markov network. In the latter we will continue our process by defining a new updating matrix $\mathcal{S}_1$ making use of the numbers $\left\{\mu_i^{N_i}(\omega), i \in \Lambda, \omega \in \Omega\right\}$ for its construction. In other words, we will set

$\mathcal{S}_1(\omega,\omega\prime) = 0$, if ω and $\omega\prime$ differ in more than one component,

$\mathcal{S}_1(\omega,\omega\prime) = p(i)Pr(x_i = \omega_i\prime \mid x_j = \omega_j\prime = \omega_j, j \in N_i) = p(i)\mu_i^{N_i}(\omega\prime)$

$\mathcal{S}_1(\omega,\omega) = 1 - \sum_{i=1,i\neq j}^{N} p(i)Pr(x_i = \omega_i\prime \mid x_j = \omega_j\prime = \omega_j, j \in N_i)$.

This new updating matrix $\mathcal{S}_1$ will not have μ as left invariant vector. We will call μ_1 the vector $\mathcal{S}_1^T\mu = \mu\mathcal{S}_1$ and proceed in an analogous way, by constructing an updating matrix $\mathcal{S}_2$ making use of μ_1 to define its entries. Going on like this we obtain an algorithm and the important aspect of this algorithm is that it converges ! We have a sequence $(\mathcal{S})_{n\in\mathbb{N}}$ of stochastic, irreducible, primitive and positive matrices, and a sequence $(\mu_n)_{n\in\mathbb{N}}$ of probability vectors on Ω such that $(\mathcal{S}_n)_{n\in\mathbb{N}}$ is defined as the sequential updating matrix determined by $\mathcal{S}_{n-1}^T\mu_{n-1}$ and we can prove:

Theorem 3 *Let $(\mathcal{S}_n)_{n\in\mathbb{N}}$ and $(\mu_n)_{n\in\mathbb{N}}$ be the sequence defined in the preceding discussion, where $\mathcal{S}_0$ is the sequential updating matrix determined by a given set of confidence numbers $\left\{\kappa_i^{N_i}(\omega), i \in \Lambda, \omega \in \Omega\right\}$. $\mathcal{S}_1$ is the updating matrix determined by μ_0, the normalized left invariant vector of $\mathcal{S}_0$, $\mathcal{S}_2$ the matrix determined by $\mu_1 = \mathcal{S}_1^T\mu_0$ and so on.*
The sequence

$$(\mu_n)_{n\in\mathbb{N}} = (\mu_0, \mu_1 = \mathcal{S}_1^T\mu_0, \mu_2 = \mathcal{S}_2^T\mathcal{S}_1^T\mu_0, \ldots, \mu_n = \mathcal{S}_n^T\cdots\mathcal{S}_1^T\mu_0, \ldots) \quad (10)$$

converges in any norm of $\mathbb{R}^{2^N}$ towards a Markov network on $(\Lambda, \mathcal{N}, \Omega)$.

PROOF: We look at the sequence $(\nu_n))_{n\in\mathbb{N}}$ defined by

$$\nu_0 = \mu_0, \nu_1 = \mathcal{S}_1\mu_0, \nu_2 = \mathcal{S}_2\mathcal{S}_1\mu_0, \ldots, \nu_n = \mathcal{S}_n\cdots\mathcal{S}_1\mu_0, \ldots .$$

Observe that all matrices $(\mathcal{S}_n)_{n \in \mathbb{N}}$ are row stochastic. We want to prove that there exists a $\beta < 1$ such that

$$max_{i,k} \mid (\nu_n)_i - (\nu_n)_k \mid \leq \beta max_{i,k} \mid (\nu_{n-1})_i - (\nu_{n-1})_k \mid \tag{11}$$

for n large enough. Since the diagonal elements of each $(\mathcal{S}_n)_{n \in \mathbb{N}}$ are strictly positive, the product $\mathcal{S}_n \cdots \mathcal{S}_1$ is a strictly positive, stochastic matrix, the same being true about $\mathcal{S}_{n+m} \cdots \mathcal{S}_1$, for all $m \in \mathbb{N}$. The contracting property of stochastic matrices [2] ([19], Theorem 3.1) implies

$$max_{i,k} \mid (\nu_{n+m})_i - (\nu_{n+m})_k \mid \leq \tau(\mathcal{S}_{n+m} \cdots \mathcal{S}_1) max_{i,k} \mid (\mu_0)_i - (\mu_0)_k \mid, \tag{12}$$

where the contraction coefficient $\tau(.)$ is defined as follows: for any stochastic matrix $\mathcal{S}$

$$\tau(\mathcal{S}) = 1 - min_{i,j} \sum_{s=1}^{2^N} min\left((\mathcal{S})_{is}, (\mathcal{S})_{js}\right). \tag{13}$$

It is possible to prove that $\tau(\mathcal{S}_{n+m} \cdots \mathcal{S}_1), m \in \mathbb{N}$, can be estimated in terms of the original vector μ_0. In fact

$$\tau(\mathcal{S}_{n+m} \cdots \mathcal{S}_1) < \frac{1}{1 + \frac{max_k \left(\sum_{i=1}^{2^n} (\mu_0)_i - (\mu_0)_k\right)}{min_k (\mu_0)_k}} < 1. \tag{14}$$

Thus, $(\nu_n))_{n \in \mathbb{N}}$ converges towards a constant (not necessarily normalized) vector μ (in any of the equivalent norms of $\mathbb{R}^{2^N}$). A similar argument shows that also the sequence

$$(\mu_0, \mathcal{S}_1\mu_0, \mathcal{S}_1\mathcal{S}_2\mu_0, \ldots, \mathcal{S}_1 \cdots \mathcal{S}_n\mu_0, \ldots)$$

is convergent. This, in turn, means that the adjoint vectors

$$(\mu_0^T, \mu_0^T\mathcal{S}_1, \mu_0^T\mathcal{S}_1\mathcal{S}_2, \ldots, \mu_0^T\mathcal{S}_1 \cdots \mathcal{S}_n, \ldots)$$

also form a convergent sequence, and this is what we wanted to prove. The limit distribution μ is again strictly positive and its associated matrix (the limit of the products $\mathcal{S}_0, \mathcal{S}_0\mathcal{S}_1, \ldots, \mathcal{S}_0 \cdots \mathcal{S}_n, \ldots$) is again primitive, stochastic

[2] This theorem states that, given an arbitrary vector w=(w_i) and a stochastic n × n matrix P=(P_{ij}) and z=Pw, we have $max_j(z_j) - min_j(z_j) \leq \tau(P)(max_j(w_j) - min_j(w_j))$, where $\tau(P) = 1 - min_{i,j} \sum_{s=1}^{n} min(p_{is}, p_{js})$.

and irreducible $\square$

The sequence of products $\mathcal{S}_0, \mathcal{S}_0\mathcal{S}_1, \ldots, \mathcal{S}_0 \cdots \mathcal{S}_n, \ldots$ of matrices is a (convergent) strongly ergodic inhomogeneous Markov chain satisfying the so called Condition (C) (see [19], Theorem 4.12), which is a requirement on the "boundedness away from zero" of the nonnegative elements in the matrices involved. More precisely, Condition (C) requires the existence of a number $\gamma > 0$ such that every nonnegative entry of the matrices in the products is bigger than γ.
It is easy to see that we could have used parallel updating matrices as well. We could have started with a matrix $\mathcal{P}_0$ determined by the confidence numbers given by equation (9) in Section 1.2. Again, by performing an analogous iteration process, we could obtain a sequence $(\mathcal{P}_n)_{n\in N}$ and a sequence (π_n) of normalized vectors, such that $\pi_n = \mathcal{P}_{n-1}\pi_{n-1}$. Since all matrices $\mathcal{P}_n$ are strictly positive by construction and the same estimate for the ergodic coefficient $\tau(\mathcal{P}_n)$ is valid, we would obtain the convergence of (π_n). In fact, we have

$$\lim_{n\to\infty} \pi_n = \lim_{n\to\infty} \mu_n. \tag{15}$$

One characterization of the limiting distribution μ (beside being a Gibbs distribution) is, that it minimizes the socalled "relative entropy" or "cross entropy", first proposed by Kullback [12]. The principle of minimum relative entropy is a generalization of the principle of maximum entropy and applies in cases where a prior probability distribution (or, stated otherwise, a set of confidence numbers) is known. As is well known, the methods of maximum entropy and minimum relative entropy are correct methods of inference in estimating a probability distribution. In our case we are given a set or a subset of confidence numbers corresponding to a starting distribution. This starting distribution can be viewed as an estimate of the consistent one, and applying the Confidence Correcting Algorithm (CCA) corresponds to finding the consistent probability distribution, which "automatically" minimizes the relative entropy. Stated otherwise, the CCA is a correct way for finding the probability distribution, which "is uniquely determined as the one which is maximally noncommittal with regard to missing information" as Jaynes [10] states. A very similar situation, where an inhomogenous Markov chain of stochastic matrices minimizes relative entropy is described in [7], Theorem 1.

3 Connectivity Matrices and Neural Networks

A suggestive example of Markov networks is furnished by graphs whose edge structure is determined by connectivity matrices. Such graphs can be used to describe the activity determined by synaptic interconnections in a biological neural network (see for example [15]).

Definition 3 *A matrix* $J = (J_{ij})$, $1 \leq i, j \leq N$ *with real entries is called a connectivity matrix if* $(J_{ii}) = 0$ *for each* $1 \leq i \leq N$

A connectivity matrix determines an edge structure on vertices numbered 1 through N if one establishes that an edge connects vertex i with vertex j iff either $J_{ij} \neq 0$ or $J_{ji} \neq 0$.
We consider the function $H : \Omega \rightarrow I\!R$ defined by

$$H(\omega) = \tilde{\omega} J \omega, \tag{16}$$

where we identify ω with the column vector and $\tilde{\omega}$ with the row vector. This function is the energy determined by the potential $U = \{U_A : A \subset \{1, \ldots, N\}\}$, where

$$U_A = 0$$

if A is not a pair of vertices, and

$$U_{\{i,j\}}(\omega) = \omega_i J_{ij} \omega_j + \omega_j J_{ji} \omega_i = \omega_i \omega_j (J_{ij} + J_{ji}).$$

Clearly $U_{\{i,j\}} = 0$ iff $J_{ij} = 0$ and $J_{ji} = 0$.
The theorem of Hammersley and Clifford implies the following result.

Proposition 1 *Let* J *be a connectivity matrix on* Λ.
The function $\mu(\omega) = \frac{1}{Z} e^{H(\omega)}$, *where* $H(\omega)$ *is defined in equation (16), defines a Gibbs distribution on* Ω *and a Markov network on the graph determined by* J *on the set* Λ *of vertices.*

This proposition is a corollary of the characterization of Markov networks (see [8], Th. 12-16). $\square$

Indeed, Markov networks can be used to model the activity of neural networks, where the details of the biological neurons are compressed in the stochastic description of the dynamics taking place in the configuration space. This approach is, of course, different from the one of the "classical" models (see the Little-Shaw model or the stochastic version of the Hopfield model, described in [18]). If π is given by a connectivity matrix, i.e., $\pi(\omega) = \frac{1}{Z}e^{H}(\omega)$,where $H(\omega) = \bar{\omega}J\omega$, then removing a pair of terms J_{ij} and J_{ji} and replacing them by zero correspond to eliminating the $\overline{ij}$ edge from the graph. If $\tilde{J}$ represents the matrix we obtain through this substitution, then $\tilde{\pi} = \frac{1}{Z}e^{\bar{\omega}\tilde{J}\omega}$ is consistent with the new graph. Yet, as computations easily show, the distribution $\tilde{\pi} = \frac{1}{Z}e^{\bar{\omega}\tilde{J}\omega}$ is far more distant from $\pi(\omega) = \frac{1}{Z}e^{H}(\omega)$, than the distribution we obtain by performing the CCA starting with π on the network determined by $\tilde{J}$. Thus we feel free to say that the CCA is a global learning algorithm by which the distribution $\pi(\omega) = \frac{1}{Z}e^{H}(\omega)$ gradually adapts to the edge structure defined by $\tilde{J}$. The problem is now to associate this global learning with local learning rules.

4 Simulation Results And Conclusions

In our computer simulations we modeled stochastic networks with a maximum number of 8 vertices where we used an IBM-PC with an Intel 486DX processor.

For a graph with N vertices, every row vector of the $2^N \times 2^N$ transition matrix $\mathcal{U}$ contains only $N+1$ non-zero elements, namely N strict transitions and one diagonal element. There are $\frac{1}{2}N(N-1)$ elements to be computed since for each $\mathcal{U}_{\omega\prime,\omega}$ we have $\mathcal{U}_{\omega\prime,\omega} = p(\imath)(1 - \frac{1}{p(\imath)}\mathcal{U}_{\omega,\omega\prime})$, where ω and $\omega\prime$ denote two arbitrary configurations and $p(\imath)$ is the selection probability of the i'th vertex. Therefore these transition matrices are sparse and the computation time grows polynomially in N.

The program we used produced the transition matrix for a given graph according to eqs. (2) and (3) in Section 1.2. The inhomogenous Markov chain described in Theorem 3 started with a randomly chosen probability distribution. In every iteration step the program computes the "new" confidence numbers from the resulting distribution and compares "old" and "new" ones. If the new confidence numbers are equal to the old ones the program stops and these last confidence numbers are the local characteristics for the graph. Our program was not optimized for computational velocity. This has to be done in order to perform simulations for larger graphs.

In the appendix we list some simulation results for different graphs. An interesting feature in the context of expert systems is the possibility of fixing a subset of confidence numbers (or local characteristics: one may have certainty about those outcomes) and correcting the remaining ones. This is shown in Example 2 of the appendix. It is a difficult combinatorial and open problem to deduce how many confidence numbers can be kept fixed for a given graph.

In all three examples one can see that the values of the resulting local characteristics and invariant distributions are only slightly modified by the algorithm and we conjecture that the resulting distributions are those which are closest to the starting distributions in the simplex of all probability vectors.

ACKNOWLEDGEMENT

Prof. G. Palm made us familiar with the mathematical aspects of stochastic relaxation. We thank him for suggestions and feedback through illuminating discussions.

This work was partially supported by the Bundesministerium für Forschung und Technik.

APPENDIX

Here we list some simulation results for various graphs starting with random confidence numbers. The first example is the graph of Figure 1 in Section 1. In the first table for each example we give the confidence numbers and the "corrected" ones, the local characteristics. The second tables shows the values of the starting distribution (corresponding to the confidence numbers) and the invariant and consistent distribution for each configuration.

The second example shows the simulations for the same graph, but this time some confidence numbers were kept fixed. These are marked as "fixed" in Table 3. Note that this time the number of iterations is much larger than in the example before.

The third and last example is the simulation of a graph like in Figure 4.

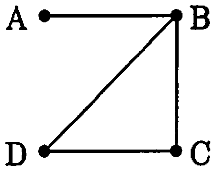

Figure 4: The graph of example 3

Note that here only the confidence numbers of vertex B are changed. The other confidence numbers corresponding to vertices A, C and D were not kept fixed, yet stay unaltered. In all three examples we started with the same probability distribution.

conditional probability	confidence numbers	local characeristics
Pr(A=1 | B=0,D=0) =	0.541806	0.613624
Pr(A=1 | B=0,D=1) =	0.737659	0.622097
Pr(A=1 | B=1,D=0) =	0.267344	0.19827
Pr(A=1 | B=1,D=1) =	0.144778	0.204036
Pr(B=1 | A=0,C=0) =	0.851462	0.78467
Pr(B=1 | A=0,C=1) =	0.420699	0.48478
Pr(B=1 | A=1,C=0) =	0.229497	0.362017
Pr(B=1 | A=1,C=1) =	0.197129	0.127793
Pr(C=1 | B=0,D=0) =	0.797437	0.789136
Pr(C=1 | B=0,D=1) =	0.608243	0.638913
Pr(C=1 | B=1,D=0) =	0.441728	0.491434
Pr(C=1 | B=1,D=1) =	0.374046	0.3136
Pr(D=1 | A=0,C=0) =	0.586756	0.558195
Pr(D=1 | A=0,C=1) =	0.346096	0.373967
Pr(D=1 | A=1,C=0) =	0.513583	0.567026
Pr(D=1 | A=1,C=1) =	0.411147	0.382406

Table 1: Confidence numbers and local characteristics of example 1

configuration	starting distribution	invariant distribution
0000	0.022249	0.0257088
0001	0.0182591	0.0324815
0010	0.133671	0.0962122
0011	0.0376964	0.0574734
0100	0.0904472	0.0936839
0101	0.141756	0.118364
0110	0.059765	0.0905278
0111	0.0646852	0.0540777
1000	0.046682	0.0408294
1001	0.0653	0.0534704
1010	0.137691	0.1528
1011	0.0920376	0.0946114
1100	0.0240121	0.0231683
1101	0.0093422	0.0303413
1110	0.0307998	0.0223878
1111	0.0256055	0.0138622
number of iterations:	100	

Table 2: Starting and invariant distributions of example 1

conditional probability	confidence numbers	local characeristics
Pr(A=1 | B=0,D=0) =	0.541806 (fixed)	0.541806
Pr(A=1 | B=0,D=1) =	0.737659	0.634584
Pr(A=1 | B=1,D=0) =	0.267344	0.136895
Pr(A=1 | B=1,D=1) =	0.144778	0.188926
Pr(B=1 | A=0,C=0) =	0.851462 (fixed)	0.851462
Pr(B=1 | A=0,C=1) =	0.420699	0.531278
Pr(B=1 | A=1,C=0) =	0.229497	0.43467
Pr(B=1 | A=1,C=1) =	0.197129	0.131969
Pr(C=1 | B=0,D=0) =	0.797437 (fixed)	0.797437
Pr(C=1 | B=0,D=1) =	0.608243 (fixed)	0.608243
Pr(C=1 | B=1,D=0) =	0.441728	0.437704
Pr(C=1 | B=1,D=1) =	0.374046	0.23489
Pr(D=1 | A=0,C=0) =	0.586756 (fixed)	0.586756
Pr(D=1 | A=0,C=1) =	0.346096	0.358968
Pr(D=1 | A=1,C=0) =	0.513583	0.675877
Pr(D=1 | A=1,C=1) =	0.411147	0.451273

Table 3: Confidence numbers and local characteristics of example 2

```
dos2unix %1 e:temp.txt
rsh neuro print temp.txt
```

configuration	starting distribution	invariant distribution
0000	0.022249	0.0219203
0001	0.0182591	0.0311241
0010	0.133671	0.086294
0011	0.0376964	0.0483234
0100	0.0904472	0.125653
0101	0.141756	0.178412
0110	0.059765	0.0978109
0111	0.0646852	0.0547726
1000	0.046682	0.0259203
1001	0.0653	0.0540503
1010	0.137691	0.102041
1011	0.0920376	0.0839186
1100	0.0240121	0.0199296
1101	0.0093422	0.0415581
1110	0.0307998	0.0155136
1111	0.0256055	0.0127584
number of iterations:	1238	

Table 4: Starting and invariant distributions of example 2

conditional probability	confidence numbers	local characeristics
Pr(A=1 | B=0) =	0.617267	0.617267 (unchanged)
Pr(A=1 | B=1) =	0.201068	0.201068 (unchanged)
Pr(B=1 | A=0,C=0,D=0) =	0.802575	0.776095
Pr(B=1 | A=0,C=0,D=1) =	0.885891	0.790562
Pr(B=1 | A=0,C=1,D=0) =	0.308965	0.410608
Pr(B=1 | A=0,C=1,D=1) =	0.631805	0.592301
Pr(B=1 | A=1,C=0,D=0) =	0.339662	0.351023
Pr(B=1 | A=1,C=0,D=1) =	0.12516	0.370685
Pr(B=1 | A=1,C=1,D=0) =	0.182798	0.0980531
Pr(B=1 | A=1,C=1,D=1) =	0.217654	0.184808
Pr(C=1 | B=0,D=0) =	0.797437	0.797437 (unchanged)
Pr(C=1 | B=0,D=1) =	0.608243	0.608243 (unchanged)
Pr(C=1 | B=1,D=0) =	0.441728	0.441728 (unchanged)
Pr(C=1 | B=1,D=1) =	0.374046	0.374046 (unchanged)
Pr(D=1 | B=0,C=0) =	0.547964	0.547964 (unchanged)
Pr(D=1 | B=0,C=1) =	0.323448	0.323448 (unchanged)
Pr(D=1 | B=1,C=0) =	0.568985	0.568985 (unchanged)
Pr(D=1 | B=1,C=1) =	0.499242	0.499242 (unchanged)

Table 5: Confidence numbers and local characteristics of example 3

configuration	starting distribution	invariant distribution
0000	0.022249	0.0263822
0001	0.0182591	0.0319808
0010	0.133671	0.103859
0011	0.0376964	0.0496535
0100	0.0904472	0.0914452
0101	0.141756	0.120717
0110	0.059765	0.0723551
0111	0.0646852	0.0721361
1000	0.046682	0.0425489
1001	0.0653	0.051783
1010	0.137691	0.167503
1011	0.0920376	0.0800805
1100	0.0240121	0.0230142
1101	0.0093422	0.0303811
1110	0.0307998	0.0182097
1111	0.0256055	0.0181546
number of iterations:	51	

Table 6: Starting and invariant distributions of example 3

References

[1] Ackley, D.H.; Hinton, G.E.; Sejnowski, T.J.: "A learning algorithm for Boltzmann machines", **Cogn. Sci. 9 (1985) 147-169.**

[2] Clark, J.W.: "Statistical Mechanics of Neural Networks", **Phys. Rep. 158 (1988) 91-157.**

[3] Dobrushin, R.L.: "The Gibbsian random fields for lattice systems with pairwise interaction", **Func. Anal. Appl. 2 (1968) 292-301.**

[4] Geffner, H.; Pearl, J.: "On the probabilistic Semantics of Connectionist Networks", **Cognitive Systems Laboratory, UCLA Computer Science Department, L.A. CA 90024 II-187,II-195 (1989).**

[5] Geman, S.;Geman, D.: "Stochastic Relaxation, Gibbs distributions, and the Bayesian restoration of images", **IEEE Trans. Pattern Anal. Machine Intell. 6 (1984) 721-741.**

[6] Glauber, R.J.: "Time-Dependent Statistics of the Ising Model", **J. Math. Phys. 4 (1963) 294-307.**

[7] Goutsias, J.K.: "A Theoretical Analysis of Monte Carlo Algorithms for the Simulation of Gibbs Random Field Images", **IEEE Trans. Inf. Theor. 37(1991)1618-1628**

[8] Griffeath, D.: "Introduction to random fields", in: Denumerable Markov Chains by Kemeny,Knapp and Snell, 2nd edition,Springer-Verlag, 1976.

[9] Hammersley, J.M.; Clifford, P.: "Markov fields on finite graphs and lattices", **Preprint Univ. of CAL. Berkeley (1968).**

[10] Jaynes, E.T.: "Information Theory and statistical mechanics 1", **Phys.Rev. 106(1957)620-630**

[11] Hrycej, T.: "Gibbs sampling in Bayesian Networks", **Art. Intell. 46 (1990) 351-363.**

[12] Kullback, S.: "Information Theory and Statistics", New York: Wiley,1959

[13] Martignon, L.;Nagel, R.: "The Powers of Positive Matrices", to appear.

[14] Mazaika, P.K.: "A Mathematical Model of the Boltzmann Machine" in IEEE First International Confrence on Neural Networks (San Diego 1987), eds. M.Caudill and C.Butler, vol.3,157-163. New York: IEEE

[15] Palm,G.: "Neural Assemblies: An alternative approach to artificial intelligence", Springer-Verlag 1982

[16] Pearl, J.: "Evidential Reasoning Using Stochastic Simulation of Causal Models", **Art. Intell. J. 32 (1986) 245-257.**

[17] Pearl, J.: Probabilistic Reasoning in Intelligent systems, Kaufman San Mateo 1989.

[18] Peretto, P.: "Collective Properties of Neural Networks: A Statistical Physics Approach", **Biol. Cybern. 50 (1984) 51-62.**

[19] Seneta, E.: "Non-Negative Matrices and Markov Chain", Springer Series in Statistics, 2nd edition, Springer-Verlag 1981.

[20] Spiegelhalter, D.J.: "Probabilistic reasoning in predictive expert systems", in: Kanal, L.N.;Lemmer, J.F. (eds.) "Uncertainty in Artificial Intelligence", North-Holland, Amsterdam, 1986. 47-67.

Liste der bisher erschienenen Ulmer Informatik-Berichte:
List of technical reports currently available from the University of Ulm:

91-01 Ker-I Ko, P. Orponen, U. Schöning, O. Watanabe:
Instance Complexity.

91-02 K. Gladitz, H. Fassbender, H. Vogler:
Compiler-Based Implementation of Syntax-Directed Functional Programming.

91-03 Alfons Geser:
Relative Termination.

91-04 Johannes Köbler, Uwe Schöning, Jacobo Toran:
Graph Isomorphism is low for PP.

91-05 Johannes Köbler, Thomas Thierauf:
Complexity Restricted Advice Functions.

91-06 Uwe Schöning:
Recent Highlights in Structural Complexity Theory.

91-07 Frederic Green, Johannes Köbler, Jacobo Toran:
The Power of the Middle Bit.

91-08 V. Arvind, Y. Han, L. Hemachandra, J. Köbler, A. Lozano,
M. Mundhenk, M. Ogiwara, U. Schöning, R. Silvestri, T. Thierauf:
Reductions to Sets of Low Information Content.

92-01 Vikraman Arvind, Johannes Köbler, Martin Mundhenk:
Bounded Truth-Table and Conjunctive Reductions to Sparse and Tally Sets.

92-02 Thomas Noll, Heiko Vogler:
Top-down Parsing with Simultaneous Evaluation of Noncircular Attribute Grammars

92-03 Program and Abstracts:
17. Workshop über Komplexitätstheorie, effiziente Algorithmen und Datenstrukturen
am 26. Mai 1992 in Ulm

92-04 V. Arvind, J. Köbler, M. Mundhenk
Lowness and the Complexity of Sparse and Tally Descriptions

92-05 Johannes Köbler
Locating P/poly Optimally in the Extended Low Hierarchy

92-06 Armin Kühnemann, Heiko Vogler
Synthesized and inherited functions - a new computational model
for syntax-directed semantics

92-07 Heinz Fassbender, Heiko Vogler
A Universal Unification Algorithm Based on Unification-Driven
Leftmost Outermost Narrowing.

92-08 Uwe Schöning
On Random Reductions from Sparse Sets to Tally Sets

92-09 Hermann von Hasseln, Laura Martignon
Consistency in Stochastic Networks

Ulmer Informatik-Berichte

ISSN 0939-5091

Herausgeber: Fakultät für Informatik

Universität Ulm, Oberer Eselsberg, W-7900 Ulm