Heuristics for Automatically Decomposing a Stochastic Process for Factored Inference

Charlie Frogner and Avi Pfeffer

TR-04-07

Computer Science Group
Harvard University
Cambridge, Massachusetts

Heuristics for Automatically Decomposing a Stochastic Process for Factored Inference

Charlie Frogner
School of Engineering and Applied Sciences
Harvard University
frogner@deas.harvard.edu

Avi Pfeffer
School of Engineering and Applied Sciences
Harvard University
avi@eecs.harvard.edu

Abstract

Dynamic Bayesian networks are factored representations of stochastic processes. Despite their factoredness, exact inference in DBNs is generally intractable. One approach to approximate inference involves factoring the variables in the process into components. In this paper we study efficient methods for automatically decomposing a DBN into weakly-interacting components so as to minimize the error in inference entailed by treating them as independent. We investigate heuristics based on two views of weak interaction: mutual information and the degree of separability ([Pf01] and [Pf06]). It turns out, however, that measuring the degree of separability exactly is probably intractable. We present a method for estimating the degree of separability that includes a mechanism for trading off efficiency and accuracy. We use the aforementioned heuristics in two clustering frameworks to find weakly-interacting components, and we give an empirical comparison of the results in terms of the error encountered in the belief state across the whole system, as well as that in the belief states across single components.

1 Introduction

Dynamic Bayesian networks (DBNs) are factored representations of stochastic processes. They generalize hidden Markov models, and are used in a broad range of applications, including speech processing and genomics. Generally speaking, inference in DBNs is intractable due to the size of the belief state over the variables in the process, and so approximate inference is necessarily used for monitoring this belief state as the process evolves.

Factored inference, and in particular the Boyen-Koller algorithm (BK), is widely used for approximate monitoring. BK reduces the size of the belief state representation by decomposing the belief state into the product of smaller, local belief states, thus grouping the variables in the process into components that are taken to be independent. [BK98] shows that the error that results from this approximation does not accumulate: the total approximation error is bounded over the lifetime of the process, and this bound depends on the quality of the approximation, i.e. how accurate it is to treat the components as independent. In other words, the approximation error introduced in BK is bounded as a function of how weakly the components interact.

In this report we attempt to answer the question of how best automatically to decompose a stochastic process into components so as to minimize the error entailed by treating those components as independent. Intuitively, one would want to decompose a system to maximize some measure of independence between the components. We are constrained by the fact that a dynamic Bayesian network specifies only the probabilistic transition of the state of a system from one timestep to the next, and so we are attempting to find components that we *expect* to interact weakly over the course of the process by examining only these transition probabilities. In this work we investigate efficient heuristics for predicting weak interaction, which are suitable for doing automatic decomposition of a DBN.

We focus primarily on two conceptions of weak interaction. The mutual information between random variables is a measure of their correlation, while the degree of separability ([Pf01] and [Pf06]) gives the degree to which the given transition probabilities will result in error-free propagation of the factors' marginal distributions. Although it is probably intractable to compute the degree of separability exactly for a given probability distribution, we describe in this work an algorithm for computing a useful lower bound on this

quantity. We compare empirically the performance of several heuristics based on these two quantities, which are used in a graph partitioning and a divisive clustering framework to find weakly-interacting components, given a DBN. Among other things, we find that a useful predictor of correlation between variables is their relatedness as children of common parents in the DBN. The mutual information between such variables, when used in clustering, efficiently found the components that yielded the lowest error, both in terms of the error in the full joint belief state and in each component's marginal belief state.

2 Background

2.1 DBNs

A *dynamic Bayesian network* (DBN) ([DK89], [Mu02]) represents a dynamic system consisting of some set of variables that co-evolve in discrete timesteps. We will denote the set of variables in the system by $\mathbf{X}$. We call the probability distribution over the possible states of the system at a given timestep the *belief state*. The DBN gives us the probabilities of transitioning from any given system state at t to any other system state at time $t+1$, and it does so in a factored way: the probability that a variable takes on a given state at $t+1$ depends only on the states of a subset of the variables in the system at t. We can hence represent this transition model as a Bayesian network containing the variables in $\mathbf{X}$ at timestep t, say $\mathbf{X}_t$, and the variables in $\mathbf{X}$ at timestep $t+1$, say $\mathbf{X}_{t+1}$ – this is called a *2-TBN* (for two-timeslice Bayesian network). By inferring the belief state over $\mathbf{X}_{t+1}$ from that over $\mathbf{X}_t$, and conditioning on observations, we propagate the belief state through the system dynamics to the next timestep. The specification of a DBN also includes a prior belief state at time $t = 0$.

2.2 Factored Inference

Note that, although each variable at $t+1$ may only depend on a small subset of the variables at t, its belief state might be correlated implicitly with the belief state for any variable in the system, as the influence of any variable might propagate through intervening variables over multiple timesteps. As a result, the whole belief state over $\mathbf{X}$ (at a given timestep) in general is not factored. [BK98] finds that, despite this fact, we can factor the system into components that we treat as being independent, and the error incurred by doing so remains bounded over the course of the process. As a result, we can reduce the overall size of the belief state representation by choosing an appropriate factorization and just keeping independent beliefs over the components. The BK algorithm hence approximates the belief state at a given timestep as the product of the local belief states for the components (their marginal distributions), and does exact inference to propagate this approximate belief state to the next timestep. Both the Factored Frontier ([MW01]) and Factored Particle ([NPP02]) algorithms also rely on this idea of a factored belief state representation.

2.3 Information Theory

We will be concerned primarily with the error between probability distributions: either the error between the product of the component belief states at a given time and the exact joint belief state at that time – the whole belief state error – or that between a single component's local belief state and the marginal distribution over the component's variables in the exact joint belief state – the component belief state error. Both errors we will define by the *relative entropy* between the distributions.

Definition 2.1 (Relative entropy)**.** *The **relative entropy** between two probability distributions* $\mathbf{p}_1(\mathbf{x})$ *and* $\mathbf{p}_2(\mathbf{x})$ *is*

$$\boldsymbol{D}(\mathbf{p}_1(\mathbf{x})||\mathbf{p}_2(\mathbf{x})) = \boldsymbol{E}_{\mathbf{p}_1(\mathbf{x})} \log \frac{\mathbf{p}_1(\mathbf{x})}{\mathbf{p}_2(\mathbf{x})}$$

where $\boldsymbol{E}_{\mathbf{p}}$ *denotes the expectation taken with respect to distribution* $\mathbf{p}$*.*

The mutual information between two variables X and Y, jointly distributed according to $\mathbf{p}(XY)$, is a measure of how strongly they interact (or, inversely, how independent they are). It corresponds to the reduction in uncertainty about X's value – the entropy of X – from knowing Y's value, and vice versa. If Y completely determines X, then their mutual information will be exactly the entropy of X to begin with: knowing Y removes all of the uncertainty about X. If they are completely independent, then their mutual information will be 0: knowing Y leaves X's uncertainty unchanged.

Definition 2.2 (Mutual information)**.** *The **mutual information** between* X *and* Y *is*

$$\begin{aligned}\boldsymbol{I}(X;Y) &= \boldsymbol{D}(\mathbf{p}(XY)||\mathbf{p}(X)\mathbf{p}(Y)) \\ &= \boldsymbol{E}_{\mathbf{p}(XY)} \log \frac{\mathbf{p}(XY)}{\mathbf{p}(X)\mathbf{p}(Y)}\end{aligned}$$

This definition extends to the case that we are conditioning on a third variable, Z.

Definition 2.3 (Conditional mutual information)**.** *The **conditional mutual information** between* X

and Y, conditional on Z, is

$$\begin{aligned}\boldsymbol{I}(X;Y|Z) &= \boldsymbol{D}(\mathbf{p}(XY|Z)||\mathbf{p}(X|Z)\mathbf{p}(Y|Z)) \\ &= \boldsymbol{E}_{\mathbf{p}(Z)}\boldsymbol{E}\mathbf{p}(XY|Z)\log\frac{\mathbf{p}(XY|Z)}{\mathbf{p}(X|Z)\mathbf{p}(Y|Z)}\end{aligned}$$

2.4 Sufficiency and Separability

[Pf01] and [Pf06] introduce conditions under which a single variable's (or component's) marginal distribution will be propagated accurately through the probabilistic transition. The *degree of separability* is a property of a conditional probability distribution that describes the degree to which that distribution can be decomposed as the sum of simpler conditional distributions, each of which depends on only a subset of the conditioning variables. For example, let $\mathbf{p}(Z|XY)$ give the probability distribution for Z given X and Y. If $\mathbf{p}(Z|XY)$ is separable in terms of X and Y to a degree α, this means that we can write

$$\begin{aligned}\mathbf{p}(Z|XY) = \alpha(\gamma\mathbf{p}_X(Z|X) + (1-\gamma)\mathbf{p}_Y(Z|Y)) \\ + (1-\alpha)\mathbf{p}_{XY}(Z|XY)\end{aligned} \quad (1)$$

for some conditional probability distributions $\mathbf{p}_X(Z|X)$, $\mathbf{p}_X(Z|Y)$, and $\mathbf{p}_{XY}(Z|XY)$ and some parameter γ. We will say that the degree of separability is the maximum alpha such that there exist $\mathbf{p}_X(Z|X)$, $\mathbf{p}_X(Z|Y)$, and $\mathbf{p}_{XY}(Z|XY)$ and γ that satisfy (1).

It turns out that the degree of separability corresponds to the degree to which the marginal value of the child variable, Z, only depends on the marginal values of the parent variables, X and Y. This property is called *sufficiency*: the parents' marginals are sufficient to determine the child marginal. [Pf01] and [Pf06] have showed that if a system is highly separable, then the BK algorithm encounters low error in the components' marginal distributions.

3 Computing the Degree of Separability

[Pf06] incorrectly presents the computation of the degree of separability as a linear program. We correct this here, and describe a new algorithm for estimating this quantity.

Say we are given a conditional distribution $\mathbf{p}(Z|XY)$. To compute the degree of separability for $\mathbf{p}(Z|XY)$ in terms of X and Y we must find some distributions $\mathbf{p}_X(Z|X)$, $\mathbf{p}_Y(Z|Y)$ and $\mathbf{p}_{XY}(Z|XY)$ and some constants α and γ such that their combination as given in equation (1) sums exactly to the values of $\mathbf{p}(Z|XY)$ and such that α is as large as possible. In other words, we have a constrained optimization problem, attempting to maximize α, as follows:

Given $\mathbf{p}(z_i|x_jy_k), \forall i,j,k$, find

$$\begin{aligned}
\max\ & \alpha \\
s.t.\ & \alpha\gamma_1\mathbf{p}_X(z_i|x_j) + \alpha\gamma_2\mathbf{p}_Y(z_i|y_k) \\
& \quad + (1-\alpha)\mathbf{p}_{XY}(z_i|x_jy_k) = P(z_i|x_jy_k) && \forall i,j,k \\
& \alpha\gamma_1 + \alpha\gamma_2 + (1-\alpha) = 1 \\
& (1-\alpha) \geq 0 \\
& 0 \leq \mathbf{p}_X(z_i|x_j) \leq 1 && \forall i,j \\
& 0 \leq \mathbf{p}_Y(z_i|y_k) \leq 1 && \forall i,k \\
& 0 \leq \mathbf{p}_{XY}(z_i|x_jy_k) \leq 1 && \forall i,j,k \\
& \sum_i \mathbf{p}_X(z_i|x_j) = 1 && \forall j \\
& \sum_j \mathbf{p}_Y(z_i|y_k) = 1 && \forall k \\
& \sum_k \mathbf{p}_{XY}(z_i|x_jy_k) = 1 && \forall j,k
\end{aligned}$$

Although the objective function is linear, this is a non-linearly constrained problem: Each constraint of the first type, specifying that $\mathbf{p}(Z|XY)$ actually decomposes into the sum $\alpha\gamma_1\mathbf{p}_X(Z|X) + \alpha\gamma_2\mathbf{p}_Y(Z|Y) + (1-\alpha)\mathbf{p}_{XY}(Z|XY)$, is of degree 2. Unfortunately the feasible region for this optimization is highly non-convex.

We find, however, that a relaxation of the problem features a landscape of optimal values that is suitable for hill-climbing. This gives us a relatively efficient means of solving the problem approximately. We define the relaxed problem as follows.

Given α, γ_1, γ_2 and $\mathbf{p}(z_i|x_jy_k), \forall i,j,k$, find

$$\begin{aligned}
\min\ & (\max_{(i,j,k)}|\beta_{(i,j,k)}|) \\
s.t.\ & \alpha\gamma_1\mathbf{p}_X(z_i|x_j) + \alpha\gamma_2\mathbf{p}_Y(z_i|y_k) \\
& \quad + (1-\alpha)\mathbf{p}_{XY}(z_i|x_jy_k) \\
& \quad + \beta_{(i,j,k)} = P(z_i|x_jy_k) && \forall i,j,k \\
& 0 \leq \mathbf{p}_X(z_i|x_j) \leq 1 && \forall i,j \\
& 0 \leq \mathbf{p}_Y(z_i|y_k) \leq 1 && \forall i,k \\
& 0 \leq \mathbf{p}_{XY}(z_i|x_jy_k) \leq 1 && \forall i,j,k \\
& \sum_i \mathbf{p}_X(z_i|x_j) = 1 && \forall j \\
& \sum_j \mathbf{p}_Y(z_i|y_k) = 1 && \forall k \\
& \sum_k \mathbf{p}_{XY}(z_i|x_jy_k) = 1 && \forall j,k
\end{aligned}$$

We are fixing the parameters α, γ_1 and γ_2 and finding the distributions $\mathbf{p}_X(Z|X)$, $\mathbf{p}_Y(Z|Y)$, and $\mathbf{p}_{XY}(Z|XY)$ that get as close as possible, in terms of the L_∞ distance $[\max_{(i,j,k)}|\beta_{(i,j,k)}|]$, to summing to $\mathbf{p}(Z|XY)$. We'll denote the solution to the linear program for the parameters α, γ_1, and γ_2 by $\mathcal{B}(\alpha,\gamma_1,\gamma_2)$

– this is the minimum L_∞ distance given the parameters. If this distance is 0, then we have found a solution to the original optimization problem, and so the given α is a lower bound on the degree of separability for $\mathbf{p}(Z|XY)$. This relaxed optimization problem is a linear program.

The optimal value of this linear program in changes continuously with changes in the parameters α, γ_1 and γ_2. As a result, at any given solution to this program, i.e. for any set of distributions $\mathbf{p}_X(Z|X)$, $\mathbf{p}_Y(Z|Y)$, and $\mathbf{p}_{XY}(Z|XY)$, we can compute a gradient of the optimal value of the objective function with respect to changes in the parameters γ_1 and γ_2. In other words, the problem of finding the parameters γ_1 and γ_2 that minimize the value of $\mathcal{B}(\alpha, \gamma_1, \gamma_2)$, for a fixed α, is amenable to a gradient search. As a result, we can reduce the original problem to that of finding the maximum α such that there exists γ_1 and γ_2 that yield $\mathcal{B}(\alpha, \gamma_1, \gamma_2) = 0$.

3.1 Finding a Gradient

Say we formulate the linear program in matrix form:

$$\begin{aligned} min \;\; & \mathbf{cx} \\ s.t. \;\; & \mathbf{Ax} \leq \mathbf{b} \end{aligned}$$

We will discuss *coefficients* in $\mathbf{A}$, some of which correspond to the values $\alpha\gamma_1$, $\alpha\gamma_2$, and $(1-\alpha)$ from the relaxed problem. [Fr85] and [DW03], among others, analyze the gradient of the optimal value with respect to changes in the coefficients in $\mathbf{A}$. Suppose we evaluate the linear program given some parameters α, γ_1, γ_2 and $\mathbf{p}(Z|XY)$. Let $\mathbf{x}$ give a solution to the *primal* linear program, i.e. values of $\mathbf{p}_X(z_i|x_j)$, $\mathbf{p}_Y(z_i|y_k)$, $\mathbf{p}_{XY}(z_i|x_jy_k)$, and $\beta_{(i,j,k)}$, $\forall i, j, k$, that yield $[\max_{(i,j,k)} |\beta_{(i,j,k)}|] = \mathcal{B}(\alpha, \gamma_1, \gamma_2)$. Then, to get the gradient of $\mathcal{B}(\alpha, \gamma_1, \gamma_2)$ with changes in each of the coefficients in $\mathbf{A}$, we evaluate the *dual* of this linear program, to get the dual solution $\mathbf{u}$. The gradient is

$$\nabla_{\mathbf{A}} = -\mathbf{u} \times \mathbf{x}$$

where $\times$ here denotes the outer product.

Note, however, that we are looking for the gradient of $\mathcal{B}(\alpha, \gamma_1, \gamma_2)$ with respect to changes in γ_1 and γ_2, but $\nabla_{\mathbf{A}}$ gives us the gradient with respect to changes in *each* of the coefficients in $\mathbf{A}$, irrespective of the correspondence between the linear program coefficients and our parameters. Specifically, multiple coefficients in $\mathbf{A}$ correspond to the values $\alpha\gamma_1$ and $\alpha\gamma_2$ in the linear program, and so we must sum together the gradient values for those coefficients. Let ∇_γ be the gradient of $\mathcal{B}(\alpha, \gamma_1, \gamma_2)$ with respect to changes in the values of $\alpha\gamma_1$ and $\alpha\gamma_2$, obtained by summing the appropriate values of $\nabla_{\mathbf{A}}$.

In order to find the values of γ_1 and γ_2 that minimize the optimal value $\mathcal{B}(\alpha, \gamma_1, \gamma_2)$, we use the gradient to find the direction of maximum negative change in the optimal value, respecting the constraints on γ_1 and γ_2 that are given by the definition of separability. Say that we are changing each γ_m by at most ε. Then the step of maximum negative change is Δ:

$$\begin{aligned} max \;\; & -\nabla_\gamma \cdot \Delta \\ s.t \;\; & \alpha(\gamma_1 + \Delta_1) + \alpha(\gamma_2 + \Delta_2) + (1-\alpha) = 1 \\ & -\varepsilon \leq \Delta_1, \Delta_2 \leq \varepsilon \end{aligned}$$

Suppose the parameters for which we solved this iteration of the linear program are γ_1 and γ_2. We can let these parameters at the next iteration be $\gamma_1 + \Delta_1$ and $\gamma_2 + \Delta_2$, repeating the procedure until $\Delta = \mathbf{0}$, in which case we have found a local minimum of the optimal value function with respect to γ_1 and γ_2. If the optimal value is $[\max_{(i,j,k)} |\beta_{(i,j,k)}|] = 0$, then we know that $\mathbf{p}(Z|XY)$ is α-separable.

3.2 Dealing with Complexity

Although the preceding approximation algorithm for the degree of separability is formulated as a sequence of linear programs, the size of these linear programs goes as the size of the input distribution $\mathbf{p}(Z|X_1...X_n)$ – it is exponential in the number of variables. We have, for example, for every input value $\mathbf{p}(z_i|x_jy_k)$ one constraint of the type $\mathbf{p}(z_i|x_jy_k) = \alpha\gamma_1\mathbf{p}_X(z_i|x_j) + \alpha\gamma_2\mathbf{p}_Y(z_i|y_k) + (1-\alpha)\mathbf{p}_{XY}(z_i|x_jy_k)$, in which appear the linear program variables $\mathbf{p}_X(z_i|x_j)$, $\mathbf{p}_Y(z_i|y_k)$, and $\mathbf{p}_{XY}(z_i|x_jy_k)$.

One method for trading off the size of the linear program with the quality of the approximation is to introduce only a subset of the constraints of the above type: only certain of the values of $\mathbf{p}(Z|XY)$ are given to the algorithm as constraints, and the variables $\mathbf{p}_X(z_{i'}|x_{j'})$, $\mathbf{p}_Y(z_{i'}|y_{j'})$, and $\mathbf{p}_{XY}(z_{i'}|x_{j'}y_{k'})$ that *do not* appear in any of the constraints of the above type are unconstrained – we don't have to include them in our linear program at all. We can hence control the exponential increase in linear program size by choosing a subset of the values of $\mathbf{p}(Z|XY)$ to use as constraints. Which constraints are introduced matters: as we increase the number of constraints being introduced, choosing the groups of constraints $\mathbf{p}(z_i|x_jy_k)$, $\forall i$ and fixing x_j and y_k yields on average slower convergence to the best estimated value of the degree of separability than when we introduce the groups of constraints $\mathbf{p}(z_i|x_jy_k)$, $\forall j, k$ and fixing z_i. Introducing the constraints at random performs slightly worse than the latter (figure 1).

4 Automatically Finding Components

We examine the efficacy of several different heuristics for predicting weak interaction between components,

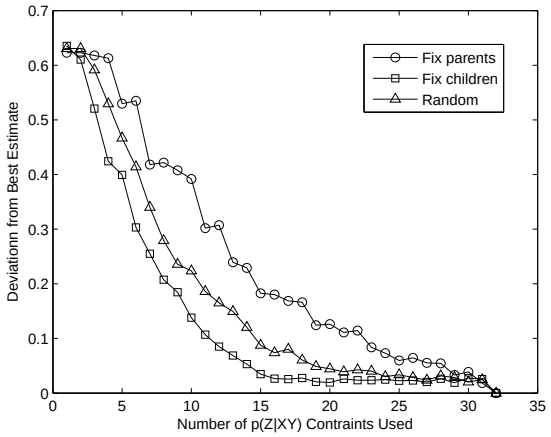

Figure 1: Convergence of approximate degree of separability as more constraints are introduced.

used in two different clustering frameworks. The first uses graph partitioning techniques and the second does divisive clustering. The goal is efficiently to assign the variables in the DBN to components in such a way as to minimize the error entailed by taking those components to be independent.

4.1 Heuristics for Strength of Interaction

Determining the degree to which two sets of random variables interact usually involves computing some quantity as a function of their joint probability distribution. Doing so is inherently exponential in the number of variables involved. We deal with this complexity in two ways. We measure primarily the *pairwise* interactions between variables, rather than interactions between entire entire components. We moreover attempt to leverage the factoredness of the DBN to measure interactions only between those canonical variables that are somehow related by one step of the transition model. In defining a heuristic, then, we make two decisions: how to define adjacency between variables, and how to define their strength of interaction. Note that, as we will discuss later, in most cases we must assume some prior belief state at time t in order to compute the desired quantity.

4.1.1 Children of Common Parents

Suppose first that X_{t+1} and Y_{t+1} depend on some common parents $\mathbf{Z}_t$. As X and Y share a common, direct influence, we might expect them to to become correlated in the joint belief state.

1. *mi-cc*: The strength of interaction between X and Y is $I(X_{t+1}; Y_{t+1})$. By the definition of mutual information, variables are close to being independently distributed if and only if they carry very little information about each other – if $I(X_{t+1}; Y_{t+1})$ tends to be low, then, we can put X and Y in different components and expect little error (with respect to their joint distribution). Although there might be no direct dependence on Y_{t+1} in X_{t+1}'s conditional probability distribution (and vice versa), we can obtain their mutual information by assuming some belief state over their parents at t, and with this prior computing $\mathbf{p}(X_{t+1}, Y_{t+1}, Parents(X_{t+1}) \cup Parents(Y_{t+1}))$. Marginalizing out all parents we get $\mathbf{p}(X_{t+1}, Y_{t+1})$, in terms of which $I(X_{t+1}; Y_{t+1})$ is defined.

2. *mci-cc*: The strength of interaction between X and Y is $I(X_{t+1}; Y_{t+1}|\mathbf{Z}_t)$: Instead of computing the mutual information between X_{t+1} and Y_{t+1} by getting their marginal distribution, we compute their expected mutual information given an observation of $\mathbf{Z}_t$. Intuitively, we capture the degree to which we expect X and Y to become correlated by observations at the previous timestep As in *mi-cc*, we assume a prior over all the parents of X_{t+1} and Y_{t+1} in order to derive $I(X_{t+1}; Y_{t+1}|\mathbf{Z}_t)$.

3. *sep-cc*: We can compute the degree of separability for the joint conditional distribution $\mathbf{p}(X_{t+1}, Y_{t+1}|Parents(X_{t+1}) \cup Parents(Y_{t+1}))$. We want to *maximize* this value for variables that are joined in a component, as a high degree of separability implies that the error of the factor marginal distribution at any time will be low. Note that the degree of separability is defined in terms of *groups* of parent variables. If we have, for example, $\mathbf{p}(Z|WXY)$, then this distribution might be highly separable in terms of the groups XY and W, but not in terms of WX and Y. If, however, $\mathbf{p}(Z|WXY)$ is highly separable in terms of W, X and Y grouped separately, then it is at least as separable in terms of any other groupings.

4.1.2 Parents of Common Children

Suppose now that X_t and Y_t influence common child variables $\mathbf{Z}_{t+1}$. In this case we might *care* more about any correlations between X and Y, because they jointly influence $\mathbf{Z}$ – we would expect that, if X and Y are placed in different components, then the accuracy of $\mathbf{Z}$'s marginal distribution will depend on how correlated X and Y were.

1. *mci-pp*: The strength of interaction between X and Y is their conditional mutual information, given $\mathbf{Z}_{t+1}$: $I(X_t; Y_t|\mathbf{Z}_{t+1})$. We compute

$I(X_t; Y_t|\mathbf{Z}_{t+1})$ similarly to the other information-theoretic quantities, assuming first a prior distribution over all the parents of $\mathbf{Z}_{t+1}$ and using Bayes rule to get the distribution $\mathbf{p}(X_t, Y_t|\mathbf{Z}_{t+1})$.

2. *sep-pp*: If two parents are highly *non*-separable in a common child's conditional distribution, then the child's marginal distribution can be rendered inaccurate by placing these two parents in different components. The strength of interaction between X and Y is defined to be the maximum degree of non-separability over the variables in $\mathbf{Z}_{t+1}$ taken individually. Intuitively, although $Z_1 \in \mathbf{Z}_{t+1}$ might be highly separable in terms of X_t and Y_t, if $Z_2 \in \mathbf{Z}_{t+1}$ is highly non-separable in terms of X_t and Y_t, then we don't want to put X and Y in different components.

4.1.3 Parent to Child

Suppose next that X_t is a parent of Y_{t+1}. Then X and Y are directly correlated by X_t's influence on Y_{t+1}.

1. *mi-pc*: The strength of interaction between X and Y is $I(X_t; Y_{t+1})$. By the definition of mutual information, the more information X_t carries about Y_{t+1}, the more X_t and Y_{t+1} are correlated. As with the other information-theoretic quantities, in computing $I(X_t; Y_{t+1})$ we assume a prior over all the parents of Y_{t+1}.

2. *mi-ts*: Between any two canonical variables X and Y such that either X_t is a parent of Y_{t+1} or vice versa, the strength of their interaction is the mutual information between the two pairs of variables (X_t, X_{t+1}) and (Y_t,Y_{t+1}). This heuristic extends *mi-pc* by including all the possible direct influences between X and Y. Here we assume a prior over all the parents of X_{t+1} and Y_{t+1}.

4.1.4 Other Heuristics

We lastly look at two non-pairwise heuristics.

1. *out-deg*: [BK98] shows that the total error in the joint belief state over $\mathbf{X}$ is bounded as a function of the *mixing rate* of the system. Intuitively, the mixing rate is the rate at which the process forgets its old states, and corresponds approximately to the stochasticity of the probabilistic transition from t to $t+1$. Importantly, the error bound goes inversely with the mixing rate, and the overall mixing rate of the system is inversely exponential in the maximum *out-degree* of all factors – for a tighter error bound, we want to minimize the maximum out-degree of our components.

2. *in-deg*: The same statement for the whole-system mixing rate also shows that it goes inversely with the maximum *in-degree* of all the factors. Again, we want to minimize the component in-degrees for a higher mixing rate and, by extension, a lower error bound.

4.2 Assuming a Prior

The DBN specifies only transition probabilities, such as $\mathbf{p}(Y_{t+1}|X_t)$, without specifying the prior distribution, e.g. $\mathbf{p}(X_t)$ – this prior is the belief state that we are propagating from timestep to timestep. In order to compute information theoretic quantities like the mutual information between two variables we need to compute their joint distribution, for example $\mathbf{p}(Y_{t+1}, X_t) = \mathbf{p}(Y_{t+1}|X_t) \cdot \mathbf{p}(X_t)$. To do so we need to assume a prior. One approach would be to compute the maximum over all possible priors of the quantity we are interested in. For the mutual information, this can be done with the Blahut and Arimoto algorithm, given in [Ar72], which converges asymptotically to the maximum mutual information prior. We find, however, that the mutual information found using the uniform prior does not underestimate the maximum by a large margin, and the Blahut and Arimoto algorithm presents nontrivial computational costs. The uniform prior does often underestimate, however, the mean value of the *conditional* mutual information quantities we compute, while the maximum value of those quantities can be substantially higher than the mean. We investigated the effect of randomly sampling prior distributions to estimate the mean value of these quantities, instead of assuming a single prior distribution. We found no significant difference in the average performance of the clustering algorithms as compared to that using the uniform prior. Although for efficiency we used the uniform prior in our subsequent investigations, the effect of different prior assumptions deserves further inquiry.

4.3 Graph Partition

Suppose we construct a graph over the state variables in the DBN, with connectivity defined as per our pairwise scheme and undirected edges weighted according to the heuristic strength of interaction between the two variables. We can imagine clustering the variables so as to minimize the net interaction between components. This is a *k-way min-cut*, found with a spectral graph partitioning algorithm. For more information see, e.g., [NJW01]. Note that we must select an appropriate number of components to find (k). To do so, we set a *maximum component weight*, being the maximum allowable size of the belief state representation for that component. We find the minimum number of

components such that the weight of the largest is at most this maximum weight.

4.4 Divisive Clustering

We can gain some flexibility in defining between-component interactions by using a divisive clustering framework. It is possible, for example, to minimize the number of children of the variables in a component (the out-degree). We find that a particular formulation of divisive clustering works reasonably well for finding weakly-interacting components.

Again, we have defined a maximum component weight. We begin with all of the canonical variables assigned to one component and, as long as there exists a component whose weight is larger than the maximum, we do the following. For every component that is too large, we pick the two variables in that component that are farthest apart, according to whichever metric we are using, and use them as the seeds for two new components. For each remaining variable from the original component, we place that variable in the new component to which it is closest. Subsequent to this division, we iterate over the variables in each new component and for each move it to the other cluster if doing so will make the clusters farther apart. Once no advantageous switch is found, we stop, and find another component whose weight is larger than the maximum, repeating until no such component exists.

5 Empirical Results

We compared the performance of the preceding heuristics on 1000 randomly-generated DBNs, where belief state monitoring was done with the BK algorithm. In 500 of these, each variable at time $t+1$ depended on itself and one other, randomly-chosen variable at time t (Table 1). In the remaining 500, each variable at $t+1$ depended on itself and two others at t (Table 2). Every DBN contained 12 state variables – we found that this number admitted interesting partitions while also allowing a relatively large number of experiments. Each DBN additionally had 4 evidence variables that were noisy observations of one state variable each. For each DBN we did exact inference to generate a sequence of observations to be used in all trials, and for each trial the results were averaged over 20 timesteps. For efficiency in doing partitioning with *sep-cc*, we introduced only half of the $\mathbf{p}(Z|XY)$ constraints, as described in section 3.2. All errors are given here in terms of the relative entropy between the approximate belief state and the true belief state, found by doing exact inference.

We also tried to capture some of the variation resulting from the different definitions of adjacency between variables by using unweighted values for the strength of interaction. These results are called *unw-cc*, where children of a common parent were connected, *unw-pc*, where parents were connected to children, and *unw-pp*, where parents of a common child were connected.

We take three conclusions from the results. The first is that correlations that arise as a result of sharing common parent influences provide the best proxy for weak interaction. Interestingly, although *unw-cc* underperformed in terms of per-component error, *mi-cc* provided enough information about the relevant correlations to outperform all other heuristics by a significant margin, in terms of both per-component error and joint error. The second is that taking into account correlations between parents of common children, although useful for achieving relatively low error in the childrens' marginals, tends to group variables that are not correlated themselves, producing significantly higher joint error. Our third conclusion, shown in Figure 2, is that the similarity in performance between *mci-pp* and *sep-pp* may be due to the fact that the degree of separability (taken individually for each child variable's CPD) and the mutual conditional information between parents of common children are linked. In highly separable conditionals, the parents were close to being conditionally independent given the children. Generally speaking, the degree of separability has not proved itself amenable to good pairwise heuristics. Our results suggest, however, that the mutual information between children of common parents provides an effective heuristic for weak interaction.

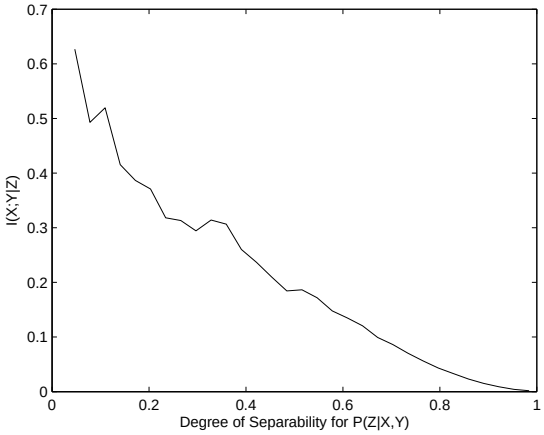

Figure 2: Mutual conditional information between parents vs. degree of separability

6 Conclusions

We have investigated the efficacy of heuristics for automatically decomposing a dynamic system into weakly-

Table 1: DBNs with 2 parents per canonical variable

Heuristic	Graph Error		Divisive Error	
	Factor	Joint	Factor	Joint
	$\times 10^{-4}$	$\times 10^{-2}$	$\times 10^{-4}$	$\times 10^{-2}$
mci-cc	0.960	4.48	1.05	4.34
mci-pp	0.517	5.28	0.696	5.17
mi-cc	**0.372**	**2.19**	**0.430**	**1.98**
mi-pc	0.888	4.35	0.983	4.28
mi-ts	0.843	4.33	0.990	4.36
sep-pp	0.650	5.29	0.654	5.12
sep-cc	0.731	3.92	0.885	3.99
out-deg	-	-	1.10	4.93
in-deg	-	-	1.05	4.88
unw-cc	0.848	3.89	8.81	3.77
unw-pc	1.28	4.93	1.30	4.91
unw-pp	0.884	5.20	0.936	5.08

Table 2: DBNs with 3 parents per canonical variable

Heuristic	Graph Error		Divisive Error	
	Factor	Joint	Factor	Joint
	$\times 10^{-4}$	$\times 10^{-2}$	$\times 10^{-4}$	$\times 10^{-2}$
mci-cc	1.93	3.67	2.21	3.68
mci-pp	1.12	3.55	1.19	3.51
mi-cc	**0.847**	**2.17**	**0.963**	**2.12**
mi-pc	1.15	2.78	1.35	2.76
mi-ts	1.00	2.87	1.00	2.88
sep-pp	1.45	3.40	1.72	3.44
sep-cc	1.70	3.41	1.88	3.50
out-deg	-	-	1.41	3.20
in-deg	-	-	1.40	3.18
unw-cc	2.01	3.47	2.02	3.42
unw-pc	1.71	3.21	1.78	3.24
unw-pp	1.98	3.56	1.89	3.51

interacting components to be used in factored inference algorithms. Two basic views of the strength of interaction were used: the mutual information between variables measures their correlation, while the degree of separability is a property of variable's conditional distribution that guarantees the accuracy of its marginal. Computing the degree of separability exactly is probably intractable, and we gave an approximation algorithm along with a mechanism for trading off efficiency for accuracy. Examining heuristics based on both views of the strength of interaction, we find that the correlation between children that share parents in the DBN was the most effective proxy for approximation error in factored inference. Clustering based on the mutual information between children of common parents provides an efficient method for automatic decomposition of a DBN.

The approaches given here might be extended to find overlapping components, which have been found to entail less error in practice [BK98]. In addition, this work has focused primarily on pairwise interactions between variables, and we are exploring the use of higher-order information about the interactions of groups of variables – this may enable more effective use of the degree of separability, in particular.

References

[Ar72] S. Arimoto. An Algorithm for Calculating the Capacity of an Arbitrary Discrete Memoryless Channel. In *IEEE Trans. Inf. Theory*, IT-18:14-20, 1972.

[BK98] X. Boyen and D. Koller. Tractable Inference for Complex Stochastic Processes. In *Uncertainty in Artificial Intelligence*, 1998.

[DK89] T. Dean and K. Kanazawa. A Model for Reasoning about Persistence and Causation. In *Computational Intelligence*, 5:142-150, 1989.

[DW03] D. De Wolf. Generalized Derivatives of the Optimal Value of a Linear Program with respect to Matrix Coefficients. In *Quarterly Journal of the Belgian, French and Italian Operation Research Society*, 2003.

[Fr85] R. M. Freund. Postoptimal Analysis of a Linear Program Under Simultaneous Changes in the Matrix Coefficients. In *Mathematical Programming Study*, 24, 1985.

[Mu01] K. Murphy and Y. Weiss. The Factored Fronteir Algorithm for Approximate Inference in DBNs. In *Uncertainty in Artificial Intelligence*, 2001.

[Mu02] K. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning.* PhD thesis, U.C. Berkeley, Computer Science Division, 2002.

[NJW01] A. Ng, M. Jordan and Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. In *Neural Information Processing Systems (NIPS)*, 2001.

[NPP02] B. Ng, L. Peshkin and A. Pfeffer. Factored Particles for Scalable Monitoring. In *Uncertainty in Artificial Intelligence*, 2002.

[Pf01] A. Pfeffer. Sufficiency, Separability and Temporal Probabilistic Models. In *Uncertainty in Artificial Intelligence*, 2001.

[Pf06] A. Pfeffer. Approximate Separability for Weak Interaction in Dynamic Systems. In *Uncertainty in Artificial Intelligence*, 2006.