Monte Carlo Inference for Belief Networks Using Coupling From the Past

by

Michael Harvey

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

Monte Carlo Inference for Belief Networks Using Coupling From the Past

Michael Harvey

Master of Science

Graduate Department of Computer Science

University of Toronto

1999

A common method of inference for belief networks is Gibbs sampling, in which a Markov chain converging to the desired distribution is simulated. In practice, however, the distribution obtained with Gibbs sampling differs from the desired distribution by an unknown error, since the simulation time is finite. Coupling from the past selects states from exactly the desired distribution by starting chains in every state at a time far enough back in the past that they reach the same state at time $t = 0$. To track every chain is an intractable procedure for large state spaces. The method proposed in this thesis uses a summary chain to approximate the set of chains. Transitions of the summary chain are efficient for noisy-or belief networks, provided that sibling variables of the network are not directly connected, but often require more simulation time steps than would be needed if chains were tracked exactly. Testing shows that the method is a potential alternative to ordinary Gibbs sampling, especially for networks that have poor Gibbs sampling convergence, and when the user has a low error tolerance.

Acknowledgements

Thank you to Radford Neal for a substantial amount of guidance in producing this thesis, to Mick Molloy for checking and suggestions, and to my wife Marissa and children for a great deal of support and patience.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Belief networks are used in expert systems as a means of dealing with uncertainty. Traditional rule based systems cannot successfully do this. Attempts were made to incorporate classical probability theory into expert systems (Gorry and Barnett 1968; Gorry 1973) without success because of intractable calculations. Pearl (1986) arrived at a more feasible solution by mapping the variables of the problem into a belief network in which edges are used to specify probability relationships. This greatly reduced the calculation load by allowing much of it to be localized within the network.

Still, calculation of exact probabilities remains limited to problems with a small number of variables, or networks that can be represented as tree structures. Junction trees are used as a way of converting general belief networks into tree structures that can be evaluated by propagating update messages through the tree (Jensen 1996). If each node of the tree contains only a small number of variables, evaluations of the tree remain feasible even as the tree grows. However, in general, belief network inference by junction trees is an intractable procedure, since calculations take time that grows exponentially with the number of variables in a node.

A general solution for inference in large networks is approximation by Gibbs sampling. It is subject to difficulties due both to error and to the uncertainty about the magnitude of this error. The error can be measured by the total variation distance between the desired distribution and the distribution of the simulation samples. Gibbs sampling, started from an arbitrary initial state, converges asymptotically from an initial distribution to the desired distribution. The error is greatest in the first samples collected. These are thrown away in a burn-in phase until it is felt that the error has dropped to within the desired error tolerance. The burn-in time must usually be estimated because the rate of convergence of the Markov chain is not known, theoretically. It is possible to overestimate and waste computing time by throwing away good samples, or underestimate and include samples that are too far from the desired distribution. The conservative user might greatly overestimate the required burn-in time, just to minimize the possibility of getting the wrong answer.

To overcome the problem of initialization bias, Propp and Wilson (1997) introduced exact sampling, also known as perfect simulation, using the method of coupling from the past to select initial states from the invariant distribution. Instead of starting a single chain at some arbitrary initial state at time $t = 0$, chains for every possible state are started at some time $t < 0$, where t is far enough back to ensure that all the chains coalesce to a single state by time $t = 0$. The state at $t = 0$ then comes from the correct distribution, and useful sampling can begin at that time with zero bias, or systematic error.

Propp and Wilson demonstrated coupling from the past on monotonic problems. Monotonic problems have the property that the state space can be partially ordered with unique maximal and minimal states in a way that is preserved through transitions. Two chains for the maximal and minimal states are all that need to be simulated, since when they coalesce, all of the other chains have coalesced as well. They show that the computational effort for coupling from the past on monotonic problems is comparable or even less than a practical burn-in time, since the rate of coalescence of the chains determines a lower bound on the time for burn-in to achieve a particular error tolerance.

In general, belief networks do not have the monotonicity property, so some other way of keeping track of all possible chains is needed. It is possible to represent a set of chains by a single chain on a state space of sets of states. The set of chains may contain both those that are of interest and spurious chains added at intermediate stages of the simulation. When the single chain on the state set space is summarizing only one chain, then coalescence of the true chains, as well as the spurious chains has occurred. The chain will then exactly represent the one chain that it summarizes, since it is the result of coalescence of the true chains.

This thesis addresses exact sampling for noisy-or belief networks, with a provision that sibling variables are not directly connected. For these networks, the summary chain transitions can be performed efficiently. The amount of work is related to the average coalescence time of the chains, plus a factor for the overhead of coupling the spurious chains.

When is it advantageous to incur the computational expense of exact sampling to gain the benefit of a guaranteed zero initialization bias? Problems of particular interest are those that require a lot of effort in Gibbs sampling burn-in. These problems have poor convergence properties; they take a lot of time to reach a given error tolerance. Furthermore, the convergence characteristics of Markov chain problems are in general not known. Therefore, a cautious user tends to further exacerbate the cost factor by greatly overestimating the needed burn-in time.

Considering this type of scenario, a hypothetical burn-in time may be estimated, and compared to the computational cost of experimental exact sampling runs. The burn-in time estimate can actually come from the average experimental coalescence times, since it can be used to arrive at a burn-in time guaranteed to satisfy the error tolerance. Then the burn-in time estimate is inflated to reflect the uncertainty of the user about the error.

The cost of exact sampling is sometimes more than the time needed for burn-in to achieve a given error tolerance. However, a user can still be better off using exact sampling since it removes that uncertainty which causes the burn-in times to be exaggerated. For problems that have better convergence properties, the computational effort of exact sampling is relatively larger, but in absolute terms is less than it is for the difficult cases. Exact sampling always has the benefit of eliminating uncertainty about the amount of error, since the bias, or systematic error, is always zero.

Chapter 2

Belief Networks

2.1 Example - medical diagnosis

An expert system might be designed to assist parents in diagnosing diseases based on observable symptoms. The goal of the user is to decide on some action, either to go to the hospital, the family doctor, or just stay indoors. Evidence consisting of the presence or absence of each symptom is input into the system. The output of the system is the probability of each disease being present. The action taken would depend on an expected cost/benefit analysis. If the probability of pneumonia is only 1% and the hospital is 50 miles away, one would weigh the benefits (1% times the benefit of treating actual pneumonia) against the cost of driving. The system might give a warning about the validity of the evidence if the combination of given variables is highly unlikely, or decide if more definitive answers can be obtained by adding a new evidence variable by performing a costly diagnostic test.

The variables in the expert system illustrated in figure 2.1 represent diseases and symptoms. These are binary valued, either present or absent, and are not exclusive - e.g. a patient may suffer from more than one disease at once. The arrows represent causal relationships between the diseases and the symptoms. The strength of these relationships

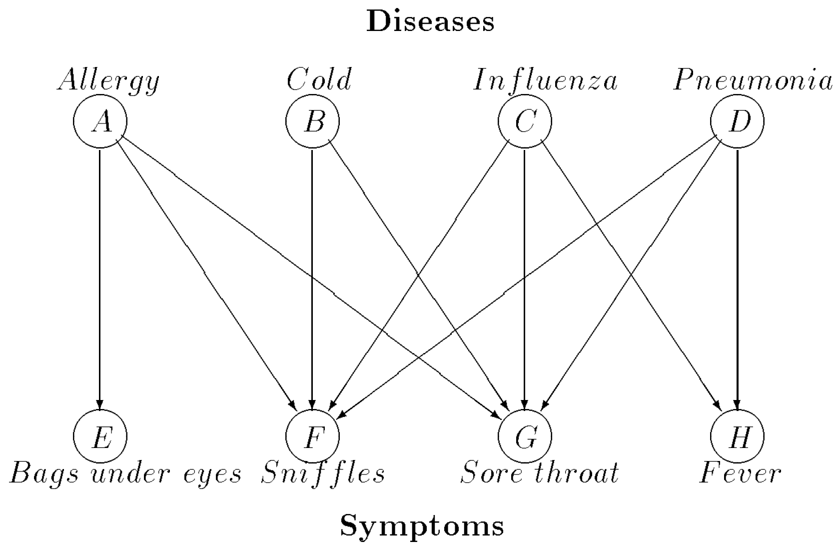


Figure 2.1: Medical Diagnosis System

may be obtained either from statistical frequencies or from the opinions of human experts. There are no arrows between symptoms, since they are independent of each other, given the diseases. Also, there are no arrows between diseases, since they are independent of each other until the symptoms are known.

For each configuration of disease variables, probabilities for symptoms must be specified. If there are n edges converging to a symptom, 2^n probabilities for that symptom being present are required. The fact that there are no edges between some of the diseases and symptoms means that the existence of that disease has no bearing on the likelihood of the symptom being present. In the example, $P(H|A,B,C,D) = P(H|C,D)$, so variables A and B can be disregarded.

The causation of "bags under eyes" is simple; there are only two probabilities, $P(E|A)$, the probability of bags under eyes given that allergy is present, and $P(E|\neg A)$, the probability of bags under eyes given that allergy is not present. These also determine the probability of not having bags under eyes, since the probabilities must sum to 1. Other symptoms can be caused by more than one disease, so a more complex model is needed.

More examples of belief networks can be found in Jensen (1996), where the following definitions are given.

2.2 Definition

A Belief network, also called a causal network or a Bayesian network, consists of:

- a set of variables represented by the nodes in a graph or network and a set of directed edges, forming a directed acyclic graph (DAG). A variable with an edge pointing to another variable is a parent of that variable.
- a set of mutually exclusive states for each variable.
- for each variable A with parents $B_1, \cdots, B_n$, a conditional probability table $P(A|B_1, \cdots, B_n)$.

If $A_1, \cdots, A_m$ is the set of variables of a belief network, then the joint probability distribution $P(A_1, \cdots, A_m)$ is the product of all conditional probabilities

$$P(A_1, \cdots, A_m) = \prod_i P(A_i| \; parents \; of \; A_i).$$

Any joint distribution may be represented as a Bayesian network, by writing the joint distribution as a product of conditional distributions:

$$P(A_1 = a_1, \cdots, A_m = a_m) =$$

$$P(A_1 = a_1)P(A_2 = a_2|A_1 = a_1) \cdots P(A_m = a_m|A_1 = a_1, \cdots, A_{m-1} = a_{(m-1)}),$$

and then using independence assumptions, e.g. $P(A_3|A_1, A_2) = P(A_3|A_1)$, to minimize the parent sets. There can be many different ways to represent a joint distribution as a belief network, but we usually prefer to use one that corresponds to natural relationships of cause and effect.

2.3 Noisy-Or scheme

For a variable related to many parents, the number of combinations of parent states to maintain in the table of conditional probabilities is unwieldy. Also it is not natural for

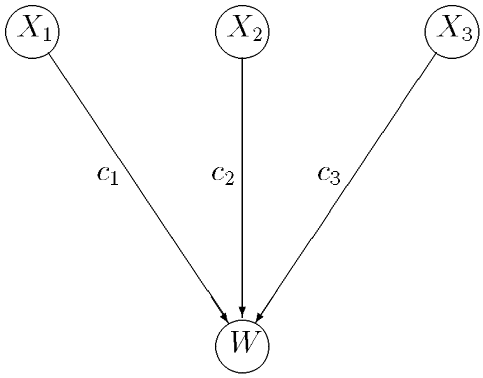


Figure 2.2: Noisy-OR scheme

an expert to attach probabilities conditioned on multiple parents. A solution to this problem is the Noisy-OR scheme, which provides a simplified way of specifying how parent variables influence the outcome of a child variable.

The noisy-or relationship defines the dependencies between a child variable and each of its parents. It specifies the degree of influence each parent has on the child, independently of the other parents, and then determines by calculation the conditional probability of the child variable given all of the parents. Thus the amount of information required to generate conditional probability tables is linear in the number of parents.

All variables are binary valued 0/1. Each parent variable influences the child variable to be turned on (value = 1) when it is turned on. The degree of influence is determined by a weight, c_i, on the link between the parent X_i and child W (see figure 2.2). This weight is the probability of turning on the child given that the parent X_i is turned on. If the weight is one for all links, the scheme is just an OR-gate. The probability $(1 - c_i)$ of failing to turn on the child determines the amount of noise in the noisy-or network. Variable W also possesses a prior probability $P_W = P(W = 1|X_i = 0, \forall i)$.

The noisy-or scheme assumes that the strength of the link between parent and child is independent of the other parents. This is perhaps realistic for a medical diagnostic system. For example, sniffles may be caused by the presence of either a cold or an allergy, but each disease does not affect the probabilities of the other disease failing to cause sniffles. The probability of all parents failing to set the child to 1 is the product of the probabilities of all such factors:

$$P(W=0|X_1,X_2,X_3) = (1-P_W) \times \prod_{i:X_i=1} (1-c_i).$$

2.4 Calculating Probabilities

Given that certain symptoms are present, what diseases are most likely? In the example, one might ask for the probability of having influenza, given that the only symptoms present are a sore throat and a fever, and it is known that pneumonia is absent. The desired information is not directly available from the belief network, which instead tells us directly about symptom probabilities if the diseases are known.

The conditional probability required is a quotient of marginal probabilities, since by definition, $P(A|B) = \frac{P(A,B)}{P(B)}$. For example,

$$P(C=1|D=0,E=0,F=0,G=1,H=1) = \frac{P(C=1,D=0,E=0,F=0,G=1,H=1)}{P(D=0,E=0,F=0,G=1,H=1)}.$$

The most direct way of evaluating these marginal distributions is to sum the joint distribution of all states, for every combination of the other variables. In this case the sum is over the combinations of only 2 and 3 other variables. Since the number of terms in a sum over k variables is 2^k (for binary variables), this calculation is not feasible for larger problems.

A more efficient method involves building a junction tree or tree of cliques (Jensen 1996). This provides a framework for updating the marginal distribution of variables when evidence instantiates some of the other variables. If a network is not in a tree structure, the variables are judiciously grouped together in such a way that the relationship between groups has a tree structure. The groups, called cliques, overlap by sharing the parents of variables in child groups. For example, if a network is already in tree form, its groups would consist of one group for each variable and would contain that variable plus the parents of the variable. Conditional probabilities for bidirectional link matrices between the cliques are calculated from the belief network conditional probabilities.

Neapolitan (1990) discusses the complexity of junction trees. A simple estimate of the complexity of building the tree of cliques is $O(nr^m)$, where n is the number of variables, r is the maximum number of alternatives for a variable, and m is the maximum number of variables in a clique. As the network becomes more densely connected, m increases, and the building time increases. The estimate is not valid when $m = n$, however, since the junction tree becomes one clique consisting of the entire belief network. In this case there is no work to do.

When evidence is received, the marginal distributions of each of the variables are updated by passing probability "messages" through the link matrices. The computational effort for updates is $O(pr^m)$, where p is the number of cliques. Queries are slowed by the grouping into cliques, since the number of states of each clique increases with the number of variables it contains. In the worst case, the entire network becomes one big clique and the method reverts to the first method of summing over all combinations of variables.

Since junction tree building and updating is an intractable problem for large densely connected networks, in general, exactly calculated solutions must be abandoned in favour of approximation techniques such as Markov Chain Monte Carlo (MCMC) sampling.

Chapter 3

Gibbs Sampling

3.1 Markov chains

A Markov chain is specified by:

- A sequence of discrete random variables $X^{(0)}, X^{(1)}, \cdots$,
- A marginal distribution for the initial state $X^{(0)}$,

$$p_0(x),$$

- Transition probabilities for state $X^{(t+1)}$ to follow state $X^{(t)}$,

$$P(x^{(t+1)} \mid x^{(t)}),$$

This determines the joint distribution of $X^{(0)}, X^{(1)}, \cdots$ by making the Markov assumption that $X^{(t)}$ is conditionally independent of $X^{(t-k)}$ for $k > 1$, given $X^{(t-1)}$.

Stationary Markov chains have transition probabilities that do not depend on time, which can be represented with a transition matrix M. The value in the $i'th$ row and $j'th$ column of the transition matrix, $M_{i,j}$, stands for the probability of a transition to state j given that the system is in state i. The rows of the transition matrix must therefore sum to 1. The state probabilities at time t (i.e. $P(X^{(t)} = j)$) can be represented as a row vector p_t, and $p_t = p_0 M^t$. A distribution π is invariant (or stationary) if $\pi = \pi M$. If $p_t = \pi$ for some t, then the Markov chain has reached the invariant distribution by time t

and will continue in that distribution forever. An ergodic Markov chain has an invariant distribution that is reached asymptotically no matter what the initial distribution p_0 is, i.e.

$$\lim_{t \to \infty} p_t = \pi.$$

A stationary Markov chain can be generated by a randomized subroutine $Markov_M()$ that takes as input some state i. It outputs state $j = Markov_M(i)$, with probability $M_{i,j}$.

3.2 Gibbs Sampling Markov Chains

Suppose that the state X consists of several variables $X_1, X_2, \cdots, X_n$. If these variables are binary, there will then be 2^n states. A transition matrix that leaves a desired distribution π invariant may be built from a set of base transition matrices representing transitions that can change only a single variable. For each variable X_k, a base transition matrix B_k is defined.

The transition probabilities specified by B_k are zero for those transitions that change any other variable besides X_k. In the Gibbs sampling scheme, the transition probabilities to states changing X_k are the conditional probabilities under π of the variable X_k taking on its various values, given the current values of the other variables. The value of X_k may remain the same or it may change. Its new value is independent of its previous value.

Each B_k leaves the distribution π invariant. This is intuitively so, since first of all the marginal distribution of any other variable besides X_k remains what it was before, and so will be that given by π if it was previously. Furthermore, the conditional distribution of X_k is explicitly forced to be that given by π. Therefore, the joint distribution which is the product of the marginal distribution and the conditional distribution will be π if the distribution for the previous state was π. This is shown by Neal (1993, pp. 36-52).

The transition matrix M can be defined as $M = B_1 B_2 \cdots B_n$ for a system of n

variables. The Markov chain with transition matrix M is simulated by applying each B_k in sequence. To generate the Markov chain, a randomized subroutine $Markov_{B_k}()$ takes as input some state i. It outputs state $j = Markov_{B_k}(i)$, with probability $(B_k)_{i,j}$. $Markov_M(i)$ can then be defined as $Markov_{B_n}(Markov_{B_{n-1}}(\cdots(Markov_{B_1}(i)\cdots)$.

The Markov chain is said to be ergodic if it converges to π regardless of the initial distribution. A base transition matrix B_k does not on its own define an ergodic Markov chain, since the probability of visiting most of the states in the state space is zero, given a particular initial state. However, the transition matrix $M = B_1 B_2 \cdots B_n$ will be ergodic if none of the conditional probabilities in the B_k's are zero. In fact, it is often the case that some of the base transition matrices do have zero conditional probabilities, but the Markov chain is still ergodic. Each case must be analyzed by itself.

3.3 Gibbs Sampling for Belief Networks

Pearl (1987) shows how to derive Gibbs sampling transition probabilities for a belief net. The necessary conditional probabilities of the base transition matrices are available directly from the specification of the belief network. When all the other variables are fixed, the conditional probability of a particular variable is dependent only upon its parents, its children, and its children's parents (figure 3.1). The conditional distribution of the variable can be expressed as

$$P(V_k|V_1, V_2, \cdots, V_{k-1}, V_{k+1}, \cdots, V_n) = \frac{P(V_1, \cdots, V_k, \cdots, V_n)}{P(V_1, V_2, \cdots, V_{k-1}, V_{k+1}, \cdots, V_n)}.$$

Since the denominator is constant with respect to V_k, this is proportional to $P(V_1, \cdots, V_k, \cdots, V_n)$. From the definition of belief networks,

$$P(V_k|V_1, V_2, \cdots, V_{k-1}, V_{k+1}, \cdots, V_n)$$

$$\propto P(V_k|V_1, V_2, \cdots, V_{k-1}) \times \prod_{j>k} P(V_j|V_1, V_2, \cdots, V_k, \cdots, V_{j-1}).$$

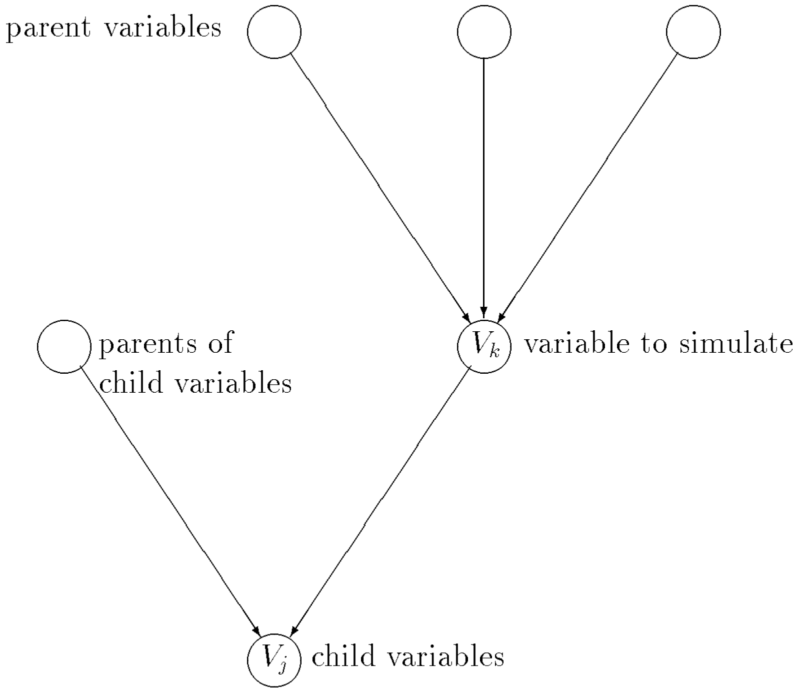


Figure 3.1: Conditional dependencies for Gibbs sampling

Here $V_1, V_2, \cdots, V_{k-1}$ are possible parents of V_k, and V_j for $j > k$ are possible children of V_k, since V is ordered with parents before children. Each factor in the expression can be evaluated using what Pearl refers to as the link matrix of the variable, which specifies the conditional distribution of a variable given its parents. When V_j is not actually a child of V_k, the factor for it can be omitted.

$$P(V_k = x | V_1, V_2, \cdots, V_{k-1}, V_{k+1}, \cdots, V_n) =$$
$$\frac{P(V_k = x | V_1, V_2, \cdots, V_{k-1}) \times \prod_{j>k} P(V_j | V_1, V_2, \cdots, V_k = x, \cdots, V_{j-1})}{\sum_u \left(P(V_k = u | V_1, V_2, \cdots, V_{k-1}) \times \prod_{j>k} P(V_j | V_1, V_2, \cdots, V_k = u, \cdots, V_{j-1}) \right)}.$$

In case evidence has been received that $V_k = x$, then V_k is fixed at this value during Gibbs sampling rather than being updated.

For noisy-or belief networks, if a child variable $V_j = 0$, then the other parents of the child can be ignored in the calculation. Child variables instantiated to 0 cannot transmit information between parents. Therefore, noisy-or weights on the edges between V_j and the other parents do not need to be included to calculate the probability of another parent besides V_k failing to instantiate V_j. Those terms cancel out in the quotient.

3.4 Initialization Bias

Gibbs sampling is subject to an initialization bias due to the arbitrary choice of the initial state. For example, an implementation of Gibbs sampling might set the initial state to have all variables equal to 0. Thus the distribution of the initial state is 100% in the all zeroes state and 0% in all other states. An initial state chosen randomly with a uniform distribution over all possible states will also not in general be the invariant distribution that we wish to sample from.

The bias or error in the distribution of the Markov chain at a given time can be measured by the total variation distance between the distribution of the state at that time and the invariant distribution. For a finite state space χ, where μ_t is the distribution of the Markov chain at time t, this is

$$||\mu_t - \pi|| = \frac{1}{2}\sum_{x\in\chi} |\mu_t(x) - \pi(x)|.$$

The error decays exponentially with time (figure 3.2), asymptotically according to

$$error = ae^{-t/c},$$

where a and c are constants that depend on the Markov chain specification. (Rosenthal (1995) discusses in depth the convergence rates of Markov chains on finite state spaces.) If the user has an error tolerance ϵ, the burn-in time for Gibbs sampling should be

$$burn\ in\ time = -c\ln(\epsilon/a).$$

However, the convergence behaviour of the Markov chain is usually not known (i.e. constants a and c are not known). The conservative user will try to choose as large a burn-in

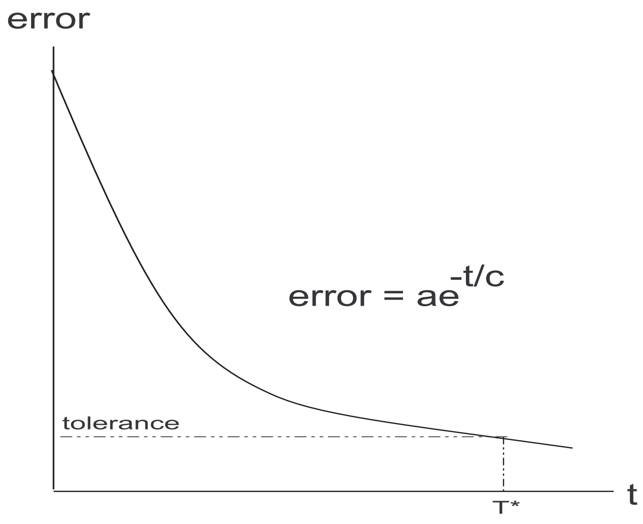


Figure 3.2: Error vs. Burn-in time

time as is practical to lessen the chances of it being too small. Computational considerations become more important for Markov chains that converge slowly, as it may be felt there is not enough time to acceptably reduce the risk. The next section deals with the coupling from the past technique, which eliminates initialization bias. Whether or not this is a practical alternative to standard Gibbs sampling with a burn-in period is addressed by comparing computational costs and considering the superior results due to elimination of error.

Chapter 4

Exact Sampling

4.1 The idea of coupling from the past

Propp and Wilson (1997) proposed exact sampling using coupling from the past as a solution to the problem of Gibbs sampling error and error uncertainty. Their method is able to obtain samples that are exactly from the desired distribution. They efficiently implemented an exact sampling algorithm that works for monotonic problems, where there is a partial ordering on the state space that is preserved through the Markov chain transitions. Coupling is a technique for allowing multiple chains to coalesce into one chain, by introducing dependancies between the chains. Coupling of chains had been used before in sampling schemes, by starting the coupling runs from the present (Johnson 1996). However, those schemes are unable to remove the error completely, but only provide some measure of the magnitude of the error.

Propp and Wilson use the fact that the invariant distribution, π, of an ergodic Markov chain on a finite state space can be reached if the chain is run for an infinite amount of time, no matter what the initial distribution p_0 is, that is

$$\lim_{t\to\infty} p_0 M^t = \pi,$$

where M is the transition matrix (see section 3.1). Therefore, if one were willing to wait forever, one could be sure that the correct distribution of the Markov chain had been reached. They show that it is not necessary to wait forever to arrive at this result, however: that there is a way to find the exact result with a finite number of computations. The state found in this way may be used as the starting state for a Gibbs sampling run that will be free of bias.

Coupling from the past starts chains at some time, $T < 0$, in the past starting from every possible state, and attempts to make the chains coalesce into one chain by introducing dependencies between the chains. The minimal requirement for coalescence, is that two chains arriving at the same state use the same pseudo-random number to make the transition to the next state. This causes them to stay together. In practice, coupling from the past uses the same pseudo-random variable at each time step for all of the chains, as this greatly facilitates separate chains to come together. If by time $t = 0$ they have all coalesced into one chain, then it can be said that no matter what state was started from at time $t = T$, the same state at time $t = 0$ results.

Propp and Wilson (1997) show (*Theorem 1, p. 228*) that with probability 1 coupling from the past returns a state, which is from the invariant distribution. The state returned is the final coalesced state of the chains at time $t = 0$. They consider the total simulation time, from the time the chains are started in the past to time $t = 0$, to be laid out in sequential time segments. A time segment is long enough that there is a positive probability $\epsilon > 0$ of chains in every possible state at the beginning of the segment to all coalesce into one chain by the end of the segment, assuming the chain is ergodic. Of course the time segment must be long enough to incorporate all of the base transitions in order for the chain to have a positive probability of visiting all states. The chains must be made dependent by using the same pseudo-random numbers, in order to have coalescence. This is described in more detail in section 4.2. If the length of one time segment is L, then time periods can be laid out from some starting time T in the past up to $t = 0$, as $(\cdots, (-3L, -2L), (-2L, -L), (-L, 0))$. The probabilities of having complete coalescence within each time segment are independent of each other. As T gets more negative, $T = -L, -2L, -3L, \cdots$, the probability of not coalescing in any of the segments, $(1-\epsilon)^{\frac{-T}{L}}$, gets smaller. Therefore, it can be said that coalescence occurs with probability 1 if T is allowed to be arbitrarily far in the past.

To show that the final coalesced state at time $t = 0$ is from the correct distribution, suppose that T is far enough back in the past that coalescence occurs. Then starting earlier than T makes no difference to what the final coalesced state is, since those chains must pass through the states at time $t = T$. In particular, starting at a time infinitely far back in the past results in the same final coalesced state. So running the experiment starting at time $t = T$ is equivalent to running the experiment starting infinitely far back in the past, as far as the final coalesced state is concerned. But any one of the chains of that experiment started infinitely far back in the past has converged to the invariant distribution π of the Markov chain. It ends up in a state at $t = 0$ that is randomly chosen from the correct distribution, with respect to the pseudo-random variables of the experiment. Therefore, performing the experiment by starting infinitely far back in the past results in a final coalesced state that is from the correct distribution. Since both experiments are equivalent, so does running the experiment started from time $t = T$.

A coalesced chain must be continued to time $t = 0$ before using the state for the initial Gibbs sampling step. Usually, coalescence occurs before reaching time $t = 0$, but if the state at that time is used, there is a bias introduced toward conditions that favour coalescence of chains. By always selecting the state at time $t = 0$, there is no dependence on the time that coalescence occurs.

If chains do not all coalesce when started at some time in the past, they are run again by restarting them further back in the past until coalescence finally occurs. The previous run that failed to coalesce must not be thrown away, but rather continued further back in the past. The explanation is that the randomly selected pseudo-random numbers completely determine the behaviour of the chain. If they are just thrown away and new

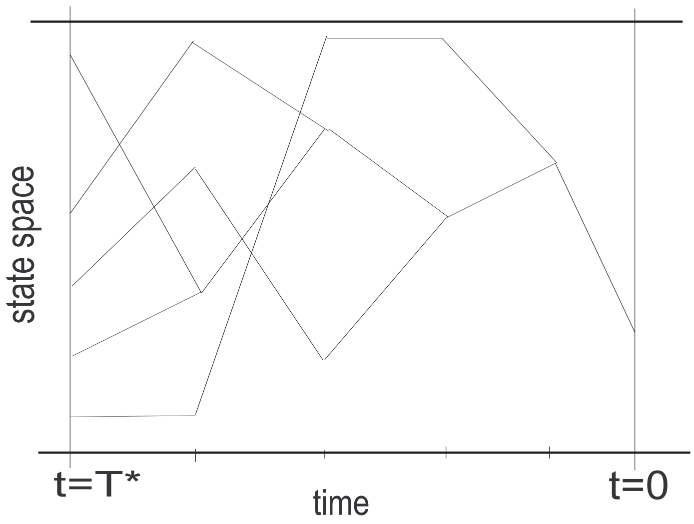


Figure 4.1: Chains Coupling

pseudo-random numbers generated, then there is a bias induced by preferring a selection of pseudo-random numbers that more easily allow coalescence.

It is not far from optimal to restart the chains at a time twice as far back in the past. Each time the chains are started they must be run all the way to time $t = 0$, so going back only one time step would be inefficient. It is more efficient to start runs at times $t = -1, -2, -4, -8, -16, \cdots$ until coalescence finally occurs. Propp and Wilson (1997) show that if $-T_*$ is the minimum number of time steps in the past necessary to achieve coalescence, then T found by the above method overshoots by less than a factor of 2 and the total number of simulation steps is less than $-4T_*$. The expected number of simulation time steps is about $-2.89T_*$. The expected value for T is about $(-2.89T_* + 1)/2$, which approaches $1.44T_*$ as T_* becomes more negative.

The coupling from the past procedure just described can be performed a number of times, each time with new pseudo-random variables, thus obtaining multiple independent states from the invariant distribution. Each time the procedure is run, it must search for a starting time that allows the chains to coalesce. The procedure typically requires varying amounts of running time to complete (i.e. having to go back varying amounts of time in the past to cause the chains to coalesce). In practice, if any of these independent attempts are observed to find appropriate starting times for allowing the chains to coalesce, it is reasonable to expect all such attempts to succeed in an observable time period. This is because the chances of coalescing further back in the past are independent of the failure to coalesce later on. If starting at time t fails to make the chains coalesce, a new attempt is made to get the chains to coalesce from time $2t$ to $t-1$, and the ending states (which may be only one state) used as the starting states of the previous attempt. However, if none of the procedure runs seem to coalesce, one must declare the results indeterminate, which is superior to getting a wrong answer using Gibbs sampling with the same number of steps.

Once Gibbs sampling is started with a state from the invariant distribution, any number of samples may be taken from the following chain. However, it is desirable to find a number of starting states from the invariant distribution and take samples from each of the chains that follow them. The starting states are more valuable than the states that follow them, since they are completely independent of each other, but they come at the cost of coupling chains from the past. At the same time, the following states are less valuable because of their dependence on prior states, but they can be produced at the much lower cost of one Markov chain transition.

4.2 Speeding up coalescence

In order to speed up coalescence, Propp and Wilson introduce additional dependencies between the chains by using the same i.i.d. pseudo-random variables for all chains. This goes beyond the minimal requirement for the chains to coalesce, which is only that chains in the same state use the same pseudo-random variables in order to stay together. The dependencies do not change the validity of the Markov chains, since transitions made at different times continue to be independent.

Exactly what constitutes the set of pseudo-random variables depends on how the generation of the Markov chains is implemented. For Gibbs sampling transitions, only one state variable is allowed to change at a time (section 3.2), which requires a random choice according to the conditional probability of the variable, given the other variables. Which variable to change may be selected by cycling through a pre-determined sequence of all of the state variables, or by random selection from the set of state variables, which requires an additional random choice. For this thesis, Gibbs sampling is considered to cycle through the sequence of state variables, so randomness is required only for setting the value of the selected state variable. Therefore, a single pseudo-random variable is sufficient for making the random choice at each time step, and should be the same for all of the chains.

Using the same pseudo-random variables to make transitions for each of the chains does speed up the coalescence of the chains. If a transition is performed to change the only variable that differs between two states, then using the same pseudo-random variable to determine the value of the variable will certainly cause the chains to coalesce. This is true because having all of the other variables equal causes the conditional probability of the transition to be the same for both chains. On the other hand, if different pseudo-random variables are used for chains that have not coalesced, the probability of the chains coalescing can be low when there are several alternatives for the next state.

The interdependence of chains within a time step is implemented by a sequence of deterministic functions $\phi_t(.,.)$ and i.i.d. random variables $\cdots, U_{-3}, U_{-2}, U_{-1}$, which are the source of all randomness. If the transition probabilities are the same for all time steps t, then $\phi_t(.,.)$ is the same for all t. Once the i.i.d. random variables U_t are established, they are paired with the deterministic function ϕ_t to define completely deterministic functions

$$f_t(i) = \phi_t(i, U_t),$$

which specify the transition from a state i at time t to a state $f_t(i)$ at time $t+1$. Thus all of the states i at any particular time step t derive their transitions from the same i.i.d. random variable U_t. $\phi_t(i, U_t)$ can be defined to return the smallest j such that $M_{i,1}^{(t)} + M_{i,2}^{(t)} + \cdots + M_{i,j}^{(t)} > U_t$, where $M^{(t)}$ is the matrix of transition probabilities for the Markov chain at step t, and U_t is uniformly distributed on the interval [0,1).

For noisy-or belief networks, the transition matrix M is composed of base transitions, as $M = B_1 B_2 \cdots B_n$. The transitions at each time step t are composed of n deterministic functions $\phi_{t,k}$, for $k = 1, ..., n$, each governed by the conditional probabilities given by B_k:

$$f_t(i) = \phi_{t,n}(\phi_{t,n-1}(\cdots(\phi_{t,1}(i, U_{t,1})), \cdots), U_{t,n-1}), U_{t,n}).$$

Provided the same base transition matrix is used to modify the k'th variable at each time step, $\phi_{t,k}$ is the same for all t. If i and i' are two states with variable k flipped, and $V_k = 0$ for state i, $\phi_{t,k}(i, U_{t,k})$ and $\phi_{t,k}(i', U_{t,k})$ can be defined to return state i if $(B_k)_{i,i} > U_{t,k}$, else return state i'. The distribution of $U_{t,k}$ in this case is uniform on the interval [0,1). The conditional distribution of $\phi_{t,k}$ is that given by π, since for any state j, $P(\phi_{t,k}(i, U_{t,k}) = j) = (B_k)_{i,j}$. Therefore, as discussed in section 3.2, $\phi_{t,k}$ and $f_t(i)$ preserve the invariant distribution. The simulation of the chain from the starting time $t = T$ in the past to time $t = 0$ is the composition

$$F_T^0 = f_{-1}(f_{-2}(...(f_T(i))...)).$$

Propp and Wilson prove in *(Theorem 3, p. 230)*, that if $F_T^0(i)$ is the same for all states i, and if the transitions produced by f_t preserve the invariant distribution, then F_T^0 has the invariant distribution.

$F_t^0(i)$ returns the state of a Markov chain started in state i at step t and ending at step 0. If started far enough back in time, F_t^0 returns a constant ending state j for all starting states, indicating that all of the Markov chains have coalesced sometime between time step t and time step 0. The distribution of this j with respect to all of the i.i.d. random variables $U_t, U_{t+1}, \cdots, U_{-1}$ is π, which is what we want to begin Gibbs sampling

$T \leftarrow -1$

Repeat

$V \leftarrow \emptyset$

For i = every possible starting state

$V \leftarrow V \bigcup F_T^0(i)$

$T \leftarrow 2T$

until V contains only one state

Begin Gibbs sampling with the initial state contained in V.

Figure 4.2: Algorithm for Exact sampling

without initialization bias. There is a maximum (least negative) t that will result in a constant F_t^0. The algorithm employs the near optimal search strategy to find some T such that (i) $2t + 2 \leq T \leq t$, and (ii) F_T^0 is constant and evaluates to the same state as F_t^0 (figure 4.2). The worst case of $T = 2t + 2$ results from missing t by one, at $t + 1$, and having to double the time again. When a coupling run must be restarted farther back in the past because of failure to coalesce, the i.i.d. random variables already generated are reused until coalescence finally occurs.

Keeping track of every chain for every possible starting state of a coupling run is an intractable procedure, as the state space size increases. Therefore, strategies are required to reduce the complexity. Propp and Wilson use the property of monotonicity of state spaces to find an efficient algorithm. This thesis uses an approximation of the set of states to be tracked, to obtain the same resulting output state from the invariant distribution.

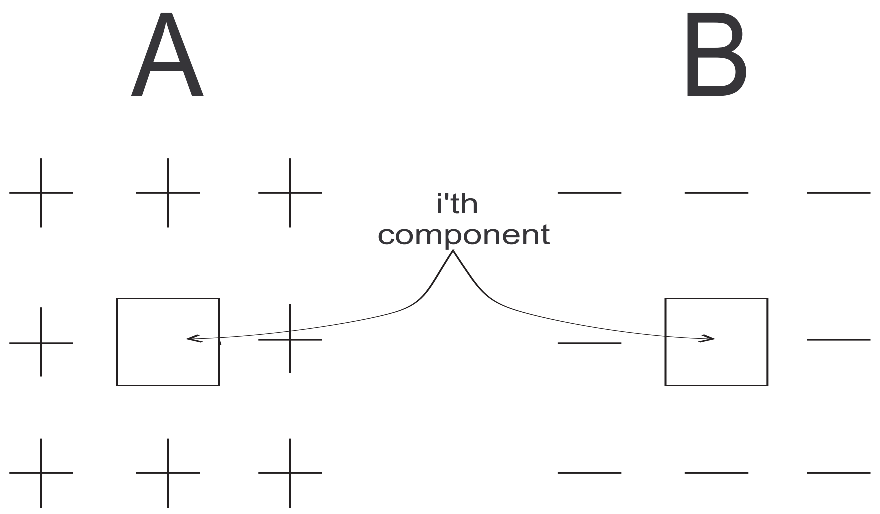


Figure 4.3: Ising spin system

4.3 Comparison to Gibbs sampling

4.3.1 Monotonic problems

A monotonic chain has a state space with a partial ordering that is preserved through Markov chain transitions. Propp and Wilson (1997) show that for monotonic chains, there is a way to couple the chains that is often very efficient. Furthermore, they show that Gibbs sampling can not converge much faster than coupling can cause chains to coalesce. This makes a strong case for using exact sampling with monotonic problems.

The Ising spin system is an example of a monotonic problem. It consists of a spatial arrangement of "spins", either $+$ or $-$, each of which tends to have a value matching its nearest neighbours. The more positive neighbours a spin has, the more strongly it is influenced to be positive itself.

The state space of the Ising system can be partially ordered. Each variable is ordered as "$+$" $>$ "$-$". For states A, B with variables A_i, B_i,

$$A \geq B \iff A_i \geq B_i, \ \forall i.$$

This ordering is preserved between Markov chain transitions. When making the transition for the $i'th$ variable, if $A \geq B$, then for the neighbours of the $i'th$ variable, $A_j \geq B_j,\ j \neq i$. Each neighbour A_j exerts as least as much influence for A_i to be "+" as the neighbour B_j does for B_i. Therefore, in total, A_i is at least as strongly influenced to be "+" as B_i is.

$$A \geq B \Longrightarrow P(A_i = \text{“}+\text{”}|A_{-i}) \geq P(B_i = \text{“}+\text{”}|B_{-i}).$$

where A_{-i}, B_{-i} are the other variables besides A_i, B_i. Since the same pseudo-random variable is used to generate the transitions for both chains, it cannot be that B_i will become a "+" while A_i becomes a "−". Thus, the partial ordering is preserved; the chain is monotonic.

In order to make a strong case for using exact sampling, it is important to show that Gibbs sampling cannot work well when coalescence is slow. For monotonic chains, Propp and Wilson (1997) make the assertion that "if the Markov chain is rapidly mixing then it is rapidly coupling".

The threshold mixing time of the chain is described in terms of the total variation distance, as:

- $d(k) = \max\limits_{\mu,v} ||\mu_k - v_k||$ is the maximum total variation distance between the distributions with k transitions after starting from any two random states governed by distributions μ and v.

- $T_{MIX} = \min\limits_{k} : d(k) \leq \frac{1}{e}$ is the mixing rate of the chain which is the minimum running time to reduce the maximum total variation distance to at most $\frac{1}{e}$, where $e = 2.718\cdots$.

A lower bound on T_{MIX} is given in terms of

- $E[T^*]$, the expected coupling time of the chain

- l, the length of the longest chain in the partially ordered state-space (the number of spins, for the Ising system) as

$$T_{MIX} \geq \frac{E[T^*]}{2(1+\ln l)}.$$

For the Ising spin system, coupling can be done very efficiently by simulating only two chains. There are unique maximal (all "+") and minimal (all "−") states in the state space, that are greater and less than all other states. By simulating chains for only the maximal and minimal starting states, a bound on the range of all the other chains with respect to the ordering is maintained. When the maximal and minimal chains coalesce, it is known that all of the other chains have coalesced as well.

4.3.2 Non-monotonic problems

Belief network states usually cannot be ordered in a way that makes the Markov chain monotonic. This presents some challenges because there is no known way to ensure the coalescence rate is not much greater than the Gibbs sampling convergence rate, and an alternative to tracking every chain in the state space is needed.

For non-monotonic problems, there is no known way to specify a lower bound on the number of steps to achieve an error tolerance in terms of the coalescence rate. That presents the possibility of there being cases where Gibbs sampling converges rapidly when coupling from the past takes a long time. There are no known examples of this, however. Although the tight relationship between coupling and convergence can not be made, one would expect conceptually a close relationship. An example can be made with a phenomenon that influences both coupling and convergence, namely the presence of modes. Many small movements will be made around some locality of the state space before a transition is made to another area of the state space. If the chain is started from some distribution that is biased towards one of the modes, then ordinary Gibbs sampling will take a long time to visit all of the other areas of the state space with the desired frequencies. Hence the convergence is poor for Gibbs sampling, and a longer burn-in period is required to minimize the chance of exceeding the error tolerance. Similarly, exact sampling will take a long time for coalescence of the chains. The probability of a transition of two chains to the same state can be quite low if they start in different modes, and it could take a long time to make a transition into or out of the mode. Since the relationship between coalescence and convergence rates is not proven or disproved, a recommendation to use exact sampling with non-monotonic problems must be weaker than for monotonic problems.

To address the efficiency issue, a scheme is needed to simplify the tracking of all the chains. For noisy-or belief networks, this thesis attempts to summarize the chains with one chain on a state set space, where states represent sets of states of the chains to be coupled. The amount of work for each transition of the summary chain is the same as for the two chains of the monotonic problem, provided that none of the sibling variables of the network are directly connected. However, the method suffers somewhat from the inability to track precisely the set of chains, which tends to result in slower coalescence. That is the topic of the next chapter.

Chapter 5

Exact Sampling for Noisy-Or Belief Networks

Coalescence of a large number of Markov chains for a noisy-or belief network can be determined by looking at one chain on a state set space. Each state of the chain on the state set space is an approximation to a set of states of the chains being coupled.

5.1 Approximating a set of states

The state space S of noisy-or belief networks has variables taking on values 0 or 1. A set of states in S is approximated by a single state in a state set space $S^{(?)}$ with variables taking on values 0, 1, or ?. The mapping from the state set space to the set of states, $\beta : S^{(?)} \to S$, is

$$\beta(V^{(?)}) = \{V \in S |\ V_i^{(?)} \in \{V_i, ?\}, \forall i\}.$$

β selects all the states in S where every variable matches the corresponding variable of the single state in $S^{(?)}$. If a variable in $S^{(?)}$ is 0 or 1, then only states in S that have the same value for that variable can be selected. If a variable in $S^{(?)}$ is ?, then states in S

can be selected that have a value of 0 or 1 for that variable. For example,

$$\beta(1?001?) = \left\{ \begin{array}{c} 100010 \\ 100011 \\ 110010 \\ 110011 \end{array} \right\}.$$

β does not map some element of $S^{(?)}$ to every set of states in S. For instance,

$$\left\{ \begin{array}{c} 100010 \\ 100011 \\ 110010 \end{array} \right\}$$

cannot be exactly represented in $S^{(?)}$. If none of the states are to be lost, then the approximation by a state in $S^{(?)}$ must sometimes include spurious states.

5.2 Approximating a set of chains

A single chain on the state set space $S^{(?)}$ can approximate a set of chains in S, such that none of the true chains are lost. Some spurious chains are introduced by spurious states appearing at various time steps during the simulation. The spurious chains mask coalescence of the true chains, until all of the spurious chains also coalesce with the coalesced true chain.

The chain is simulated by performing base transitions for each variable that allow only one variable at a time to change. For each state in S, a particular variable has some conditional probability of changing to a 1, given the other variables in that state. Over the set of states that $V^{(?)}$ maps to, this conditional probability will have some maximum and minimum values. These maximum and minimum conditional probabilities can be used to determine the transition probabilities of the variable in $S^{(?)}$, using the fact that each chain makes the transition governed by the same pseudo-random variable U. For a given U, it is known if the variable will change to 0 or 1 in all of the states or if the

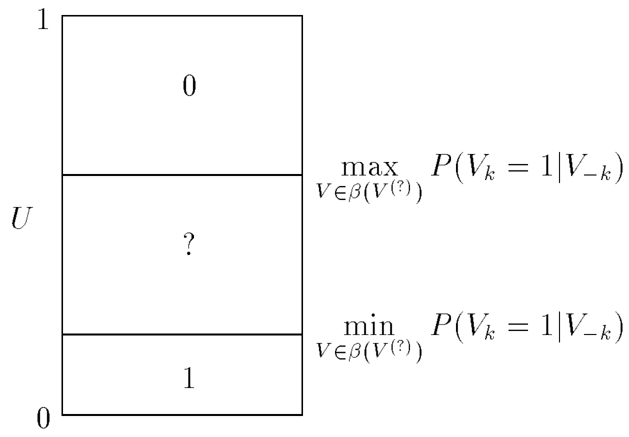


Figure 5.1: Determining a transition in state set space

variable will be 0 in some of the states and 1 in other states. The procedure for making the transition using a pseudo-random variable U is:

- If U is less than the minimum probability, then all of the states will make a transition to set the variable to 1. Set the variable in $S^{(?)}$ to 1.

- If U is greater then or equal to the maximum probability, then all of the states will make a transition to set the variable to 0. Set the variable in $S^{(?)}$ to 0.

- If U is between the minimum and maximum probabilities, then there may be states that make transitions to either 0 or to 1. Set the variable in $S^{(?)}$ to ?.

The conditional distribution of variable $V_k^{(?)}$ is illustrated in figure 5.1. V_{-k} is the set of other variables besides V_k. The transition probabilities are given in table 5.1. Transitions done this way do not lose track of any chains in the set of chains, by ensuring their new states are represented by the new state of the summary chain.

Spurious chains are introduced because the approximation of a set of states cannot exactly represent all combinations of states. This happens when a variable in $S^{(?)}$ changes from 0 or 1 to ?, and may happen when the variable remains as ?. For example, if a

$V_k^{(?)}$	$Prob(V_k^{(?)} \mid V_{-k}^{(?)})$
1	$\min_{V \in \beta(V^{(?)})} P(V_k = 1 \mid V_{-k})$
?	$\max_{V \in \beta(V^{(?)})} P(V_k = 1 \mid V_{-k}) - \min_{V \in \beta(V^{(?)})} P(V_k = 1 \mid V_{-k})$
0	$1 - \max_{V \in \beta(V^{(?)})} P(V_k = 1 \mid V_{-k})$

Table 5.1: Transition probabilities of chain in state set space

transition in $S^{(?)}$ is done by changing the last variable of

$$\beta(1?0010) = \left\{ \begin{array}{c} 100010 \\ 110010 \end{array} \right\}$$

but transitions in S are to different values in each chain, e.g.

$$\left\{ \begin{array}{c} 100010 \\ 110011 \end{array} \right\}$$

the transition in $S^{(?)}$ is to $\beta(1?001?)$ and adds two spurious chains. If in changing the last variable of $\beta(1?001?)$, there is coalescence in S so that two or three chains remain, and they have different values for that variable, the state in $S^{(?)}$ does not change and continues to map to four chains in S, adding one or two spurious chains. The effect of changing a variable in $S^{(?)}$ is shown in table 5.2.

5.3 Transition Probabilities for the Summary Chain

The minimum and maximum conditional probability, $P(V_k = 1 \mid V_{-k})$, over all of the states can be determined without searching by selecting certain states from the mapping $\beta : S^{(?)} \to S$. Let $pa(V_i)$ be the parent set of V_i, and $c(V_i)$ be the child set of V_i. The conditional probability for updating variable V_k is

$$P(V_k = x \mid V_{-k}) \propto P(V_k = x \mid pa(V_k)) \times \prod_{V_j \in c(V_k)} P(V_j \mid pa(V_j), V_k = x).$$

previous	new	effect on number of chains	spurious chains added
0/1	0/1	no change	none
0/1	?	double	2^z
?	0/1	halve	none
?	?	no change	0 to $(2^{z-1}-1)$

(z is the previous number of ?'s)

Table 5.2: Effect of changing a variable in state set space

Minimum probability that $V_k = 1$:

$$V \in \beta(V^{(?)}) : \left\{ \begin{array}{ll} V_j = 0 \text{ if } V_j^{(?)} = ? & \text{and } V_j \text{ is a child or parent of } V_k \\ V_p = 1 \text{ if } V_p^{(?)} = ? & \text{and } V_p \text{ is a parent of a child, } V_j\text{, of } V_k \\ & \text{and } V_j^{(?)} = 1 \end{array} \right\}$$

Maximum probability that $V_k = 1$:

$$V \in \beta(V^{(?)}) : \left\{ \begin{array}{ll} V_j = 1 \text{ if } V_j^{(?)} = ? & \text{and } V_j \text{ is a child or parent of } V_k \\ V_p = 0 \text{ if } V_p^{(?)} = ? & \text{and } V_p \text{ is a parent of a child, } V_j\text{, of } V_k \\ & \text{and } V_j^{(?)} \in \{?, 1\} \end{array} \right\}$$

Table 5.3: States with minimum and maximum conditional probabilities for $V_k = 1$

The desired minimum and maximum conditional probability is obtained by respectively minimizing and maximizing the ratio

$$\frac{P(V_k = 1|pa(V_k)) \times \prod_{V_j \in c(V_k)} P(V_j|pa(V_j), V_k = 1)}{P(V_k = 0|pa(V_k)) \times \prod_{V_j \in c(V_k)} P(V_j|pa(V_j), V_k = 0)}.$$

The rules for selecting the appropriate states are summarized in table 5.3.

Provided that sibling variables are not directly connected, then to minimize or maximize $P(V_k = 1|V_{-k})$, it is sufficient to minimize or maximize, respectively, each of the conditional probability ratios, $\frac{P(V_k=1|pa(V_k))}{P(V_k=0|pa(V_k))}$ and $\frac{P(V_j|pa(V_j),V_k=1)}{P(V_j|pa(V_j),V_k=0)}$ for $V_j \in c(V_k)$. To justify this, each of the terms in the product $P(V_k = x|pa(V_k)) \times \prod_{V_j \in c(V_k)} P(V_j|pa(V_j), V_k = x)$ are independent of each other, except where two children share a different parent besides V_k. However, the setting of that other parent is still the same for arriving at the minimum and maximum conditional probability ratio for each child.

For the variables that can be both 0 and 1 in a set of states, there are three cases to consider for the settings in the conditional probability expression:

1. **Parent variables :** It is clear from the noisy-or definition that the minimum and maximum of $\frac{P(V_k=1|pa(V_k))}{P(V_k=0|pa(V_k))}$ is obtained by selecting the states with parent variables equal to 0 and to 1 respectively.

2. **Child variables :** To show that the child V_j of V_k should be 0 to minimize and 1 to maximize the ratio $\frac{P(V_j|pa(V_j),V_k=1)}{P(V_j|pa(V_j),V_k=0)}$, it is shown that

$$\frac{P(V_j = 0|pa(V_j), V_k = 1)}{P(V_j = 0|pa(V_j), V_k = 0)} < \frac{P(V_j = 1|pa(V_j), V_k = 1)}{P(V_j = 1|pa(V_j), V_k = 0)}.$$

Let ω_k be the noisy-or weight between V_k and V_j. The right side is

$$\frac{1 - P(V_j = 0|pa(V_j), V_k = 0) \times (1 - \omega_k)}{1 - P(V_j = 0|pa(V_j), V_k = 0)}.$$

The left side is

$$\frac{P(V_j = 0|pa(V_j), V_k = 0) \times (1 - \omega_k)}{P(V_j = 0|pa(V_j), V_k = 0)} = 1 - \omega_k$$

$$= \frac{(1 - P(V_j = 0|pa(V_j), V_k = 0)) \times (1 - \omega_k)}{1 - P(V_j = 0|pa(V_j), V_k = 0)}$$

$$= \frac{(1 - \omega_k) - P(V_j = 0|pa(V_j), V_k = 0) \times (1 - \omega_k)}{1 - P(V_j = 0|pa(V_j), V_k = 0)},$$

which is less than the right side.

3. **Parents of child variables :** It is required to find the values of another parent V_p of a child V_j of V_k, $p \neq k$, that minimize and maximize the ratio $\frac{P(V_j|pa(V_j),V_k=1)}{P(V_j|pa(V_j),V_k=0)}$. There are three cases to consider:

(a) **The child variable $V_j = 1$:** To show that when $V_j = 1$, another parent $V_p (p \neq k)$ of V_j is 1 to minimize and 0 to maximize the ratio $\frac{P(V_j|pa(V_j),V_k=1)}{P(V_j|pa(V_j),V_k=0)}$, it is shown that

$$\frac{P(V_j = 1|pa(V_j), V_p = 1, V_k = 1)}{P(V_j = 1|pa(V_j), V_p = 1, V_k = 0)} < \frac{P(V_j = 1|pa(V_j), V_p = 0, V_k = 1)}{P(V_j = 1|pa(V_j), V_p = 0, V_k = 0)}.$$

Let ω_p be the noisy-or weight between V_p and V_j. For brevity, let $P(V_j = 0|pa(V_j), V_p = 0, V_k = 0)$ be denoted by α. The right side is

$$\begin{aligned}\frac{1 - P(V_j = 0|pa(V_j), V_p = 0, V_k = 0) \times (1 - \omega_k)}{1 - P(V_j = 0|pa(V_j), V_p = 0, V_k = 0)} &= \frac{1 - \alpha(1 - \omega_k)}{1 - \alpha} \\ &= \frac{(1 - \alpha(1 - \omega_p))(1 - \alpha(1 - \omega_k))}{(1 - \alpha(1 - \omega_p))(1 - \alpha)}.\end{aligned}$$

The left side is

$$\begin{aligned}&\frac{1 - P(V_j = 0|pa(V_j), V_p = 0, V_k = 0) \times (1 - \omega_p)(1 - \omega_k)}{1 - P(V_j = 0|pa(V_j), V_p = 0, V_k = 0) \times (1 - \omega_p)} \\ &= \frac{1 - \alpha \times (1 - \omega_p)(1 - \omega_k)}{1 - \alpha \times (1 - \omega_p)} \\ &= \frac{(1 - \alpha(1 - \omega_p)(1 - \omega_k))(1 - \alpha)}{(1 - \alpha(1 - \omega_p))(1 - \alpha)} \\ &= \frac{1 - \alpha(1 - \omega_p)(1 - \omega_k) - \alpha + \alpha^2(1 - \omega_p)(1 - \omega_k)}{(1 - \alpha(1 - \omega_p))(1 - \alpha)} \\ &= \frac{1 - \alpha(1 + (1 - \omega_p)(1 - \omega_k)) + \alpha^2(1 - \omega_p)(1 - \omega_k)}{(1 - \alpha(1 - \omega_p))(1 - \alpha)} \\ &= \frac{1 - \alpha((1 - \omega_p) + (1 - \omega_k)) + \alpha^2(1 - \omega_p)(1 - \omega_k) - \alpha\omega_p\omega_k}{(1 - \alpha(1 - \omega_p))(1 - \alpha)} \\ &= \frac{(1 - \alpha(1 - \omega_p))(1 - \alpha(1 - \omega_k)) - \alpha\omega_p\omega_k}{(1 - \alpha(1 - \omega_p))(1 - \alpha)},\end{aligned}$$

which is less than or equal to the right side.

(b) **The child variable $V_j = 0$:** If the child variable is 0, it does not matter what the other parents are.

(c) **The child variable $V_j = ?$:** To minimize or maximize the ratio $\frac{P(V_j|pa(V_j),V_k=1)}{P(V_j|pa(V_j),V_k=0)}$, the other parents of V_j besides V_k should be set as they are when $V_j = 0$ (it does not matter) or $V_j = 1$ (0), respectively.

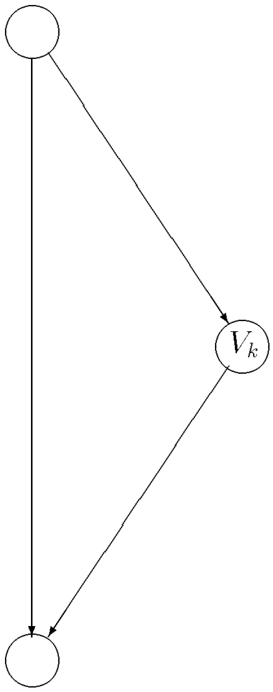


Figure 5.2: Directly-connected sibling variables

It is assumed that sibling variables are not directly connected, as in figure 5.2. That is, a child does not share a parent with its parent V_k. If this were so, the setting of that parent could have opposite effects on the conditional probability ratios for the child and for V_k. This happens if the child of V_k can be 1, and the value of the other parent is not given by evidence. For the two-level networks examined in this thesis, and more generally for any layered network where arrows go down only one layer, this case does not happen.

5.4 Algorithm

The algorithm for exact sampling using a summary chain is shown in figure 5.3. If the chain is allowed to run for long enough, the state of the summary chain will represent a single state. Since no chains are lost in the approximation, the true set of chains have coalesced when that happens. However, the summary chain may be late in detecting

$T \leftarrow -1$

Repeat

$(V^{(?)})_k \leftarrow ?, \forall k.$

run summary Markov chain with initial state $V^{(?)}$

from time $t = T$ to $t = 0$, outputting state $V^{(?)}$.

$T \leftarrow 2T$

until $V^{(?)}$ contains no ? variables

Begin Gibbs sampling with initial state $V \in S : V_i = V_i^{(?)},\ \forall i.$

Figure 5.3: Algorithm for Exact sampling using Summary chain

coalescence because of the spurious chains that can be added to the approximation.

A search strategy for a starting time successively tries powers of 2, $-T_? = 1, 2, 4, 8, ...$, until the summary chain represents one state by time $t = 0$. This starting time is typically not the latest possible starting time for obtaining a single state at time $t = 0$, which could be found by repeatedly decrementing T by 1. Propp and Wilson (1997) show that the search strategy requires an average total number of simulation of time steps around -2.89 times the latest possible starting time (section 4.1). The total amount of work is the number of simulations to be done at each time step times the total number of time steps simulated. There are two calculations of conditional probability at each time step. Therefore, the expected computational work is -5.78 times the latest possible starting time.

The performance of the summary chain method is explored in the following chapters. In chapter 6 the reasons for the delay in detecting coalescence due to spurious chains are studied. In chapter 7 the additional work factor due to the delay is determined empirically as a multiple of the number of time steps necessary to couple the chains when tracking every chain. In chapter 8 the performance of the proposed summary chain method is evaluated in comparison to the standard method, Gibbs sampling with a burn-in period discarded.

Chapter 6

Coalescence Time

Coalescence time is the minimum number of time steps in the past that are required for the coupled chains to coalesce by time $t = 0$. The benchmark for the coalescence time of the summary chain method is the coalescence time for tracking every chain. The relationship between the two is explored by examining small problems and studying eigenvalues and test results.

6.1 Transition matrix eigenvalues

The convergence of an ergodic Markov chain is related to the magnitudes of the eigenvalues of the transition matrix M. There is an eigenvector specifying the invariant distribution π of the Markov chain with eigenvalue equal to 1. The magnitudes of the other eigenvalues are less than 1. If the rows of the transition matrix are the transition probabilities out of states, then the eigenvectors are left eigenvectors.

As the Markov chain is generated from an initial distribution p_0, the distribution at time n is $p_n = p_0 M^n$. As $n \to \infty$, $p_n \to \pi$. If the absolute value of eigenvalues of M are $|\lambda_1| = 1, |\lambda_2| < 1, \cdots$, then the absolute values of eigenvalues of M^n are $|\lambda_1^n| = 1, |\lambda_2^n|, \cdots$. As $n \to \infty$, $|\lambda_i^n| \to 0, i > 1$. The larger the magnitude of the second-largest eigenvalue, the slower the convergence of the Markov chain.

The transition matrix of the Markov chain on the summarized set of states has all the eigenvectors and eigenvalues of the Markov chain on the state space of (0/1) variables. In particular, the eigenvectors are the same that have eigenvalues equal to 1. Hence, they both converge to the same invariant distribution. It contains the second largest eigenvalue of the chain on the states of (0/1) variables. In addition, there are some eigenvectors and eigenvalues associated with states with ?-valued variables, which may be larger in magnitude than the second largest eigenvalue of the chain on the states of (0/1) variables. Therefore, the chain on the summarized set of states cannot converge any faster than the chains it summarizes, and may actually converge more slowly.

The eigenvectors and eigenvalues of the transition matrices of some simple problems can be examined to verify that the results obtained through testing actually reflect the true characteristics of the problems. This lends further credence to results obtained for more complex problems.

6.2 Experimental tool

The method of the chain of summarized set of states is compared against the method of keeping track of every chain, by doing test simulations. For a given two-level belief network, and observations of known variables, a number of samples are collected from the step $t = 0$ by repeated sampling from the past. Each sample is collected in both ways, and the number of steps $-T_?$ required for the summary chain is compared to the number of steps $-T_{ES}$ required for the tracking every chain. The occurrences of the vector $(-T_{ES}, -T_?)$ are plotted on a 2-dimensional histogram (see figure 6.1 for example and figure 6.2 for interpretation).

The average $(-T_{ES}, -T_?)$ characterizes each test and is used for comparison with other tests. The $Median$ $-T_{ES}$ is used to determine the ratio between median coalescence times and average coalescence times, which is necessary to relate the work of exact sampling

Histogram of $(-T_{ES}, -T_?)$

$-T_{ES}$

$-T_?$	1	2	4	8	16	32	64	
256	0	0	0	0	1	1	1	3
128	0	0	0	5	9	5	1	20
64	0	0	0	7	7	7	2	23
32	0	1	2	7	6	11		27
16	0	0	1	3	12			16
8	0	0	0	4				4
4	0	0	4					4
2	0	2						2
1	1							1
	--	--	--	--	--	--	--	--
	1	3	7	26	35	24	4	100

$$Average(-T_{ES}, -T_?) = (18.3, 59.7),\ \frac{Average\ T_?}{Average\ T_{ES}} = 3.27$$

$$Median\ -T_{ES} = 16,\ \frac{Average\ T_{ES}}{Median\ T_{ES}} = 1.14$$

Figure 6.1: Example coalescence-time histogram

result	interpretation
Entries concentrated near the diagonal	summary chain coalesces as quickly as keeping track of every state
Entries concentrated above the diagonal	summary chain does not coalesce as quickly as keeping track of every state
Entries concentrated in columns with low value of $-T_{ES}$	Gibbs sampling mixes well
Entries concentrated in columns with high value of $-T_{ES}$	Gibbs sampling probably does not mix well (but this is not guaranteed)
Entries below the diagonal	Impossible: cannot do better than keeping track of every state

Figure 6.2: Interpretation of coalescence-time histogram

to the burn-in period for Gibbs sampling (section 8.1).

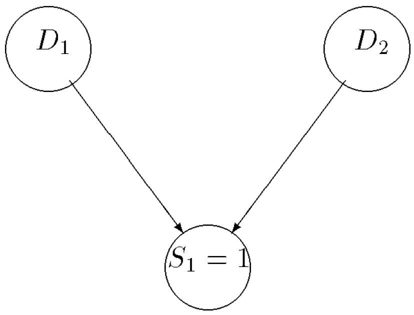
6.3 Perfectly summarized networks

The transitions of the chains being coupled from the past can be perfectly summarized in certain types of networks. A trivial example is a one-disease network with one or more symptoms that have evidence. When there is just one ? variable in the summary state, it is an exact representation of a set of two states.

Any network with two unknowns can also be perfectly summarized. Although it is possible that the state of the summary chain does not exactly represent the states of the set of chains, the variables of the summary chain always correctly summarize the variables of the set of states. Therefore, when the set of chains finally coalesces, the summary chain does represent the state of the coalesced chain.

When there are two unknowns, the first state of the summary chain has two ? variables, which correctly summarizes each variable. As long as the prior state of a summary chain transition correctly summarizes each variable of the set of states, and the transition correctly sets the variable to be changed, then the resulting state of the transition will correctly summarize each variable. The maximum and minimum conditional probabilities calculated for the summary chain transition are correct for the set of states, since they depend on a summary of sub-states with at most one ? variable, which is an exact representation of a set of sub-states. Therefore, the transition does correctly set the variable.

An example two-disease network is shown in figure 6.3. The Markov chains to be coupled have $2^2 = 4$ states of D_1, D_2, enumerated as $(00, 01, 10, 11)$. A summary chain has $3^2 = 9$ states of D_1, D_2, enumerated as $(00, 0?, 01, ?0, ??, ?1, 10, 1?, 11)$. The non-zero eigenvalues of the transition matrices, with associated eigenvectors (transposed into column vectors) are shown in table 6.1.



Two diseases D_1 and D_2, apriori probability = 0.1, status unknown
One symptom S_1, apriori probability = 0, status $S_1 = 1$
noisy-or weights = 1

Figure 6.3: Two-disease network

eigenvalues/eigenvectors

states	1	.81
	−−	−−
00	0	0
01	0.47368	0.49809
10	0.47368	−0.44828
11	0.05263	−0.04981

(for chains to be coupled)

eigenvalues/eigenvectors

states	1	.81	.81
	−−	−−	−−
00	0	0	0
0?	0	0	0
01	0.47368	0.49809	−0.44178
?0	0	0	0
??	0	0	0.49391
?1	0	0	0.05488
10	0.47368	−0.44828	−0.09631
1?	0	0	0
11	0.05263	−0.04981	−0.01070

(for summary chain)

Table 6.1: Two-disease transition matrix eigenvectors and eigenvalues

					$-T_{ES}$				
	$-T$	1	2	4	8	16	32	64	
	64	0	0	0	0	0	0	1	1
	32	0	0	0	0	0	40		40
	16	0	0	0	0	158			158
$-T_?$	8	0	0	0	260				260
	4	0	0	263					263
	2	0	184						184
	1	94							94
		−−	−−	−−	−−	−−	−−	−−	−−
		94	184	263	260	158	40	1	1000

Figure 6.4: Two-disease results histogram

Besides the eigenvalue of 1, there is no other eigenvalue of the summary chain that is larger than 0.81, the magnitude of the second largest eigenvalue of the chain on the states of (0/1) variables. This suggests that the summary chain converges just as quickly. A histogram of experimental results (figure 6.4) shows that both methods perform identically, always converging from the same number of steps $-T$ in the past.

Simple networks such as these are not of great interest by themselves, but they may exist as a sub-networks within a larger network. In a noisy-or network, symptoms which are known to be absent are not able to convey any influence between the diseases that are potential causes. Sections of the network can effectively be independent with sub-states that make transitions through simulation in a way that is independent of other sections of the network. If a large network consists only of independent single disease sub-networks, exact sampling always converges at $t = 0$ from a starting step of $T = -1$. This condition is established at run-time, depending on how the symptoms are instantiated. A small example is given, with two positive symptoms and a negative symptom separating the diseases (figure 6.5).

The conditions that allow for quickly converging networks should be avoided for test-

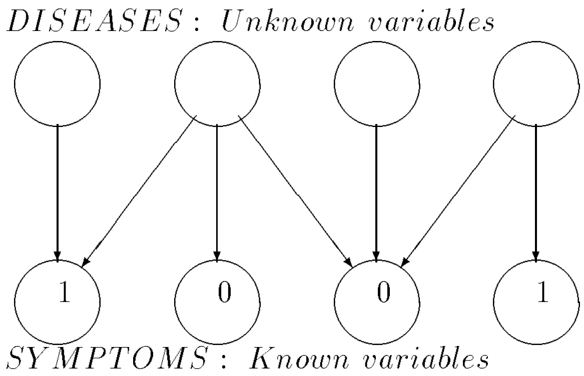


Figure 6.5: Perfectly summarized network containing independent sub-networks

ing purposes, since the objective is to look for the problem areas. Therefore, the analysis above suggests that further tests be done with networks that have sufficient number of positive symptoms and edges to suppress the occurrences of independent sections of the network.

6.4 Imperfectly summarized networks

6.4.1 Moderate example

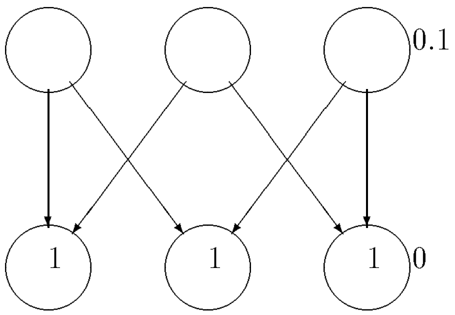
In general, network transitions cannot be perfectly summarized by chains of states of (0/?/1) variables. The example of figure 6.6 has moderately worse convergence for the summary chain.

The eigenvalues (non-zero) and associated eigenvectors (transposed into column vectors) of the summary chain transition matrix are shown in (table 6.2). Some of these are the same as for the transition matrix of the chains to be coupled. The additional eigenvector which specifies non-zero probabilities for states with ?-valued variables has a magnitude greater than all the other secondary eigenvalues. Its magnitude of .97 compared to $\sqrt{0.85050^2 + 0.07516^2} \approx 0.85$ reflects moderately worse convergence.

An experimental run shows that the summarizing method is required to go back on

	eigenvalues/eigenvectors			
	1	0.97275	$0.85050 + 0.07516i$	$0.85050 - 0.07516i$
states	−−	−−	−−	−−
000	0	0	0	0
00?	0	0	0	0
001	0	0	0	0
0?0	0	0	0	0
0??	0	0	0	0
0?1	0	0	0	0
010	0	0	0	0
01?	0	0	0	0
011	0.32143	−0.19274	$0.33826 + 0i$	$0.33826 + 0i$
?00	0	0	0	0
?0?	0	0	0	0
?01	0	0	0	0
??0	0	0	0	0
???	0	0.44104	0	0
??1	0	0.04900	0	0
?10	0	0	0	0
?1?	0	0.04900	0	0
?11	0	0.00544	0	0
100	0	0	0	0
10?	0	0	0	0
101	0.32143	−0.19097	$-0.15222 - 0.28250i$	$-0.15222 + 0.28250i$
1?0	0	0	0	0
1??	0	0.04451	0	0
1?1	0	0.00495	0	0
110	0.32143	−0.18922	$-0.16744 + 0.25425i$	$-0.16744 - 0.25425i$
11?	0	0	0	0
111	0.03571	−0.02102	$-0.01860 + 0.02825i$	$-0.01860 - 0.02825i$

Table 6.2: Imperfectly summarized network eigenvectors and eigenvalues, moderate example



Disease apriori probability	0.1000
Symptom apriori probability	0.0000
Noisy − or weight	1.0000

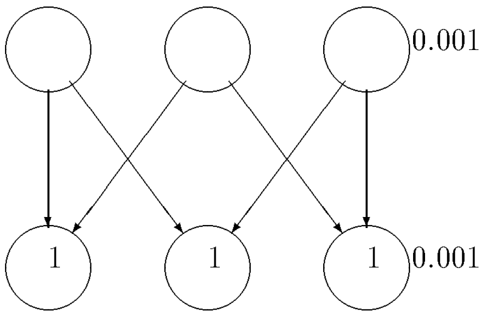
Figure 6.6: Imperfectly summarized network, moderate example

Histogram of $(-T_{ES}, -T_?)$

		$-T_{ES}$							
	$-T$	1	2	4	8	16	32	64	
	512	0	0	0	0	0	1	0	1
	256	0	0	1	0	11	12	4	28
	128	0	1	3	28	53	46	11	142
	64	0	1	11	57	111	61	21	262
	32	0	3	11	56	72	99		241
$-T_?$	16	0	0	9	34	118			161
	8	0	0	6	85				91
	4	0	2	35					37
	2	0	30						30
	1	7							7
		−−	−−	−−	−−	−−	−−	−−	−−
		7	37	76	260	365	219	36	1000

$$Average(-T_{ES}, -T_?) = (17.6, 53.9),\ \frac{Average\ T_?}{Average\ T_{ES}} = 3.06$$

Table 6.3: Imperfectly summarized network, moderate example results histogram



Disease apriori probability	0.0010
Symptom apriori probability	0.0010
Noisy − or weight	1.0000

eigenvalue	eigenvector
1.00000	invariant distribution
0.99601	component of distribution of states with ?-valued variables
$0.31138 + 0.16522i$	component of distribution of states with (0/1)-valued variables before convergence
$0.31138 - 0.16522i$	component of distribution of states with (0/1)-valued variables before convergence

Figure 6.7: Imperfectly summarized network, extreme example

average 53.9 time steps in the past, compared to only 17.6 time steps for the method of keeping track of every state (table 6.3).

6.4.2 Extreme example

If the probabilities in the imperfectly summarized network are more extreme, the method of the chain of summarized states does much worse than keeping track of every state. A problem that couples well when keeping track of every state, and hence converges easily for Gibbs sampling, can couple very badly when summarizing the states (figure 6.7).

The extreme value of the eigenvalue (.996) reflects the very poor convergence of the summary chain, compared to the convergence of the chains to be coupled (eigenvalue

Histogram of $(-T_{ES}, -T_?)$

		$-T_{ES}$					
	$-T$	1	2	4	8	16	
	2048	0	9	5	2	0	16
	1024	0	45	51	10	0	106
	512	0	120	98	19	0	237
	256	0	111	97	27	2	237
	128	0	101	67	18	1	187
	64	0	54	41	14	1	110
$-T_?$	32	0	32	22	4	0	58
	16	0	13	12	1	0	26
	8	0	8	6	2		16
	4	0	1	3			4
	2	0	2				2
	1	1					1
		−−	−−	−−	−−	−−	−−
		1	496	402	97	4	1000

$$Average(-T_{ES}, -T_?) = (3.44, 357),\ \frac{Average\ T_?}{Average\ T_{ES}} = 104$$

Table 6.4: Imperfectly summarized network, extreme example results histogram

$= \sqrt{0.31138^2 + 0.16522^2} \approx 0.35$). This difference is reflected in the test results (table 6.4), where many of the entries of the histogram are far above the diagonal. The configuration of this network helps the set of chains to coalesce in much fewer time steps than does the summary chain, by having a non-zero symptom apriori probability that allows for other explanations of the evidence besides the diseases of the network. That feature does not help the summary chain to reduce to a single chain faster, because the effect is lost when approximating the states. It is evident by the testing (chapter 7) that special kinds of networks like this do not occur very often when randomly generated.

Chapter 7

Test Cases

The objective of testing is to compare the coalescence time of the summary chain method with keeping track of every chain. A variety of network configurations are needed to make the testing as thorough as possible, within the realm of two-level medical diagnosis networks. It is convenient to search the space of possible networks, with constraints as parameters, by randomly generating the networks. Then on each network generated, a number of exact sampling runs are performed to collect statistics.

7.1 Random network generation

Randomly generated networks conform to certain criteria (table 7.1). A network configuration is randomly generated according to input parameters (table 7.2) which specify the attributes. Most of these are self-explanatory, except for the density of edges. When the network is built, edges are randomly added until the network is minimally connected. The density of edges specifies the proportion of the remaining edges to randomly add to the network. Then a test case is generated by simulating a patient entering the clinic with one or more symptoms (table 7.3).

A belief network is tested by performing a number of runs for one particular set of symptoms, or test case. Each run generates the summary chain and the complete set

- The network should have at least one symptom positive to reflect a patient visiting a clinic with a complaint.
- The status of diseases is unknown.
- Disconnected networks are not considered because the convergence behaviour can be observed in smaller fully connected networks.
- The number of unknown variables is limited to a reasonable number n because the experiment of tracking every possible state must cope with 2^n states

Table 7.1: Criteria for Random Network Generation

Number of diseases
Number of symptoms
Disease apriori probability
Symptom probability, no disease
Noisy-or weight
Density of edges

Table 7.2: Belief Network attributes

- Stochastically instantiate the state of the diseases according to their apriori probabilities.
- Given the disease state, stochastically set the given-values of the symptoms (the evidence) according to their noisy-or probabilities.
- Throw away any memory of the disease state.
- Repeat the above if necessary until at least one symptom is positive.

Table 7.3: Generating a Test Case

of chains to be coupled, by using a new set of pseudo-random numbers. A comparison statistic, $(-T_{ES}, -T_?)$, is plotted on a 2-dimensional histogram. It records the starting times in the past necessary to coalesce by $t = 0$ for the set of chains (T_{ES}) and the summary chain ($T_?$). Each network and its test case is characterized by an average vector $(-T_{ES}, -T_?)$, with components average T_{ES} and average $T_?$.

The random network generator produces and tests a batch of belief networks. At the end of the batch test, there is a batch summary that includes a histogram with plots of the comparison statistic, $(-T_{ES}, -T_?)$ for every run of every network in the batch. As well, networks are listed in order of the sum of squares magnitude of the average $(-T_{ES}, -T_?)$ vector, to identify extreme cases.

7.2 Tests on plausible networks

Parameters can be supplied to the random network generator that are plausible for a medical diagnosis application (table 7.4). The test results for networks generated with the plausible parameters show very good behaviour for the summary chain method (table 7.5). A few individual cases show slightly worse results than the rest. It is difficult looking at the individual examples to determine exactly why they are worse. A cursory observation shows that the well behaved networks often had only one or two positive symptoms. The worst behaving networks seemed to have more than a fair share of low probability diseases, as well as several positive symptoms.

Nevertheless, the results are too good to be interesting. If the network was to be expanded into a realistic network of a few hundred variables, the results may be less favourable, but for large networks it is not possible to compare with the method of keeping track of every chain. Also, we do not know for sure what networks will be used in practice.

Number of diseases		10
Number of symptoms		10
Number of DAGs		100
Samples per DAG		100
Total number of samples		10000
Disease apriori probability	1.	0.0100 to 0.0500
Symptom probability, no disease	2.	0.0000 to 0.0500
Noisy-or weight	3.	0.0100 to 1.0000
Density of edges	4.	0.2000

1. Typical diseases should vary from rare (close to 0) to common (0.05).

2. Symptoms may have causes not explained by the network. The more comprehensive is the network the lower this probability should be. For this test, it is assumed the network is fairly reliable in this respect, and no more than 5 percent of the causes of a symptom cannot be explained by the network.

3. The effect of a disease on a symptom may vary from weak to strong. There is no reason this should not vary over all the possible range, (except 0 for this would delete the edge).

4. The density of the edges will be low if the weight is 0 on many of the edges between diseases and symptoms. It seems reasonable that some of them will be zero, so after the network is minimally connected, only .2 of the remaining edges to add have non-zero probability.

(Justification for parameter choices)

Table 7.4: Plausible medical diagnosis network parameters

Histogram of $(-T_{ES}, -T_?)$ *for batch of DAGs*

		$-T_{ES}$									
	$-T$	1	2	4	8	16	32	64	128	256	
	512	0	2	1	3	2	4	2	0	0	14
	256	0	1	13	19	17	10	8	6	3	77
	128	0	3	30	42	32	17	13	13		150
	64	0	7	74	83	52	34	32			282
	32	0	10	100	113	72	86				381
$-T_?$	16	0	26	136	123	169					454
	8	0	30	181	342						553
	4	0	110	998							1108
	2	0	1880								1880
	1	5101									5101
		--	--	--	--	--	--	--	--	--	--
		5101	2069	1533	725	344	151	55	19	3	10000

$$Average(-T_{ES}, -T_?) = (3.82, 10.1), \quad \frac{Average\ T_?}{Average\ T_{ES}} = 2.65$$

Average $(-T_{ES}, -T_?)$	number of positive symptoms	number of diseases with prob $< .01$
(20.6,135)	6	3
(52.4,115)	3	3
(13.6,110)	6	1
(5.18,83.1)	5	1
(8.52,65)	3	2

(Five worst performing networks)

Table 7.5: Plausible medical diagnosis network test summary

7.3 Tests on more difficult networks

Criteria for choosing the parameters of belief network generation that are thought to produce adverse results while keeping the size of the network small are shown in table 7.6. These come about from the work of section 7.2 (Plausible networks) and chapter 6 (Coalescence time), plus additional observation of a network configuration known to cause poor Gibbs sampling convergence.

Poor Gibbs sampling convergence occurs when there are areas of the state space that have low probabilities of transitions to other areas of the state space. An simple example is shown in figure 7.1. In this network, there can be no other cause for the symptom besides the two diseases. Therefore, Gibbs sampling certainly instantiates one of the diseases to positive. Then there is not much reason to instantiate the other disease, since its apriori probability is very low. Gibbs sampling will probably not change the state for quite a while, until by a low probability both diseases are instantiated to positive. Then the first disease will have a chance to be negative. Running Gibbs sampling for a short time will produce results that strongly favour one disease over the other, when in fact they are equally probable. It is necessary to sample for a long time to converge to the correct distribution.

Test runs are included for five classes of networks. Each class includes 100 randomly generated networks, each having 100 samples obtained through exact sampling. These are designed to produce bad convergence, by using the aforementioned criteria to vary the parameters of network synthesis (table 7.7).

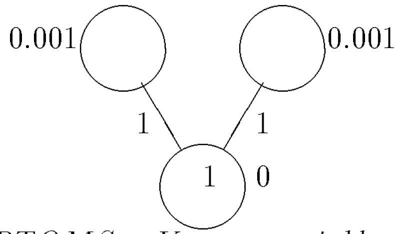
Detailed summaries of the tests for each class of network are contained in the appendix. There are 10000 exact sampling runs represented in the histogram for each class of networks, compiled from 100 runs each of 100 networks. Each network has a constant set of symptoms for each of the 100 runs.

The characteristic ratio $\frac{Average\ T_?}{Average\ T_{ES}}$ indicates the how much a coupling run on the summary chain is slowed down by the introduction of spurious chains (section 5.2). For

- more extreme (near zero) probabilities for diseases
- high density of edges to connect multiple diseases to each symptom and to activate more symptoms
- small (or zero) probability of symptom, given no diseases
- weight of edges anywhere between 0 and 1

Table 7.6: Criteria for generating poorly performing test networks

$DISEASES: \ Unknown\ variables$



$SYMPTOMS: \ Known\ variables$

Disease apriori probability	0.0010
Symptom probability, no disease	0.0000
Noisy-or weight	1.0000

$Histogram\ of\ (-T_{ES}, -T_?)$

$-T_{ES}$

	$-T$	4	8	16	32	64	128	256	512	1024	2048	4096	
	4096	0	0	0	0	0	0	0	0	0	0	2	2
	2048	0	0	0	0	0	0	0	0	0	8		8
	1024	0	0	0	0	0	0	0	0	28			28
	512	0	0	0	0	0	0	0	25				25
	256	0	0	0	0	0	0	17					17
$-T_?$	128	0	0	0	0	0	9						9
	64	0	0	0	0	7							7
	32	0	0	0	1								1
	16	0	0	0									0
	8	0	2										2
	4	1											1
		−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−	−−
		1	2	0	1	7	9	17	25	28	8	2	100

$Average(-T_{ES}, -T_?) = (721, 721), \ \frac{Average\ T_?}{Average\ T_{ES}} = 1$

Figure 7.1: Example of a network with poor Gibbs sampling convergence

Keys to Class heading:

Disease apriori probability	(1)	0 - 0.01, all cases
Symptom probability, no disease	(2)	
Noisy-or weight	(3)	
Density of edges	(4)	

Class	(2)	(3)	(4)	Average $(-T_{ES}, -T_?)$	$\frac{Average\ T_?}{Average\ T_{ES}}$	Average $\frac{Average\ T_{ES}}{Median\ T_{ES}}$
1. high density edges	0 - 0.01	0.01 - 1	1.00	(124,501)	4.04	1.22
2. medium density edges	0 - 0.01	0.01 - 1	0.50	(35.7,402)	11.3	1.22
3. low density edges	0 - 0.01	0.01 - 1	0.00	(4.06,29.9)	7.37	1.14
4. stronger weights	0 - 0.01	0.50 - 1	1.00	(1280,2520)	1.97	1.25
5. no outside causes of symptom	0	0.01 - 1	1.00	(686,1460)	2.13	1.39

Table 7.7: Results summarized by class of network

almost all of the networks tested, the ratio is moderate, even for the networks with the longest coalescence times. The exceptions are two extreme cases out of a total of 500 networks tested, which far exceed all other cases. One case had an average $(-T_{ES}, -T_?)$ = (51.8,19900). It was the only network that simulated a patient coming into the clinic with two rare diseases. As a result, the number of instantiated symptoms is higher for this case (8 out of 10 positive) than for the others, causing more widespread interdependencies among the variables of the network. The other extreme case had an average $(-T_{ES}, -T_?)$ = (4.56,711). It had one rare disease, but 6 out of 10 symptoms were positive.

Over all the classes of networks, about half of the networks tested have a characteristic ratio of one, indicating perfectly summarized chains. An exception is class number five, for which few of the networks have ratios that are equal to one, although the characteristic ratio for that class is still low. The classes with the lowest characteristic ratio have the least variation of the ratio for the networks in the class, making the class ratio a better predictor of the performance of the summary chain method for the networks of the class.

The problems that have the slowest coalescence rates (highest $-T_{ES}$) have the lowest $\frac{T_?}{T_{ES}}$ ratio, as evidenced in the average $(-T_{ES}, -T_?)$ for certain classes of networks (see table 7.7). As mentioned, that ratio is most reliable for these classes, giving the most confidence about the efficient performance of exact sampling on networks in the class. Having a slower coalescence rate means that Gibbs sampling probably (not guaranteed) converges poorly. This is noteworthy because it implies that the exact sampling works better where it is most needed, by eliminating error where error is the most troublesome.

Chapter 8

Conclusions

There is evidence that exact sampling using the summarized set of states may be a valuable alternative to discarding a burn-in period in Gibbs sampling, especially in difficult cases where Gibbs sampling converges poorly. Excessive Gibbs sampling burn-in times employed by users who are uncertain about the convergence characteristics of the problem strengthen the case for exact sampling. However, the inability to say that Gibbs sampling cannot perform well when exact sampling performs poorly weakens the case for exact sampling. The always present benefit of exact sampling is the removal of uncertainty about the systematic error, since it is always zero. (As in any Monte Carlo problem, sampling error is present, but its magnitude can easily be assessed.)

8.1 Comparison to Gibbs Sampling

The test results give averages for various classes of synthetic networks produced by varying the parameters of synthesis. The averages are T_{ES}, the time in the past needed to couple the chains keeping track of every state, and $T_?$, the time needed to reduce the approximating chain to a mapping to one chain. As well, the ratio of these averages $\frac{Average\ T_?}{Average\ T_{ES}}$ is obtained. For each network, a median of T_{ES} is determined empirically. Over all the networks in a class, the average of the $\frac{Average\ T_{ES}}{Median\ T_{ES}}$ ratio is calculated.

The median coalescence time can be used to arrive at an upper bound on the Gibbs sampling error. This provides a guarantee, that if Gibbs sampling is burned in for a given length of time, then an error tolerance is satisfied. The amount of computational effort to guarantee Gibbs sampling can then be compared to the effort of coupling from the past, for various error tolerances. Thus a quantitative evaluation of coupling from the past in terms of the alternative, ordinary Gibbs sampling, can be arrived at. The median coalescence time is estimated by a rule of thumb ratio of the average $T_?$, using the above empirically determined ratios, which is meant to be universally applied to any of the medical diagnosis networks studied here.

Median coalescence time: Let T_{med} be the median coalescence time for the method that tracks every chain. If T_* is a random variable which is the coalescence time of a coupling simulation, then $Prob(T_* > T_{med}) \leq \frac{1}{2}$. This is the probability of the chains failing to coalesce after T_{med} time steps. If the chains are coupled starting from $t = 0$, the tail probability is the same, due to the chains being governed by the same probability distribution as in the case when they are started in the past.

To determine the required burn-in time, suppose Gibbs sampling is started from an initial distribution μ_0, and burned-in for kT_{med} steps. We can measure the error by the total variation distance between the distribution of the chain and the invariant distribution. For a finite state space χ, this is

$$||\mu_{kT_{med}} - \pi|| = \frac{1}{2}\sum_{x\in\chi} |\mu_{kT_{med}}(x) - \pi(x)|.$$

The error can be bounded as

$$error = ||\mu_{kT_{med}} - \pi|| \leq Prob(T_* > kT_{med}) \leq (\frac{1}{2})^k.$$

The left inequality indicates that the error is bounded by the tail probability of the coalescence time (Rosenthal 1995). To arrive at the bound for $Prob(T_* > kT_{med})$, consider running k couplings for T_{med} steps independently. The probability of all k of the couplings failing to coalesce is at most $(\frac{1}{2})^k$. If coupling is done for kT_{med} steps, then $Prob(T_* > kT_{med}) \leq (\frac{1}{2})^k$, and is likely to be less since the k couplings are not independent. Then, for example, an error tolerance of $.004 = (\frac{1}{2})^8$ is guaranteed by a burn-in time of $8 \times T_{med}$. In general, if the median coalescence time can be determined, and an error tolerance of ϵ is required, then a burn-in time that is guaranteed to satisfy the error tolerance is

$$burn\ in = -log_2\epsilon \times T_{med}.$$

For a particular network, the coalescence time can be estimated from the coalescence time found empirically by the search strategy, which overshoots by an expected factor of 1.44 (section 4.1). The expected median coalescence time is $T_{med} = \frac{Median\ -T_{ES}}{1.44}$. The expected average coalescence time is $\frac{Average\ -T_{ES}}{1.44}$.

The median coalescence time over a class of networks can be estimated from the average coalescence time over the class of networks. Empirically, table 7.7 shows that the average of the ratio $\frac{Average\ T_{ES}}{Median\ T_{ES}}$ varies from 1.14 to 1.39, depending on the class of the network. On individual networks, the ratio varies from slightly less then 1 to slightly more than 2. A rule of thumb might be stated as $Median\ T_{ES} \approx \frac{3}{4} \times Average\ T_{ES}$. Then a reasonable estimate of the median coalescence time for a class of networks is $T_{med} \approx \frac{\frac{3}{4} \times Average\ -T_{ES}}{1.44} \approx \frac{Average\ -T_{ES}}{2}$.

This ratio is derived empirically for chains coupled in the past. Since the behaviour of the chain is the same whether started in the past or from $t = 0$, it can be considered valid for chains started from $t = 0$. A burn-in time that guarantees an error tolerance ϵ can then be stated as

$$burn\ in \approx -log_2\epsilon \times \frac{Average\ -T_{ES}}{2}.$$

The work of coupling from the past: A coupling from the past run using the summary chain method searches for a starting time in the past, $T_?$, which allows the chains represented by the summary chain to coalesce by time $t = 0$. The computational

1.	Transition probabilities calculated at each time step	2 transitions
2.	Searching for a starting time $T_?$ in the past	$(-T_? - 1)$ time steps
3.	Simulating summary chain from starting time $T_?$	$-T_?$ time steps
	Total amount of computational work	$(-4T_? - 2)$ transitions

Table 8.1: Computational work of the summary chain method

work required by the summary chain method is given in table 8.1. Each simulation time step requires doing the work of two Markov chain transitions for the chains that have the minimum and maximum transition probabilities in the set of chains represented by the summary chain. There is an additional $-T_? - 1$ simulation time steps required to search for a starting time in the past, for a total of $2T_? - 1$ time steps. Thus, there is a total of $4T_? - 2$ Markov chain transitions to be simulated. The average amount of computation for the summary chain method is about $4 \times (Average\ T_?)$ transitions.

The ratio $\frac{T_?}{T_{ES}}$ is a random variable characterizing each coupling run, that indicates how much longer it takes for the summary chain to coalesce compared to the set of chains it summarizes. A ratio greater than one indicates that coupling of the summary chain is slowed down the introduction of spurious chains (section 5.2). The characteristic ratio of a network or class of networks is the ratio of averages, $\frac{Average\ T_?}{Average\ T_{ES}}$. The characteristic ratio has been determined empirically for the various classes of networks tested (table 7.7).

When the number of Markov chain transitions for a burn-in period and for coupling from the past are equal, the cross-over point at which one method is more efficient than the other is reached. The factor relating the two quantities of work depends on the tolerance ϵ and the characteristic ratio $\frac{Average\ T_?}{Average\ T_{ES}}$ for the class of network of interest, as:

$$\begin{array}{c} burn\ in \\ time \end{array} \approx -log_2\epsilon \times \frac{Average\ \ - T_{ES}}{2} \approx \left(\frac{-log_2\epsilon}{8 \times \frac{Average\ T_?}{Average\ T_{ES}}} \right) \times \left(\begin{array}{c} number\ of \\ transitions \\ for\ summary \\ chain\ method \end{array} \right).$$

As the user's error tolerance decreases, the summary chain method becomes more favoured. The required burn-in effort is equal to the work of coupling from the past when

$$log_2\epsilon = -8\frac{Average\ T_?}{Average\ T_{ES}}.$$

A few examples in table form are:

$\frac{Average\ T_?}{Average\ T_{ES}}$	1	2	4	8	16
ϵ	2^{-8}	2^{-16}	2^{-32}	2^{-64}	2^{-128}

The accuracy obtained by doing the same amount of work on Gibbs sampling burn-in as on exact sampling is greater than needed in most cases. However, an experimenter unsure about how long to burn-in may easily do more than that amount of work. The characteristic ratio $\frac{Average\ T_?}{Average\ T_{ES}}$ was found to range around 2 or 4, without much variation, for networks that exhibit slower coalescence rates (table 7.7). Practically, the amount of work required for exact sampling for these types of network is not unreasonable. They could be considered as good candidates for exact sampling. Moreover, networks with slow coalescence likely (but not proven) have the most problem with systematic error in Gibbs sampling; exact sampling has no systematic error at all.

8.2 Qualitative considerations

Other considerations may broaden the interpretation of the quantitative analysis above, to cause a preference for exact sampling even if the estimate of ideal burn-in is less than the work of coupling from the past, or to take an apparently advantageous comparison with scepticism.

- The Gibbs sampling user is ignorant about the convergence behaviour. Conservative users tend to exceed the ideal burn-in time, but still fall short of it sometimes and receive a wrong answer without knowing it. This is because they have to guess how well the Markov chain converges. In practice the burn-in runs in use may exceed the ideal by many times.

- There is no lower bound on how fast Gibbs sampling can converge (see section 4.3.1) in terms of the coalescence rate. Therefore, it may be possible that the Markov chain could converge very rapidly when it does not coalesce quickly. However, there is no particular reason to believe that it should, and there are no examples known to show such behaviour. It is a matter of further research to try to specify a lower bound on the mixing rate, or to find examples where Gibbs sampling can work much better.

These conclusions are based on the results of testing on small synthetic networks using extreme values for parameters to cause poor convergence characteristics. Comparisons are made with exact sampling by tracking every chain, a strategy only possible with small networks. It would be helpful to try the summary chain method of exact sampling on real full-scale noisy-or belief networks and compare the performance with existing inference methods used in the field.

Chapter 9

Appendix

Random DAG Test Parameters			
Number of diseases	10		
Number of symptoms	10		
Number of DAGs	100		
Samples per DAG	100		
Total number of samples	10000		
Disease apriori probability	0.0000	to	0.0100
Symptom probability, no disease	0.0000	to	0.0100
Noisy-or weight	0.0100	to	1.0000
Density of edges	1.0000		

Comparison of chain lengths $(-T)$ to couple from the past.

Histogram of $(-T_{ES}, -T_?)$ *for batch of DAGs*

	$-T_{ES}$ →														
	$-T$	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	
$-T_?$	32768	0	0	0	0	0	0	0	2	0	4	1	0	0	7
	16384	0	0	0	0	0	0	0	0	8	4	4	0	1	17
	8192	0	0	0	0	0	4	4	4	17	30	21	13	4	97
	4096	0	0	2	2	4	11	11	23	51	47	52	38	12	253
	2048	0	3	7	16	19	29	49	63	84	113	97	56		536
	1024	0	3	20	35	67	65	81	123	125	122	164			805
	512	0	9	23	48	96	108	122	137	135	223				901
	256	0	5	30	62	89	113	100	90	226					715
	128	0	3	21	55	76	88	68	217						528
	64	0	0	17	32	48	48	163							308
	32	0	0	6	21	34	123								184
	16	0	0	5	19	73									97
	8	0	0	5	56										61
	4	0	1	93											94
	2	0	155												155
	1	5242													5242
		--	--	--	--	--	--	--	--	--	--	--	--	--	--
		5242	179	229	346	506	589	598	659	646	543	339	107	17	10000

$$Average(-T_{ES}, -T_?) = (124, 501),\ \frac{Average\ T_?}{Average\ T_{ES}} = 4.04$$

List of DAGs ordered from slowest coalescence to fastest:

DAGNumber	Average$(-T_{ES}, -T_?)$	$\frac{Average\ T_?}{Average\ T_{ES}}$	$\frac{Average\ T_{ES}}{Median\ T_{ES}}$
1	(525,7.13e+003)	13.6	1.02
53	(850,4.29e+003)	5.05	1.66
79	(879,2.56e+003)	2.91	1.72
50	(974,2.23e+003)	2.29	1.9
26	(626,2.19e+003)	3.49	1.22
58	(896,2.02e+003)	2.25	1.75
48	(527,2.09e+003)	3.96	2.06

84	(480,1.82e+003)	3.79	0.938
7	(774,1.61e+003)	2.08	1.51
5	(206,1.67e+003)	8.11	1.61
93	(372,1.46e+003)	3.93	1.45
8	(646,1.32e+003)	2.04	1.26
31	(381,1.38e+003)	3.63	1.49
96	(362,1.39e+003)	3.83	1.41
90	(408,1.23e+003)	3.02	1.59
12	(290,1.02e+003)	3.5	1.13
47	(245,829)	3.39	1.91
37	(233,802)	3.44	1.82
92	(195,786)	4.03	1.52
43	(293,693)	2.36	1.15
9	(223,714)	3.2	1.74
32	(110,738)	6.7	1.15
97	(20.2,716)	35.4	1.26
55	(93.6,668)	7.14	1.46
62	(68.8,638)	9.27	1.43
66	(263,509)	1.93	2.06
33	(137,549)	4.01	1.07
52	(171,524)	3.07	1.33
98	(24.3,541)	22.3	1.52
4	(116,484)	4.19	1.81
74	(80.4,488)	6.08	1.26
20	(192,439)	2.28	1.5
30	(54.4,473)	8.7	1.7
13	(29.7,452)	15.2	1.86
35	(199,398)	2	1.55
64	(87,419)	4.81	1.36
44	(17.9,419)	23.4	1.12
3	(37,381)	10.3	1.16
49	(25.5,368)	14.4	1.6
94	(26.5,343)	13	1.66
57	(44.3,330)	7.47	1.38
28	(70.5,310)	4.4	1.1
19	(40.9,218)	5.33	1.28
56	(13.9,188)	13.6	0.868
41	(13.7,182)	13.3	1.71
59	(4.88,6.34)	1.3	1.22
72	(2.18,2.18)	1	1.09
65	(2.09,2.09)	1	1.04
70	(1.06,1.06)	1	1.06
54	(1.04,1.04)	1	1.04
14	(1.03,1.03)	1	1.03
17	(1.03,1.03)	1	1.03

46	(1.03,1.03)	1	1.03
91	(1.03,1.03)	1	1.03
11	(1.02,1.02)	1	1.02
2	(1.01,1.01)	1	1.01
21	(1.01,1.01)	1	1.01
29	(1.01,1.01)	1	1.01
36	(1.01,1.01)	1	1.01
42	(1.01,1.01)	1	1.01
95	(1.01,1.01)	1	1.01
6	(1,1)	1	1
10	(1,1)	1	1
15	(1,1)	1	1
16	(1,1)	1	1
18	(1,1)	1	1
22	(1,1)	1	1
23	(1,1)	1	1
24	(1,1)	1	1
25	(1,1)	1	1
27	(1,1)	1	1
34	(1,1)	1	1
38	(1,1)	1	1
39	(1,1)	1	1
40	(1,1)	1	1
45	(1,1)	1	1
51	(1,1)	1	1
60	(1,1)	1	1
61	(1,1)	1	1
63	(1,1)	1	1
67	(1,1)	1	1
68	(1,1)	1	1
69	(1,1)	1	1
71	(1,1)	1	1
73	(1,1)	1	1
75	(1,1)	1	1
76	(1,1)	1	1
77	(1,1)	1	1
78	(1,1)	1	1
80	(1,1)	1	1
81	(1,1)	1	1
82	(1,1)	1	1
83	(1,1)	1	1
85	(1,1)	1	1
86	(1,1)	1	1
87	(1,1)	1	1
88	(1,1)	1	1

89	(1,1)	1	1
99	(1,1)	1	1
100	(1,1)	1	1

Average of $\frac{Average\ T_{ES}}{Median\ T_{ES}} = 1.22$

Random DAG Test Parameters			
Number of diseases	10		
Number of symptoms	10		
Number of DAGs	100		
Samples per DAG	100		
Total number of samples	10000		
Disease apriori probability	0.0000	to	0.0100
Symptom probability, no disease	0.0000	to	0.0100
Noisy-or weight	0.0100	to	1.0000
Density of edges	0.5000		

Comparison of chain lengths ($-T$) to couple from the past.

Histogram of $(-T_{ES}, -T_?)$ *for batch of DAGs*

		$-T_{ES}$												
	$-T$	1	2	4	8	16	32	64	128	256	512	1024	2048	
	65536	0	0	0	0	2	2	4	3	1	0	0	0	12
	32768	0	0	0	2	1	9	6	1	0	0	0	0	19
	16384	0	0	0	0	0	6	11	2	0	0	0	0	19
	8192	0	0	0	1	6	9	9	3	1	0	0	0	29
	4096	0	0	1	2	7	14	13	7	9	5	1	1	60
	2048	0	0	14	17	27	29	36	23	26	11	5	5	193
	1024	0	3	38	65	79	89	84	57	49	26	44		534
	512	0	7	98	128	156	167	134	103	52	62			907
$-T_?$	256	0	16	98	145	134	150	146	118	143				950
	128	0	17	100	132	130	137	130	180					826
	64	0	22	76	108	106	103	186						601
	32	0	21	61	67	68	156							373
	16	0	14	41	53	125								233
	8	0	10	27	117									154
	4	0	13	99										112
	2	0	436											436
	1	4542												4542
		--	--	--	--	--	--	--	--	--	--	--	--	--
		4542	559	653	837	841	871	759	497	281	104	50	6	10000

$$Average(-T_{ES}, -T_?) = (35.7, 402),\ \frac{Average\ T_?}{Average\ T_{ES}} = 11.3$$

List of DAGs ordered from slowest coalescence to fastest:

DAGNumber	Average($-T_{ES}, -T_?$)	$\frac{Average\ T_?}{Average\ T_{ES}}$	$\frac{Average\ T_{ES}}{Median\ T_{ES}}$
68	(51.8,1.99e+004)	384	1.62
43	(351,1.48e+003)	4.22	1.37
24	(137,1.45e+003)	10.6	1.07
25	(151,1.1e+003)	7.29	1.18
39	(29.2,1.02e+003)	34.9	0.912
64	(35.3,918)	26	1.1

78	(112,691)	6.19	1.75
37	(476,503)	1.06	1.86
6	(386,568)	1.47	1.51
23	(26.6,650)	24.4	1.11
53	(45.4,594)	13.1	1.42
20	(6.18,589)	95.3	1.54
46	(81.1,491)	6.06	1.27
62	(63.4,476)	7.51	0.99
97	(18.2,462)	25.4	1.14
8	(26.8,454)	17	1.67
89	(10.2,452)	44.4	1.27
15	(9.12,451)	49.4	1.14
58	(124,431)	3.49	1.93
56	(124,402)	3.24	0.97
98	(187,374)	2	1.46
91	(34.6,396)	11.4	1.44
38	(78.7,380)	4.83	1.23
71	(28.4,372)	13.1	1.78
84	(8.16,350)	42.9	1.02
48	(46.6,324)	6.96	1.46
10	(22.8,310)	13.6	1.43
33	(63.8,300)	4.71	0.997
19	(6.46,293)	45.4	0.808
81	(82,276)	3.36	1.28
92	(41.1,280)	6.82	1.28
60	(5.62,283)	50.3	1.41
69	(76.6,270)	3.52	1.2
18	(104,258)	2.47	1.63
99	(24,267)	11.1	1.5
63	(39.9,256)	6.41	1.25
27	(38.3,251)	6.56	1.2
72	(82,238)	2.9	1.28
51	(42.9,232)	5.42	1.34
5	(75.7,223)	2.95	1.18
96	(6.89,225)	32.6	1.72
29	(6.48,166)	25.7	1.62
45	(20.8,164)	7.91	1.3
82	(64.2,149)	2.32	2.01
26	(17.8,145)	8.17	1.11
61	(15.3,89.4)	5.83	1.92
86	(3.1,71.9)	23.2	0.775
83	(14.5,63.3)	4.37	1.81
42	(10.3,32.1)	3.13	1.28
44	(4.91,21.8)	4.45	1.23
9	(2.69,2.87)	1.07	1.35

50	(1.88,1.88)	1	0.94
3	(1.68,1.84)	1.1	1.68
52	(1.67,1.79)	1.07	1.67
40	(1.34,1.34)	1	1.34
55	(1.28,1.28)	1	1.28
47	(1.2,1.2)	1	1.2
85	(1.18,1.18)	1	1.18
59	(1.16,1.16)	1	1.16
70	(1.14,1.14)	1	1.14
36	(1.13,1.13)	1	1.13
1	(1.12,1.12)	1	1.12
57	(1.09,1.09)	1	1.09
75	(1.09,1.09)	1	1.09
14	(1.07,1.07)	1	1.07
28	(1.06,1.06)	1	1.06
74	(1.06,1.06)	1	1.06
2	(1.05,1.05)	1	1.05
49	(1.05,1.05)	1	1.05
13	(1.04,1.04)	1	1.04
21	(1.04,1.04)	1	1.04
32	(1.03,1.03)	1	1.03
41	(1.03,1.03)	1	1.03
79	(1.03,1.03)	1	1.03
16	(1.02,1.02)	1	1.02
54	(1.02,1.02)	1	1.02
66	(1.02,1.02)	1	1.02
77	(1.02,1.02)	1	1.02
88	(1.02,1.02)	1	1.02
93	(1.02,1.02)	1	1.02
95	(1.02,1.02)	1	1.02
4	(1.01,1.01)	1	1.01
12	(1.01,1.01)	1	1.01
31	(1.01,1.01)	1	1.01
65	(1.01,1.01)	1	1.01
90	(1.01,1.01)	1	1.01
7	(1,1)	1	1
11	(1,1)	1	1
17	(1,1)	1	1
22	(1,1)	1	1
30	(1,1)	1	1
34	(1,1)	1	1
35	(1,1)	1	1
67	(1,1)	1	1
73	(1,1)	1	1
76	(1,1)	1	1

80	(1,1)	1	1
87	(1,1)	1	1
94	(1,1)	1	1
100	(1,1)	1	1

Average of $\frac{Average\ T_{ES}}{Median\ T_{ES}} = 1.22$

Random DAG Test Parameters			
Number of diseases	10		
Number of symptoms	10		
Number of DAGs	100		
Samples per DAG	100		
Total number of samples	10000		
Disease apriori probability	0.0000	to	0.0100
Symptom probability, no disease	0.0000	to	0.0100
Noisy-or weight	0.0100	to	1.0000
Density of edges	0.0000		

Comparison of chain lengths ($-T$) to couple from the past.

Histogram of $(-T_{ES}, -T_?)$ *for batch of DAGs*

	$-T_{ES}$											
	$-T$	1	2	4	8	16	32	64	128	256	512	
$-T_?$	8192	0	0	0	0	0	0	1	0	0	0	1
	4096	0	0	0	0	1	0	0	2	0	1	4
	2048	0	3	12	3	3	4	3	6	2	1	37
	1024	0	3	20	5	4	4	8	8	2	1	55
	512	0	6	27	21	14	10	19	13	3	4	117
	256	0	8	39	22	7	17	16	8	5		122
	128	0	8	61	33	17	24	9	13			165
	64	0	5	47	30	11	12	21				126
	32	0	5	54	22	14	42					137
	16	0	11	76	45	74						206
	8	0	28	108	169							305
	4	0	39	493								532
	2	0	1663									1663
	1	6530										6530
		--	--	--	--	--	--	--	--	--	--	--
		6530	1779	937	350	145	113	77	50	12	7	10000

$$Average(-T_{ES}, -T_?) = (4.06, 29.9),\ \frac{Average\ T_?}{Average\ T_{ES}} = 7.37$$

List of DAGs ordered from slowest coalescence to fastest:

DAGNumber	Average$(-T_{ES}, -T_?)$	$\frac{Average\ T_?}{Average\ T_{ES}}$	$\frac{Average\ T_{ES}}{Median\ T_{ES}}$
97	(124,908)	7.34	1.29
99	(4.56,711)	156	1.14
85	(54.1,436)	8.05	1.69
64	(5.49,236)	43	1.37
57	(6.04,208)	34.4	1.51
75	(33.6,200)	5.95	1.05
67	(5.51,51.9)	9.41	1.38
19	(4.92,45.6)	9.27	1.23
9	(17,23.7)	1.39	1.06

29	(12.7,14.7)	1.16	1.58
15	(4.2,10.3)	2.46	1.05
1	(6.12,8.42)	1.38	1.53
90	(5.36,7.04)	1.31	1.34
61	(4.31,6.67)	1.55	1.08
83	(3.12,7.2)	2.31	0.78
10	(3.92,4.12)	1.05	0.98
24	(3.28,3.72)	1.13	1.09
66	(2.57,3.99)	1.55	1.28
88	(2.95,3.03)	1.03	1.48
30	(2.58,2.62)	1.02	1.29
44	(2.3,2.3)	1	1.15
80	(2.25,2.29)	1.02	1.13
70	(2.18,2.34)	1.07	1.09
48	(2.01,2.01)	1	1
22	(1.95,2.01)	1.03	0.975
91	(1.93,1.93)	1	0.965
26	(1.87,1.89)	1.01	0.935
100	(1.85,1.85)	1	0.925
12	(1.79,1.79)	1	0.895
69	(1.75,1.75)	1	0.875
43	(1.74,1.74)	1	0.87
79	(1.67,1.67)	1	1.67
72	(1.55,1.55)	1	1.55
31	(1.53,1.53)	1	1.53
33	(1.44,1.44)	1	1.44
49	(1.38,1.38)	1	1.38
5	(1.37,1.37)	1	1.37
3	(1.36,1.36)	1	1.36
68	(1.36,1.36)	1	1.36
16	(1.32,1.32)	1	1.32
93	(1.32,1.32)	1	1.32
34	(1.31,1.31)	1	1.31
62	(1.27,1.27)	1	1.27
27	(1.26,1.26)	1	1.26
46	(1.24,1.24)	1	1.24
47	(1.24,1.24)	1	1.24
86	(1.22,1.22)	1	1.22
35	(1.2,1.2)	1	1.2
76	(1.2,1.2)	1	1.2
6	(1.18,1.18)	1	1.18
4	(1.17,1.17)	1	1.17
8	(1.17,1.17)	1	1.17
45	(1.17,1.17)	1	1.17
56	(1.16,1.16)	1	1.16

13	(1.15,1.15)	1	1.15
37	(1.13,1.13)	1	1.13
65	(1.13,1.13)	1	1.13
7	(1.11,1.11)	1	1.11
23	(1.11,1.11)	1	1.11
73	(1.11,1.11)	1	1.11
94	(1.11,1.11)	1	1.11
11	(1.1,1.1)	1	1.1
28	(1.09,1.09)	1	1.09
52	(1.09,1.09)	1	1.09
55	(1.07,1.07)	1	1.07
71	(1.07,1.07)	1	1.07
25	(1.06,1.06)	1	1.06
63	(1.06,1.06)	1	1.06
98	(1.06,1.06)	1	1.06
50	(1.05,1.05)	1	1.05
58	(1.05,1.05)	1	1.05
81	(1.05,1.05)	1	1.05
96	(1.05,1.05)	1	1.05
21	(1.04,1.04)	1	1.04
42	(1.04,1.04)	1	1.04
36	(1.03,1.03)	1	1.03
41	(1.03,1.03)	1	1.03
51	(1.03,1.03)	1	1.03
82	(1.02,1.02)	1	1.02
87	(1.02,1.02)	1	1.02
17	(1.01,1.01)	1	1.01
18	(1.01,1.01)	1	1.01
59	(1.01,1.01)	1	1.01
84	(1.01,1.01)	1	1.01
92	(1.01,1.01)	1	1.01
95	(1.01,1.01)	1	1.01
2	(1,1)	1	1
14	(1,1)	1	1
20	(1,1)	1	1
32	(1,1)	1	1
38	(1,1)	1	1
39	(1,1)	1	1
40	(1,1)	1	1
53	(1,1)	1	1
54	(1,1)	1	1
60	(1,1)	1	1
74	(1,1)	1	1
77	(1,1)	1	1
78	(1,1)	1	1

89	(1,1)	1	1

Average of $\frac{Average\ T_{ES}}{Median\ T_{ES}} = 1.14$

Random DAG Test Parameters			
Number of diseases	10		
Number of symptoms	10		
Number of DAGs	100		
Samples per DAG	100		
Total number of samples	10000		
Disease apriori probability	0.0000	to	0.0100
Symptom probability, no disease	0.0000	to	0.0100
Noisy-or weight	0.5000	to	1.0000
Density of edges	1.0000		

Comparison of chain lengths $(-T)$ to couple from the past.

Histogram of $(-T_{ES}, -T_?)$ *for batch of DAGs*

$-T_{ES}$

$-T$	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65536	131072	262144	
262144	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	
131072	0	0	0	0	0	0	0	0	0	0	0	1	1	0	7	4	7	9		
65536	0	0	0	0	0	0	0	0	0	0	1	1	7	10	14	18	20			
32768	0	0	0	0	0	0	1	1	4	2	6	10	21	27	22	49				
16384	0	0	1	0	0	0	0	3	5	4	21	30	55	46	106					
8192	0	1	0	0	2	1	2	4	16	25	40	56	53	120						
4096	0	0	0	2	1	4	5	19	37	46	61	75	179							
2048	0	1	2	4	3	21	21	31	57	67	71	211								
1024	0	0	7	15	18	27	59	78	79	68	204									
512	0	2	11	13	28	66	105	100	108	211										
256	0	2	23	31	63	82	100	95	221											
128	0	2	13	29	48	78	77	201												
64	0	0	25	49	53	57	178													
32	0	2	26	40	33	169														
16	0	2	20	28	107															
8	0	4	14	81																
4	0	1	56																	
2	0	31																		
1	5004																			5
	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	
	5004	48	198	292	356	505	548	532	527	423	404	384	316	204	150	71	27	10	1	1

$Average(-T_{ES}, -T_?) = (1.28e+003, 2.52e+003), \quad \frac{Average\ T_?}{Average\ T_{ES}} = 1.97$

List of DAGs ordered from slowest coalescence to fastest:

DAGNumber	Average$(-T_{ES}, -T_?)$	$\frac{Average\ T_?}{Average\ T_{ES}}$	$\frac{Average\ T_{ES}}{Median\ T_{ES}}$
6	(4.28e+004,5.74e+004)	1.34	1.31
28	(1.6e+004,4.02e+004)	2.52	1.95
34	(1.17e+004,2.74e+004)	2.34	1.43
89	(9.48e+003,1.63e+004)	1.72	1.16
24	(5.34e+003,1.37e+004)	2.56	1.3
68	(4.27e+003,1.34e+004)	3.14	2.09
81	(8.28e+003,1.1e+004)	1.33	1.01
73	(3.31e+003,9.36e+003)	2.83	1.62
94	(2.79e+003,6.76e+003)	2.43	1.36
48	(3.33e+003,5.35e+003)	1.61	1.63
19	(2.09e+003,5.04e+003)	2.41	2.04
82	(2.63e+003,4.52e+003)	1.72	1.28
57	(1.38e+003,4.94e+003)	3.59	1.35
32	(1.64e+003,4.28e+003)	2.61	1.61
16	(1.76e+003,4.06e+003)	2.3	1.72

54	(2.3e+003,3.53e+003)	1.54	2.24
25	(1.59e+003,2.43e+003)	1.53	1.55
67	(1.04e+003,2.38e+003)	2.28	2.04
26	(789,2.45e+003)	3.1	1.54
39	(893,1.86e+003)	2.08	1.74
63	(505,1.95e+003)	3.86	1.31
23	(451,1.95e+003)	4.33	1.76
21	(458,1.21e+003)	2.65	1.79
64	(222,837)	3.78	0.865
44	(287,797)	2.78	1.12
97	(243,777)	3.2	1.9
55	(263,673)	2.56	2.06
75	(199,620)	3.12	1.55
84	(264,555)	2.1	2.06
10	(130,592)	4.56	1.01
46	(120,514)	4.27	1.25
30	(111,500)	4.52	1.73
29	(167,450)	2.69	1.31
74	(108,446)	4.13	1.69
37	(109,432)	3.95	1.71
33	(77.4,387)	5	1.21
50	(84.4,382)	4.53	1.32
13	(102,363)	3.56	1.59
77	(77.8,341)	4.39	1.22
12	(52.7,193)	3.67	1.65
18	(48.2,178)	3.7	1.51
86	(36.5,171)	4.69	1.14
15	(25.9,171)	6.62	1.62
4	(31.7,144)	4.55	1.98
90	(30.7,130)	4.25	0.96
59	(25.5,114)	4.46	1.6
40	(28.2,98.9)	3.51	0.881
61	(17.1,74.1)	4.33	1.07
83	(8.77,41.1)	4.68	1.1
78	(7.08,14)	1.98	0.885
1	(1,1)	1	1
2	(1,1)	1	1
3	(1,1)	1	1
5	(1,1)	1	1
7	(1,1)	1	1
8	(1,1)	1	1
9	(1,1)	1	1
11	(1,1)	1	1
14	(1,1)	1	1
17	(1,1)	1	1

20	(1,1)	1	1
22	(1,1)	1	1
27	(1,1)	1	1
31	(1,1)	1	1
35	(1,1)	1	1
36	(1,1)	1	1
38	(1,1)	1	1
41	(1,1)	1	1
42	(1,1)	1	1
43	(1,1)	1	1
45	(1,1)	1	1
47	(1,1)	1	1
49	(1,1)	1	1
51	(1,1)	1	1
52	(1,1)	1	1
53	(1,1)	1	1
56	(1,1)	1	1
58	(1,1)	1	1
60	(1,1)	1	1
62	(1,1)	1	1
65	(1,1)	1	1
66	(1,1)	1	1
69	(1,1)	1	1
70	(1,1)	1	1
71	(1,1)	1	1
72	(1,1)	1	1
76	(1,1)	1	1
79	(1,1)	1	1
80	(1,1)	1	1
85	(1,1)	1	1
87	(1,1)	1	1
88	(1,1)	1	1
91	(1,1)	1	1
92	(1,1)	1	1
93	(1,1)	1	1
95	(1,1)	1	1
96	(1,1)	1	1
98	(1,1)	1	1
99	(1,1)	1	1
100	(1,1)	1	1

Average of $\frac{Average\ T_{ES}}{Median\ T_{ES}} = 1.25$

Random DAG Test Parameters			
Number of diseases	10		
Number of symptoms	10		
Number of DAGs	100		
Samples per DAG	100		
Total number of samples	10000		
Disease apriori probability	0.0000	to	0.0100
Symptom probability, no disease	0.0000		
Noisy-or weight	0.0100	to	1.0000
Density of edges	1.0000		

Comparison of chain lengths ($-T$) to couple from the past.

Histogram of $(-T_{ES}, -T_?)$ *for batch of DAGs*

$-T$ \ $-T_{ES}$	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65536	
65536	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	2
32768	0	0	0	0	0	3	0	0	0	0	2	3	3	3	4	7		25
16384	0	0	0	0	1	0	1	5	3	9	12	7	7	20	37			102
8192	0	0	1	1	6	1	8	12	17	36	64	52	37	143				378
4096	0	0	3	5	11	15	23	52	81	98	107	121	250					766
2048	0	3	8	17	41	51	72	129	150	172	185	435						1263
1024	0	3	25	58	68	126	170	214	237	194	529							1624
512	0	9	61	135	153	208	225	240	194	515								1740
256	0	6	82	189	206	175	205	177	537									1577
128	0	7	80	150	176	116	150	410										1089
64	0	6	61	141	101	82	289											680
32	0	2	27	64	51	214												358
16	0	3	21	40	148													212
8	0	1	12	96														109
4	0	0	50															50
2	0	20																20
1	5																	5
	5	60	431	896	962	991	1143	1239	1219	1024	899	618	297	167	41	7	1	10000

$Average(-T_{ES}, -T_?) = (686, 1.46e+003)$, $\frac{Average\ T_?}{Average\ T_{ES}} = 2.13$

List of DAGs ordered from slowest coalescence to fastest:

DAGNumber	Average$(-T_{ES}, -T_?)$	$\frac{Average\ T_?}{Average\ T_{ES}}$	$\frac{Average\ T_{ES}}{Median\ T_{ES}}$
90	(4.58e+003,1.07e+004)	2.34	1.12
100	(8.14e+003,8.14e+003)	1	1.99
76	(5.16e+003,7.51e+003)	1.46	1.26
69	(4.31e+003,5.41e+003)	1.26	1.05
91	(2.11e+003,6.32e+003)	3	1.03
58	(4.07e+003,4.98e+003)	1.22	0.994
80	(2.82e+003,3.92e+003)	1.39	1.38
35	(2.34e+003,3.77e+003)	1.61	1.52
92	(1.54e+003,4.13e+003)	2.68	1.51
5	(2.61e+003,3.52e+003)	1.35	1.28
43	(966,3.27e+003)	3.38	1.89
63	(674,3.24e+003)	4.81	1.32
4	(616,2.65e+003)	4.3	1.2
7	(959,2.51e+003)	2.61	1.87
49	(1.74e+003,1.95e+003)	1.12	1.7
79	(1.14e+003,2.32e+003)	2.04	1.11

68	(1.37e+003,2.16e+003)	1.58	1.34
12	(1.2e+003,1.93e+003)	1.61	1.17
45	(1.47e+003,1.7e+003)	1.16	1.43
27	(606,2.16e+003)	3.55	1.18
13	(447,2.19e+003)	4.9	1.75
84	(937,2.03e+003)	2.16	1.83
87	(1.12e+003,1.93e+003)	1.73	1.09
78	(640,2.07e+003)	3.24	1.25
52	(732,2.03e+003)	2.77	1.43
73	(344,2.04e+003)	5.93	1.34
81	(304,1.98e+003)	6.52	1.19
83	(1.19e+003,1.6e+003)	1.35	1.16
25	(681,1.67e+003)	2.45	1.33
64	(509,1.67e+003)	3.28	1.99
37	(558,1.63e+003)	2.91	2.18
20	(785,1.45e+003)	1.85	1.53
36	(948,1.26e+003)	1.33	1.85
66	(973,1.13e+003)	1.16	0.95
21	(488,1.38e+003)	2.82	1.91
96	(378,1.38e+003)	3.66	1.48
54	(540,1.27e+003)	2.35	1.06
15	(529,1.22e+003)	2.31	1.03
70	(699,1.07e+003)	1.53	1.37
24	(721,1.04e+003)	1.44	1.41
40	(291,1.21e+003)	4.14	1.14
3	(333,1.13e+003)	3.4	1.3
57	(204,1.12e+003)	5.48	1.59
46	(336,1.06e+003)	3.17	1.31
50	(551,956)	1.74	1.08
39	(441,980)	2.22	1.15
75	(253,1.02e+003)	4.04	0.987
10	(319,959)	3	1.25
41	(169,947)	5.59	1.32
60	(122,950)	7.76	1.28
23	(234,904)	3.85	1.83
86	(177,911)	5.14	1.39
72	(127,908)	7.16	1.98
55	(124,890)	7.19	0.967
9	(150,865)	5.75	1.18
16	(100,807)	8.06	1.56
82	(164,795)	4.84	1.28
65	(110,744)	6.76	1.72
17	(131,700)	5.36	2.04
19	(158,666)	4.23	1.23
53	(213,590)	2.76	1.67

14	(101,597)	5.9	1.58
94	(29.5,543)	18.4	0.922
32	(106,501)	4.74	1.65
62	(88.9,501)	5.63	1.39
85	(118,479)	4.08	1.84
6	(161,450)	2.79	1.26
1	(32.4,471)	14.6	1.35
26	(95.2,451)	4.74	1.49
95	(130,413)	3.16	1.02
8	(91.9,422)	4.59	1.44
28	(42.1,396)	9.39	1.32
30	(62.8,391)	6.23	0.981
61	(134,371)	2.76	2.1
11	(32.5,393)	12.1	2.03
34	(12.9,383)	29.7	1.61
44	(28.9,382)	13.2	1.8
98	(66.4,359)	5.4	1.04
33	(77.7,338)	4.35	1.21
31	(27.9,332)	11.9	1.74
99	(42.4,328)	7.74	1.33
97	(15.4,296)	19.2	0.962
48	(109,273)	2.5	1.71
88	(50.7,287)	5.67	1.58
59	(34.4,276)	8.04	1.07
38	(13,272)	21	1.62
71	(65.3,264)	4.05	1.02
51	(36.6,225)	6.15	1.14
2	(10.3,218)	21.1	1.29
74	(18.7,216)	11.6	1.17
89	(11.5,208)	18.1	1.44
29	(23.9,199)	8.34	1.5
42	(11.9,190)	15.9	1.49
56	(31.3,187)	5.97	0.977
93	(10.7,187)	17.5	1.34
47	(12.8,176)	13.7	1.6
18	(11.8,164)	13.9	1.48
67	(11.3,162)	14.3	1.42
22	(7.54,149)	19.8	0.943
77	(7.42,113)	15.2	0.927

Average of $\frac{Average\ T_{ES}}{Median\ T_{ES}} = 1.39$

Chapter 10

Bibliography

- Jensen, Finn V. (1996) *An Introduction to Bayesian networks*, London, UCL Press

- Johnson, Valen E. (1996) "Studying convergence of Markov chain Monte Carlo algorithm using coupled sample paths", *J. Amer. Statist. Assoc.*, **91**, 154-166.

- Neal, Radford M. (1993) "Probabilistic Inference Using Markov Chain Monte Carlo Methods", Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto.

- Neapolitan, Richard E. (1990) *Probabilistic Reasoning in Expert Systems*, John Wiley & Sons, Inc.

- Propp, James G., and Wilson, David B. (1996) "Exact Sampling with Coupled Markov Chains and Applications to Statistical Mechanics", *Random Structures and Algorithms*, **9**, 223-252.

- Pearl, Judea (1987) "Evidential Reasoning Using Stochastic Simulation of Causal Models", *Artifical Intelligence*, **32**, 245-257.

- Pearl, Judea (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, Inc.

- Rosenthal, Jeffrey S. (1995) "Convergence rates of Markov chains", *SIAM Review*, **37**, 387-405.