Journal of Theoretical and Applied Information Technology
10th June 2013. Vol. 52 No.1
© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PARTIAL DEPENDENT GRAPHICAL MODELS FOR UNDIRECTED RELATIONSHIPS INVOLVING PASSIVE STATES

DONGYU SHI

[1]Asstt Prof., Department of Computer Science and Technology, East China Normal University, China
E-mail: dyshi@cs.ecnu.edu.cn

ABSTRACT

As a core framework in artificial intelligence, standard graphical models generally assume that every state of a variable node has effect on other nodes by propagating beliefs to them. However, in many real world problems concerning undirected relationships, it is common that certain state of an object cannot possibly change other objects' states. Besides, an object only remains in this state when no effect on it successfully changes it. This paper defines it as passive state, and presents a framework - Partial Dependent Graphical Models (PDGM) for representing undirected relationships involving these states. Through a static undirected structure with a varying directed edge system, it analytically captures the behavior of variables having passive states. An inference method is developed for reasoning in the model. Experiments and examples are provided to show the effectiveness of PDGM.

Keywords: *Graphical Models, Uncertainty Representation, Probabilistic Dependence, Message Passing*

1. INTRODUCTION

Through graphical representation for distributions over multidimensional state spaces, probabilistic graphical models provide a general approach for reasoning under uncertainty [1]. They are used as tools for representing quantitative structured uncertainties and conditional independencies, playing a fundamental role in uncertainty reasoning in all kinds of intelligent applications. Standard models include Bayesian networks (BN), Markov networks (MN) [2], and factor graphs [3], for directed, undirected, and generalized form of dependencies respectively. They represent distributions compactly as normalized products of factors. Their graph structures encode relationships of conditional independence (CI) among different variable nodes.

Generally, any state of a variable in a standard graphical model (GM) is treated in the same way: it affects its neighbors by assigning distributions on their states. The assignment can be directly calculated from parameters concerning the state in a potential. However, in many real world problems, it is common that certain states are known that they do not affect other variables. The random variables in these states are the same as they are absent. While such relationships have been explored in BN, which known as Noisy-or models [2] and other extensions [4,5], there are little work concerning variables having such state in a more general dependencies.

This paper explores probabilistic relations involving states that have the same influence as the absence of variables. It shows that such dependencies exist in real world. It reviews why those developed models, such as MN, are difficult to capture these relations. It then presents a probabilistic framework with graphs to represent the relationships. Since exact inference is computationally hard, it develops a much more efficient inference method based on message passing for reasoning in the model.

2. BACKGROUND

BN and MN are two basic forms of graphical models. A BN is represented by a directed acyclic graph. Its joint distribution can be factorized as multiplications of every local conditional probability: $P(\boldsymbol{x}) = \prod_i P(x_i|pa(x_i))$. Noisy-OR models [2] are a kind of BN incorporating a special failure state. A binary parent node in this state implies that it fails to affect its child nodes, as if it is absent. Nodes in non-failure states affect their child through an OR gate. In case no parent succeeds to affect a child, a leak probability is assigned to the child [6]. Thus a local

Journal of Theoretical and Applied Information Technology
10th June 2013. Vol. 52 No.1
© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

relation can still be represented in a conditional probability table (CPT). Extensions of this model have been developed, allowing interactions such as AND or MAX. A MN $G(\boldsymbol{V}, \boldsymbol{E})$ is an undirected graph with set of vertices $\boldsymbol{V} = \{X_1, ..., X_n\}$ and set of undirected edges $\boldsymbol{E} \subset \boldsymbol{V} \times \boldsymbol{V}$, where each X_i is a random variable. The joint distribution is normalized multiplications of local potentials: $P(\boldsymbol{x}) = \alpha \prod_i \phi_i(\boldsymbol{x}_{\{i\}})$. Each potential ϕ_i is a function of a set $\boldsymbol{x}_{\{i\}}$ of the variable nodes in a clique, and α is the normalization function. In discrete state spaces, a potential is a table which assigns a nonnegative number to every state combination of variables in the corresponding clique. Among various developed GMs, context-specific independent (CSI) models can explore different dependence under different contexts [13-15]. Edges in graphs can be deleted in specific context of certain nodes. CSI in factor graphs can be represented by a graphical notation of Gates [16]. In stead of cutting edges, it uses a selector variable to disable factors.

A fundamental inference mechanism in GM is belief propagation (BP) [2] or its equivalent form of sum-product algorithm. It can be expressed as two phases of message passing. The message a variable sent to a local potential is: $\mu_{x\to f}(x) = \prod_{h\in n(x)\setminus\{f\}} \mu_{h\to x}(x)$. The message a local potential sent to a variable is: $\mu_{f\to x}(x) = \sum_{\sim\{x\}}(f(X)\prod_{y\in n(f)\setminus\{x\}} \mu_{y\to f}(y))$ [3,7]. It yields exact results in finite steps if the (moral) graph is acyclic, and is an update rule of iterated inference algorithm if there are loops. Although not guaranteed to converge, theoretical [8-10] and empirical [11,12] studies have shown the effectiveness of loopy belief propagation (LBP).

3. PROBLEMS WITH EXAMPLES

As an example, suppose the fire states of rooms in a building are to be estimated. The only observation is whether certain room is on fire or not. Each room can be represented by a variable with 2 states: $Fire$ and $NoFire$. Clearly if a room is watched on fire, the probability that its neighboring rooms are also on fire would rise. Furthermore, a room's state is independent of its non-adjacent rooms given all of its adjacent rooms' states. With these observations of direct dependence and CI, it is natural to represent it as a GM, and estimate a room's state through marginal distribution inference. The problem left is which model to use.

Since forcing fixed directionality between rooms is unreasonable, BN is not a proper model for it. Is MN proper? Suppose there's a potential function between two adjacent rooms, as in Table 1. If only these two are considered, when one is in state $Fire$, the other can be set a probability ($p/(1+p)$) to be in the same state. Considering symmetry, and making the potential in a canonical form, 3 of the 4 entries can be fixed. However, setting any number to entry $\phi(NoFire, NoFire)$ would encounter problems.

Table 1: potential of adjacent rooms.

	B=Fire	B=NoFire
A=Fire	p	1
A=NoFire	1	q

Taking the real line part of Figure 1 as an example, if the entry is set a number $q = 3$, then when room A is observed not on fire, by computation its neighbor B would have 25% to be on fire. Even worse, since B has a non-zero probability on fire, C's probability on fire would be higher than B as long as $p > 1/q$. As this goes on, the result would be at least 25% of all the other rooms are on fire. That is far from facts. If it is set a very large number, only leaving a near-zero "leak probability" to be on fire as an approximation, then when there are rooms observed on fire, all the other rooms except their direct neighbors would have near-zero probability to be on fire by computation. The model would then lose its meaning. Similar results would be got if a singlet potential is set as a prior on every unobserved node, as the dash line part in figure 1. Low prior on $Fire$ lowers the effect of any evidence on fire. Other techniques are possibly effective for fitting specific observation data in specific structure approximately, such as setting p a large number, making some potentials asymmetric, or adding extra nodes to the structure. But when the observation or the structure differs, these techniques are easily ended in failure. The experiment in Section 7 clearly shows it. Basically, the standard framework of undirected GM neglect the fact that $NoFire$ is not a state that can change its neighbors' states. Therefore, it is not a proper model for this problem, not even an approximation.

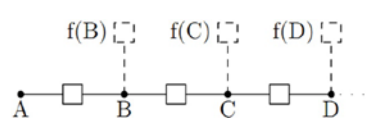


Figure 1: rooms modeled by MN

Journal of Theoretical and Applied Information Technology
10th June 2013. Vol. 52 No.1
© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

It seems that $NoFire$ is like a CSI state. Can CSI models capture its feature? According to CSI's definition [15], two sets of variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ are contextually independent given context $\boldsymbol{C}=c$, where $\boldsymbol{C}$ is also a set of variables and $c \in dom(\boldsymbol{C})$, if $P(\boldsymbol{X}|\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{c})=P(\boldsymbol{X}|\boldsymbol{c})$, for all $\boldsymbol{y} \in dom(\boldsymbol{Y})$, such that $P(\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{c})>0$. In the above example, the state $NoFire$ makes itself have no effect on other nodes. If it is a CSI model, the fact that the context $\boldsymbol{C}$ is either $\boldsymbol{X}$ or $\boldsymbol{Y}$ would make the equation trivial. It always holds without capturing the feature of the relation. Furthermore, if it is not an evidence, a room in $NoFire$ does not mean it is independent of other rooms. It still can be affected by them if they are not in $NoFire$. Either cutting edges or disable factors would block possible effects from them. In addition, a room in $NoFire$ would change to $Fire$ if any possible effect is successful to do so. It only remains on $NoFire$ *when all effects fail to change it*. This is similar to the failure state in a Noisy-OR model. For example, fire in a room can cause an alarm beeping. When there is no fire, the beeping state is predetermined by a leak probability. If there are multiple causes to make the device beeping, each cause affects it on their own, which known as the Independence of Causal Influence (ICI). When all parents of a child node are in their failure states, it can be viewed as they sending a belief message revealing the child's leak probability. Thus every state combination can be incorporated in a CPT, making it a BN. Based on it, a child node in failure state can also send messages to its parents *under the assumption of a prior distribution of its parents' states*. Usually this prior is a uniform. This is totally different from the situation of undirected relation.

States like $NoFire$ in "Fire Estimation" are common in real world. Epidemic propagation modeling normally has state "uninfected". Social influence modeling in a social network always encounters people "unaffected". Biological system modeling usually has molecules or cells that have "inactive" state. It's a state like "sleeping", having no effect and waiting to be activated. Proper models should be established to precisely represent the relationship involving these kind of states.

4. PASSIVE STATES

Comparing to standard GM, the major difference of modeling comes from the existence of the special state. It has no effect on other variables. A variable in this state is passive. It leaves the state if any effect can activate it.

Definition 1 *Suppose there is a probabilistic model on a set of random variables* $\boldsymbol{V}$, $|\boldsymbol{V}|\geq 2$. *Each variable* $X \in \boldsymbol{V}$ *has state space* $\{x^1,...,x^N\}$, $N \geq 2$. $b(X)$ *denotes* X*'s distribution. An effect* m *on* X *is in the form of a distribution on* X*'s states:* $m(x^1),...,m(x^N)$, *where* $\sum_{i=1}^N m(x^i)=1$. *Then the* ***passive state*** *of* X, *denoted by* $x^o, o \in \{1,...,N\}$, *is the state that satisfies:*

(1) *For any* $\boldsymbol{W} \subseteq \boldsymbol{V} \backslash X$, *if its distribution can be predetermined by* $P(\boldsymbol{W})$ *without considering* X, *then* $P(\boldsymbol{W}|x^o)=P(\boldsymbol{W})$. *Otherwise,* $P(\boldsymbol{W}|x^o)$ *is undetermined.*

(2) *For multiple effects* $\boldsymbol{M}=\{m_1,...,m_n\}$ *on* X, X*'s distribution under* $\boldsymbol{M}$ *is denoted by* $b(X)=\boldsymbol{M}(X)$. *For any* $m_j \in \boldsymbol{M}$, *with probability* $m_j(x^o)$, *effect* m_j *fails to have effect on* X, $b(X)=(\boldsymbol{M}\backslash m_j)(X)$.

A variable that has a passive state is called a *passive variable*. If it is in this state, it is *passive*. By contrast, other states are called *active states*. If a variable is not in passive state, it is *active*. The set of all states other than x^o is denoted by x^{-o}. $\boldsymbol{x}^o$ denotes that every variable in set $\boldsymbol{X}$ is in passive state, while $\boldsymbol{x}^{-o}$ denotes none of them in passive state. A variable that only has active states is called an *active variable*. We only concern variables that have at most one passive state. Multiple passive states cannot be distinguished in a probabilistic model, and should be combined to one state.

Corollary 1 *Suppose* $\boldsymbol{X},\boldsymbol{C} \subseteq \boldsymbol{V}$ *are two finite non-empty sets of possibly passive random variables, then* $\forall C \in \boldsymbol{C}$, *when* $C=c^o$:

$$P(\boldsymbol{X}|\boldsymbol{C})=P(\boldsymbol{X}|(\boldsymbol{C}\backslash C)) \quad (1)$$

where $P(\boldsymbol{X}|(\boldsymbol{C}\backslash C))$ *is a conditional distribution without considering* C.

Proof: According to definition 1(1), if $C=c^o$,

© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\begin{aligned} P(\boldsymbol{X}|\boldsymbol{C}) &= P(\boldsymbol{X}|(\boldsymbol{C}\backslash C), c^o) \\ &= P(\boldsymbol{X}, (\boldsymbol{C}\backslash C)|c^o)P((\boldsymbol{C}\backslash C)|c^o) \\ &= P(\boldsymbol{X}, (\boldsymbol{C}\backslash C))P(\boldsymbol{C}\backslash C) \\ &= P(\boldsymbol{X}|(\boldsymbol{C}\backslash C)) \end{aligned}$$

Corollary 2 *For a set of effects $m_1, ..., m_n$ on passive variable X, nonempty set $\boldsymbol{r} \subseteq \boldsymbol{n} = \{1, ..., n\}$ denotes any distinct subset of $\boldsymbol{n}$, X's distribution under these effects can be computed as:*

$$b(X) = \begin{cases} \sum_{\boldsymbol{r}\subseteq\boldsymbol{n}} \left(\alpha_{\boldsymbol{r}} \prod_{i\in\boldsymbol{r}} (m_i(x^{-o}) m_i(X)) \right. \\ \left. \cdot \prod_{j\notin\boldsymbol{r}} m_j(x^o)\right) & X \neq x^o \\ \prod_{i\in\boldsymbol{n}} m_i(x^o) & X = x^o \end{cases}$$

where $\alpha_{\boldsymbol{r}} = 1/\sum_{X\neq x^o} \prod_{i\in\boldsymbol{r}} m_i(X)$.

Proof: Let $\boldsymbol{M_r} = \{m_j, j \in \boldsymbol{r}\}$. For any subset r, with probability $\prod_{i\in\boldsymbol{r}} m_i(x^{-o}) \cdot \prod_{j\notin\boldsymbol{r}} m_j(x^o)$, each effect labeled in $\boldsymbol{r}$ is successful to make X in an active state, while each effect labeled in $\boldsymbol{n}-\boldsymbol{r}$ fails to activate X. By definition 1(2), $b(X) = \boldsymbol{M_r}(X)$, and effects in $\boldsymbol{M_r}$ only set X in its active states. Therefore, with this probability, $b(X) = \alpha_{\boldsymbol{r}} \prod_{i\in\boldsymbol{r}} m_i(X)$ $\forall X \neq x^o$, and $b(x^o) = 0$. The sum of the product of each $\boldsymbol{M_r}(X)$ with its probability is the first case of the equation. It only leaves $X = x^o$ with probability $\prod_{i\in\boldsymbol{n}} m_i(x^o)$. Thus Corol.2 is proved.

It can be viewed as the active part of each effect affecting X through an OR gate. Specifically, if X is binary, $b(x^{-o}) = \sum_{\boldsymbol{r}\subseteq\boldsymbol{n}} (\prod_{i\in\boldsymbol{r}} m_i(x^{-o}) \prod_{j\notin\boldsymbol{r}} m_j(x^o))$. Since $\boldsymbol{r}$ is nonempty, it equals to $1 - \prod_{i\in\boldsymbol{n}} m_i(x^o)$. Then this is exactly an OR gate on each $m_i(x^{-o})$, and its computation is rather efficient. If X has 2 or more active states, every possible situation - which effects "succeeds" (in set $\boldsymbol{r}$), and others "fails" (in set $\boldsymbol{n}-\boldsymbol{r}$), is computed separately by their probabilities. Then the overall number of different situations is $2^n - 1$, which makes it inefficient when n is large. If X has no passive state, by this corollary's computation, its each state's distribution is $\alpha_{\boldsymbol{n}} \prod_{i\in\boldsymbol{n}} m_i(x)$, which is consistent to BP in a standard model. With corol.1, conditional probability can be computed, and CI can be verified. Based on it, the static structure of related variables in a GM can be established. Consider a probabilistic model on a set of possibly passive variables $\boldsymbol{V}$, $|\boldsymbol{V}| \geq 2$. For three mutually disjoint sets $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{C} \subseteq \boldsymbol{V}$, if $P(\boldsymbol{X}|\boldsymbol{Y}, \boldsymbol{C}) = P(\boldsymbol{X}|\boldsymbol{C})$ holds for *any possible values*, then $\boldsymbol{X}$ and $\boldsymbol{Y}$ are *conditionally independent* given $\boldsymbol{C}$.

Definition 2 *Two variables X and Y have a **neighboring relation** in a graphical model on a set of variables $\boldsymbol{V}$, if there is no $\boldsymbol{C} \subset \boldsymbol{V}$, $\boldsymbol{C} \cap \{X, Y\} = \emptyset$ that makes X and Y conditionally independent. The two variables are each other's **neighbor**.*

Two neighbors are connected by a *static link*. The overall static links on $\boldsymbol{V}$ form the model's structure. Here we only consider models that any variable in it is conditionally independent of other variables given its neighbors.

5. PARTIAL DEPENDENCE

Usually, an edge in a GM represents a direct dependence that both sides have effect on each other. However, in a model that has passive variables, a static link does not imply the two connected variable nodes can always affect each other. According to definition 1, if one is in passive state, it cannot change the other's state. In this sense, the dependence between them is partial.

Definition 3 *If a neighboring relation involves at least one passive random variable, it is called a **partial dependence (PI)**.*

Describing PI in a proper form of parameters requires thorough analysis. It is known that parameters of discrete active variables are in the form of potential table. Each entry of the table corresponds to a state combination of variables in a clique, with a nonnegative real number. An example is shown in figure 2(a). The effect of one variable's state on the other variable's distribution is computed through each entry's proportion to all entries corresponding to the state. In a pair-wise model, the distribution computing on every entry is bi-directional. However, if there is a passive state, it has to be unidirectional - from active state to passive state, since passive state cannot affect other variables. Figure 2(b) and 2(c) show two examples. Furthermore, the entry corresponding to all variables in their passive states has to be left empty, since every side has no effect on others. Figure 2(b) is an example.

Journal of Theoretical and Applied Information Technology
10th June 2013. Vol. 52 No.1
© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	b^1	b^2	b^3
a^1	e_1	e_2	e_3
a^2	e_4	e_5	e_6

(a) active-active

	b^1	b^2	b^o
a^1	e_1	e_2	e_3^*
a^o	e_4^*	e_5^*	

(b) passive-passive

	b^1	b^2	b^o
a^1	e_1	e_2	e_3^*
a^2	e_4	e_5	e_6^*

(c) active-passive

Figure 2: Parameters between different types of variables. Entries with "" are unidirectional*

Definition 4 *A* ***partial potential*** *(p-pot) on a clique* $\boldsymbol{C}$ *, where every variable in* $\boldsymbol{C}$ *has neighboring relation with every other variable, is a function table denoted by* $\varphi(\boldsymbol{C})$*. For every* $X \in \boldsymbol{C}$*, it maps every combination of active states in* $\boldsymbol{C} - \{X\}$*, to a proportion on* X*'s states, which implies:*

$$\varphi(\boldsymbol{C})P((\boldsymbol{C}\backslash X)^{-o}) \propto P(X) \qquad (2)$$

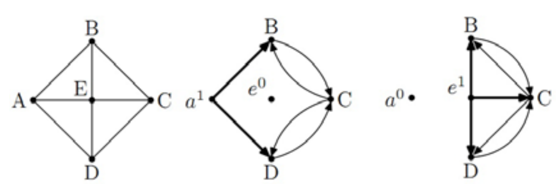
An entry of the table is a nonnegative real number corresponding to a state combination of every variable in the clique. If an entry's correspondent state combination includes at least one passive state, it is called a *partial entry*. If there exists an all-passive state combination, its correspondent entry is undefined and left empty. A distribution can only be computed under condition of active states. The proportion of a partial entry to all entries that have the same active states is interpreted as: in this proportion, the nodes in those active states *fail to activate* the node in passive state.

In a PI relation, node A having effect on node B does not mean at the same time B having effect on A. Plus, an effect only possibly holds when it comes from active states. Neither standard edges nor static links can explicitly represent such a relation. Therefore, *effect edges* are designed as an assistant system to exhibit partial dependence.

Definition 5 *An* ***effect edge*** *is a probabilistic existed directed edge between two neighboring nodes. The probability it exists is the probability that its tail node has effect on its head node. It is shown if the probability is not zero.*

An effect edge only shows one-side effect in a relation. Effect edge system can be cyclic, and differs under different observations. The nodes observed in passive states have no outgoing effect edges. The nodes observed in active states or the nodes only have active states have outgoing effect edges to each of their unobserved neighbors with probability 1, which is exhibited in thick real line in a graph. Effect edges from unobserved passive variables exist with probability less than 1, which is exhibited in thin line if the probability is not 0. The example in figure 3 shows the different effect edges under different observations of the same static structure in figure 3(a). Figure 3(b) shows the effect edges when node A is observed in active state a^1, and node E in passive state e^0. A has effect on its two neighbors with probability 1 , while E has no effect on other nodes. Both B and D have effect on C with probability less than 1, and vice versa. Figure 3(c) shows the effect edges when the observation is A in passive state and E in active state. To be consistent to standard GM, an effect on a variable node through an effect edge, in the form of the node's distribution, is called a *belief message* arrived at the node.

Since the existence of an effect from a passive variable has a probability, it is always possible that with certain probability, a node surrounded by passive nodes is affected by none of them. The node has to be estimated a *prior distribution*. This prior distribution may not be uniform. It is estimated by specific prior knowledge. For example, in "Fire Estimation", the prior on $NoFire$ should be 1 or slightly lower if a room is not supposed on fire when none of its neighbor is considered affecting it. This is a fact for most real world rooms. If it is 1, then every node's default distribution is their prior, for the non-effect property of passive state. There might be a question on how a fire is originated in such a model. The answer is intervention: the first burning room is usually for some reason outside of the model which intends to reveal the correlations of the rooms. If it is less than 1, that means every node in default has a probability in active state to affect their neighbors. In this case, when there is no observation, their default distribution in the model usually has a lower estimation on their passive states than their priors, since they are probable to affect each other through interaction.



(a) static structure (b) effect edges under a^1, e^0 (c) effect edges under a^0, e^1

Figure 3: The static link structure and effect edges with A and E are observed

6. PARTIAL DEPENDENT GM

© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A *Partial Dependent Graphical Model* (PDGM) is a graphical representation of a set of probabilistically related random variables, each of which possibly having a passive state. The nodes of the graph correspond to the variables. The static links of the graph reveal their direct dependencies. Each local structure has a partial potential as its parameters. As an assistant system, the effect edges varying by different observations explicitly show the variable nodes' effects on their neighbors. PDGM can induce independencies and factor-izations, as the standard GM. However, they have significant differences. The establishment of PDGM is not based on its variables' joint distribution, and a fixed joint distribution may not reveal relations among its objects. A joint probability table for discrete variables set every state combination a fixed probability, which means a fixed frequency. In many cases, frequency is not related to correlation. For example, for buildings in London, frequency of fire is quite different during peaceful time and during riot time. But the correlations of Fire effect between the same buildings, which mainly decided by their materials, structures, positions, etc are never change. On the contrary, a typical example of PDGM is the model that every variable's prior distribution is 1 on its passive state and 0 on other states. Its joint distribution by computation would be 1 on the combination of all passive states, and 0 on any other state combinations. Although the model exactly reveals the relationships of the nodes, its joint distribution tells nothing about them. In a model where variables have nonzero prior on their active states, the joint distribution is not trivial. In many cases, while variables' prior distributions are different, the relations among them do not change. In this sense, PDGM focuses on describing probabilistic correlations among its objects.

Although joint distribution cannot be compactly represented in PDGM, since the model is established on neighboring relations, there is at least one way to exactly compute it. It requires to compute each condition for each state combination of the variables. Taking the model in figure 3(a) as an example, first every state combination is considered separately. Then consider whether an active state is a result of its prior. And then consider whether each variable in active state which not activated by its own prior can be affected by one of its neighbors. For example, for state combination $state(ABCDE) = (11110)$, if only A is activated by its prior - $bypri(ABCD) = (1000)$, then the relations among them is exactly as figure 3 (b) shows. By separately computing the probability of each 3 possibilities:(1) B, D are activated by A, while C is activated by B, D; (2) B is activated by A, C is activated by B, and D is activated by C; (3) D is activated by A, C is activated by D, and B is activated by C, the probability of this specific situation can be obtained. Joint distribution can be obtained by summing up the probability of all possible situations in different state combinations. This is a method of exhaustive searching, which is inefficient.

6.1 Inference in Pair-wise Models

The inference is based on belief message passing. For two passive nodes X and Y linked together in a pair-wise model, the p-pot is represented as $\varphi(X,Y)$. Suppose $q(X)$ and $q(Y)$ are X and Y's distribution respectively without considering each other's effect. Let E denote effect edge $Y \to X$, $P(E) = q(y^{-o})$. Then after considering Y's effect on X, X's distribution can be computed by:

1. In case $Y \neq y^o$, which has probability $P(E)$, Y's effect on X is $m' = \alpha' q(Y^{-o})\varphi(X,Y)$, where $\alpha' = 1/\sum_X q(Y^{-o})\varphi(X,Y)$. By Corollary 2:

$$b(X) = \begin{cases} \alpha m'(x^{-o})m'(X)q(x^{-o})q(X) & \\ +q(x^o)m'(X) + m'(x^o)q(X) & X \neq x^o \\ m'(x^o)q(x^o) & X = x^o \end{cases}$$

where $\alpha = 1/\sum_{X \neq x^o} m'(X)q(X)$.

2. In case $Y = y^o$, which has probability $1 - P(E)$, $b(X) = q(X)$.

Altogether, $b(X)$ is:

$$\begin{cases} P(E)[(\alpha m'(x^{-o})q(x^{-o})q(X) + q(x^o))m'(X) & \\ +m'(x^o)q(X)] + (1-P(E))q(X) & X \neq x^o \\ m'(x^o)q(x^o) + (1-P(E))q(x^o) & X = x^o \end{cases}$$

Note that this result is equivalent to the effect $m(X)$ computed by:

$$m(X) = \begin{cases} m'(X)P(E) & X \neq x^o \\ m'(x^o) + (1-P(E)) & X = x^o \end{cases} \quad (3)$$

Thus, the "final" effect of any node on its neighbor can be obtained.

When computing Y's effect on X, the effect from X itself has to be excluded. Otherwise it would lead to infinite self-enhancement without extra information. This is analogous to the sum-product algorithm - the computation of a belief message from a node on an edge excludes the belief message arrived at this node from the edge. Thus the probability that an effect edge $Y \to X$ exists is the probability that all except X's effects on Y that

© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

make Y in its active states. For a complete inference, consider unobserved node X_v with n neighbors: $N_1, ..., N_n$, and n corresponding partial potentials $\varphi(X_v, N_1), ..., \varphi(X_v, N_n)$. X_v has an effect edge E_{vi} to each N_i, and each N_i has E_{iv} to X_v. Suppose X_v's prior distribution is $m_{0v}(X_v)$, and let $\boldsymbol{n} = \{0, 1, ..., n\}$. Suppose each normalized belief message from N_i to X_v is $m_{iv}(X_v)$. According to the above analysis, we have:

- The incoming messages sent to X_v excluding $m_{iv}(X_v)$, together with X_v' prior, on the belief of X_v in its active states is the probability of effect edge E_{vi}. It can be computed by corollary 2:

$$P(E_{vi}) = P_{\boldsymbol{n}\setminus i}(X_v \neq x_v^o) = 1 - \prod_{j \in \boldsymbol{n}\setminus i} m_{jv}(x_v^o) \quad (4)$$

- With probability $P(E_{vi})$, X_v has effect on N_i. The distribution on its active states can be viewed as a message sent to $\varphi(X_v, N_i)$:

$$\mu_{v\to\varphi} = \sum_{X_v \neq x_v^o, \boldsymbol{r} \subseteq \boldsymbol{n}\setminus i} \Big[\prod_{i' \in \boldsymbol{r}} \left(m_{i'v}(x_v^{-o}) m_{i'v}(X_v)\right) \cdot \prod_{j' \in \boldsymbol{n}\setminus i - \boldsymbol{r}} m_{j'v}(x_v^o)\Big] \cdot \alpha_{\boldsymbol{r}}$$

(5)

And the effect on N_i is:

$$m'_{vi}(N_i) = \alpha' \varphi(X_v, N_i) \mu_{v\to\varphi} \quad (6)$$

Including the part of passive state (with probability $1 - P(E_{vi})$), the altogether effect on N_i, which also can be viewed as a message sent from $\varphi(X_v, N_i)$ to N_i, is computed by:

$$\mu_{\varphi\to i} = m_{vi}(N_i) = \begin{cases} m'_{vi}(N_i) P(E_{vi}) & N_i \neq n_i^o \\ m'_{vi}(N_i) + (1 - P(E_{vi})) & N_i = n_i^o \end{cases} \quad (7)$$

Thus, each message sent from a node to its neighbor can be computed by the messages it receives from its neighbors. The equations above can be used to update each belief message. Finally, the belief of X_v can be directly computed by all messages it receives through corollary 2.

Note that the number of unknown message m are equal to the number of their equations. In theory, they can be obtained by solving these equations. However, solving high order equations is infeasible. Therefore, the following iteration algorithm - *Belief and Effect Updating* (BEU) is presented to get approximate result:

1. Each node observed in active state and each active variable establishes an effect edge to each of its unobserved neighbors with probability 1;
2. Each unobserved node establishes an effect edge to each of its unobserved neighbors with the probability of its summarized prior on all active states;
3. Each node sends out initial belief messages according to their observations or priors, to their unobserved neighbors through effect edges;
4. If a node receives a belief message from a neighbor, it establishes an effect edge to every other unobserved neighbors if the edge has not been established before, and updates the probability of the edge along with the belief message on it according to equation(4), (5) and (7). Repeat this process until they converge;
5. According to the above messages sent to each unobserved node, compute each node's belief.

The algorithm is based on BP. If the static structure is a tree, or if the static links among unobserved nodes form trees, then each evidence through different effect edge to a node on the tree, or each prior of a node on a tree can only possibly update any belief message on any effect edge at most once, which makes BEU to get an exact result in finite steps. However, if the unobserved nodes form loops, it is not guaranteed to converge, like LBP. Specifically, if there is no passive state, by equation(4), $P(E_{vi}) = 1$ for every effect. Then $m_{vi}(N_i) = \alpha' \varphi(X_v, N_i) \alpha_{\boldsymbol{n}} \prod_{i' \in \boldsymbol{n}\setminus i} m_{i'v}(X_v)$. Aside from an extra normalization which does not change any result, it is the same as LBP in a pair-wise Markov network.

6.2 Discussion of Non-Pairwise Models

In the above analysis and computations, a p-pot is not required for a node if its corresponding neighbor is in passive state. For a node in a clique with 3 or more nodes, it requires different p-pots when different neighbor in the clique is in passive state. A natural way to deal with this problem is keeping a complete p-pot table for the clique, and reducing it whenever necessary. For p-pot $\varphi(\boldsymbol{C})$ on clique $\boldsymbol{C}$, the variables in a subset $\boldsymbol{s} \subset \boldsymbol{C}$ are in their passive states with a probability. The reduced p-pot $\varphi(\boldsymbol{C})_{\boldsymbol{s}^o}$ keeps the $\varphi(\boldsymbol{C})$ entries where $\boldsymbol{s} = \boldsymbol{s}^o$, without variables in $\boldsymbol{s}$. The nodes in the clique will then compute their belief messages by the reduced p-pot. An example of reduced p-pots for a 3-node clique is shown in figure 4.

Journal of Theoretical and Applied Information Technology
10th June 2013. Vol. 52 No.1
© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A B C	φ
1 1 1	e_1
1 1 0	e_2
1 0 1	e_3
1 0 0	e_4
0 1 1	e_5
0 1 0	e_6
0 0 1	e_7

A B	φ_{c^0}	B C	φ_{a^0}	A C	φ_{b^0}
1 1	e_2	1 1	e_5	1 1	e_3
1 0	e_4	1 0	e_6	1 0	e_4
0 1	e_6	0 1	e_7	0 1	e_7

Figure 4: A p-pot with its reduced forms for a 3-node clique, 0 is the passive state

However, reduced p-pots cannot be arbitrarily parameterized, otherwise they would not be compatible in the complete p-pot. Thus their parameters are constrained. If such constraint is not reasonable, then a set of p-pots should be maintained for a clique. For arbitrary parameters, it requires one p-pot for each sub-clique. Each one of them only has entries that at most one variable is in passive state. An example of arbitrary parameterized p-pot set for a 3-node clique is shown in figure 5. In this case it can even be reduced to one node, putting the node's prior in the corresponding p-pot. For an overall computation, each possible situation, including which variables in a clique are active, and should send messages to which p-pot in the p-pot set, is computed separately by their probabilities. The computation of $\mu_{X\to\varphi}$ is the same as in Equation(5), computing the effect of all messages except the one from this clique. The computation of $\mu_{\varphi\to X}$ requires different φ for different cases of variables in their passive states.

A B C	φ_1
1 1 1	e_1
1 1 0	e_2
1 0 1	e_3
0 1 1	e_4

A B	φ_2	B C	φ_3	A C	φ_4
1 1	e_5	1 1	e_8	1 1	e_{11}
1 0	e_6	1 0	e_9	1 0	e_{12}
0 1	e_7	0 1	e_{10}	0 1	e_{13}

Figure 5: A set of arbitrary parameterized p-pots

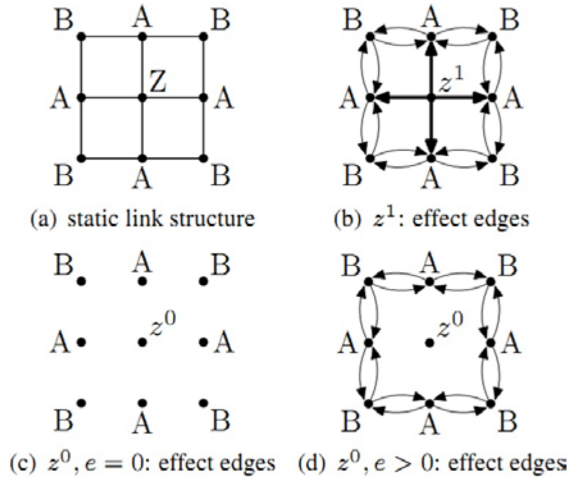


Figure 6: The structure in the experiment

7. EXPERIMENTS AND EXAMPLES

The following experiment compares PDGM with MN. It models "Fire Estimation" on a graph in figure 6(a), with each node representing a room that might be on fire. The center node Z is the only room observed, 1 ($Fire$) and 0 ($NoFire$) each time. Due to the symmetry of the graph, only beliefs of two kinds of nodes - A and B are required to be estimated. The potential table for MN modeling is table 1. The p-pot for PDGM is the same table without the parameter q. In PDGM modeling, $e = prior(Fire)$ is the prior on active state $Fire$. Figure 6(b), 6(c) and 6(d) are effect edges when Z is observed in active state, passive state with $prior(Fire) = 0$, and passive state with $prior(Fire) > 0$ separately. Note that in each modeling, exact results can be obtained by solving quadratic equations.

Table 2: Fire estimation by PDGM and MN.

		P=2		P=0.8		P=0.25	
		Z=1	Z=0	Z=1	Z=0	Z=1	Z=0
PDGM, e=0	A	0.8582	0	0.5486	0	0.2132	0
	B	0.7713	0	0.3945	0	0.0809	0
PDGM, e=0.001	A	0.8586	0.0050	0.5496	0.0026	0.2143	0.0015
	B	0.7717	0.0050	0.3957	0.0026	0.0822	0.0015
MN,q=max {1,1/p}	A	0.8638	0.7236	0.3386	0.3386	0.0154	0.0154
	B	0.7603	0.7236	0.3902	0.3902	0.0588	0.0588
MN,q=max {p,1+1/p}	A	0.7887	0.2113	0.1655	0.0911	0.0099	0.0080
	B	0.6511	0.3489	0.1987	0.1841	0.0386	0.0386
MN,q= 3(p+1/p)	A	0.1111	0.0025	0.0244	0.0044	0.0015	0.0005
	B	0.0588	0.0187	0.0304	0.0266	0.0062	0.0061
*MN,q=1 e=0.001	A	0.0020	0.0010	0.0008	0.0010	0.0002	0.0010
	B	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010
**MN, q=1		p=2/e		p=0.8/e		p=0.25/e	
	A	1.0000	0.9990	1.0000	0.0235	0.9982	0.0018

Journal of Theoretical and Applied Information Technology
10th June 2013. Vol. 52 No.1
© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

e=0.001	B	0.9998	0.9990	0.9984	0.0235	0.9816	0.0018

The result is shown in table 2. They are probabilities of $Fire$ in room A and B, by different parameters. The higher p means the rooms are less fireproof, while the lower p means better. Note that p can be less than 1 if the fireproof material is rather good. In MN modeling, setting q to be 1 or p are two normal ways to set it. But q should never be less than $1/p$ to make a fair comparison. In *MN and **MN modeling, each unobserved node is set a singlet potential as a prior, $e = prior(Fire)$. By general knowledge, when Z is on fire, the probability of A on fire should be no less than B on fire. Better fireproof rooms should have lower probability than worse fireproof rooms. When Z is $NoFire$, there is no reason to estimate high probability on $Fire$ for any room. Table 2 clearly shows that in any presented condition, only PDGM have reasonable modeling. MN modeling, either with or without prior, and no matter what q is set, cannot reasonably explain the situation. Note that **MN's strategy is setting p a large number associated with $1/e$. It can be a good approximation when there is only one observation, either in active or passive state, in a tree structure. But when there are more than 1 observations, or if the structure has loops, it fails to explain it. Only when p and q are both very large, which means the relation between every pair of nodes is very close to a deterministic relation (one room on fire, nearly every other rooms are also on fire), then the MN modeling can be close to PDGM.

The following example shows how BEU works, with structure in figure 7(b) and parameters of trinary nodes in figure 7(a). $pri(1,2,0)$ of every node is set to be $(0,0,1)$. Here, Node 3, 5 are observed in state 2 and 1 respectively, and node 8 is observed in state 0, as figure 7(b) shows. Figure 7(c) is the effect edges under the observation.

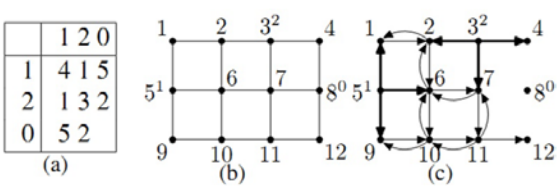


Figure 7: BEU test: parameters and structure

Table 3: The BEU iterations.

	1	2	3	4	5	6
$m_{2-1}(1)$	0.1667	0.2057	0.2189	0.2225	0.2248	0.2249
$m_{2-1}(2)$	0.2381	0.2396	0.2472	0.2466	0.2465	0.2465
$m_{6-2}(1)$	0.1848	0.2453	0.2612	0.2712	0.2713	0.2713
$m_{6-2}(2)$	0.0761	0.1313	0.1366	0.1413	0.1412	0.1412
$m_{6-7}(1)$	0.1848	0.2541	0.2692	0.2744	0.2745	0.2745
$m_{6-7}(2)$	0.0761	0.1321	0.1366	0.1402	0.1402	0.1402
$m_{6-10}(1)$	0.1848	0.2835	0.2835	0.2871	0.2871	0.2871
$m_{6-10}(2)$	0.0761	0.1699	0.1699	0.1695	0.1695	0.1695
$m_{7-6}(1)$	0.1667	0.1667	0.1881	0.1881	0.1883	0.1883
$m_{7-6}(2)$	0.2381	0.2381	0.2430	0.2430	0.2430	0.2430
$m_{7-11}(1)$	0.1667	0.2209	0.2244	0.2256	0.2256	0.2256
$m_{7-11}(2)$	0.2381	0.2463	0.2455	0.2457	0.2457	0.2457
$m_{10-6}(1)$	0	0.0923	0.1229	0.1231	0.1233	0.1233
$m_{10-6}(2)$	0	0.0468	0.0741	0.0739	0.0740	0.0740
$m_{10-9}(1)$	0	0.1534	0.1869	0.1889	0.1890	0.1890
$m_{10-9}(2)$	0	0.0933	0.1175	0.1177	0.1177	0.1177
$m_{10-11}(1)$	0	0.1899	0.1899	0.1917	0.1917	0.1917
$m_{10-11}(2)$	0	0.1041	0.1041	0.1045	0.1045	0.1045
$m_{11-7}(1)$	0	0.1006	0.1006	0.1014	0.1014	0.1014
$m_{11-7}(2)$	0	0.0585	0.0585	0.0586	0.0586	0.0586
$m_{11-10}(1)$	0	0.1432	0.1445	0.1451	0.1451	0.1451
$m_{11-10}(2)$	0	0.1216	0.1215	0.1216	0.1216	0.1216
$m_{11-12}(1)$	0	0.1843	0.1858	0.1870	0.1870	0.1870
$m_{11-12}(2)$	0	0.1336	0.1338	0.1342	0.1342	0.1342

The iteration results are shown in table 3, where each $m_{i-j}(k)$ is the belief message on state k (1 or 2) that sent on effect edge E_{i-j}. Note that there is no effect edge sent from or to node 8. Node 12 has no effect edge to 11, because it has no incoming edges other than E_{11-12}. With prior of 0 on active states, it has no effect on node 11. The belief messages sent on $3 \to 2$, $3 \to 4$, $3 \to 7$, $5 \to 1$, $5 \to 6$ and $5 \to 9$ are never change. Edge $1 \to 2$ is only affected by $5 \to 1$. Edge $2 \to 6$ is only affected by $1 \to 2$ and $3 \to 2$. Thus the belief messages they send are fixed in 1 and 2 steps of updating respectively. The rest of the edges are shown in table 3. They converge after 6 iterations. The marginal distributions of the unobserved nodes are in table 4. The nodes having more incoming effect edges tend to have low probability on their passive states, since they have more ways to be activated.

Table 4: Marginal distributions of unobserved nodes.

node	1	2	0	node	1	2	0
1	0.5354	0.2577	0.2069	2	0.4792	0.4466	0.0742
4	0.1667	0.5000	0.3333	6	0.6200	0.3333	0.0467
7	0.4011	0.4937	0.1052	9	0.5737	0.1812	0.2451
10	0.6260	0.2548	0.1192	11	0.4561	0.3346	0.2093
12	0.2978	0.2137	0.4885				

8. CONCLUSION

© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The contribution of this paper can be viewed as threefold. First, it reminds the existence of passive states, and gives a definition in probabilistic models. In real world, it is common that in an interactive relationship, an object has a passive state, in which it cannot change other objects' states. Therefore, modeling dependencies among these objects is different with normal undirected modeling. Corollaries are deduced to describe the feature of passive variables.

Second, this paper presents the Partial Dependent Graphical Models, a framework that can precisely capture the probabilistic relationships among objects that have passive states. It is made up of two correlated parts. The undirected structure with partial potentials reveals the static relations among different variable nodes. The assistant effect edge system manifests different interactions among variable nodes under certain observations. Aside from representing partial dependence, PDGM is consistent to standard graphical models.

Third, an inference method based on message passing is provided for reasoning in the model, since directly computing joint distribution is inefficient. It yields exact results when the unobserved nodes in a model form tree structure. If there are loops among unobserved nodes, the Belief and Effect Updating process is provided for obtaining approximate results. Experiments and examples show the effectiveness of the model.

A series of issues are left for future research. One is developing efficient method for computing joint distribution. Joint probability distribution of certain set of nodes may be required in some applications. Computing by exhaustive possibility searching is extremely time consuming. Efficient approximate algorithms are required to be developed. Learning parameters in PDGM is another issue. Since a partial potential has both normal and partial entries as its parameters, and computing joint distribution is not easy, developing learning algorithms can be a challenge. Other issues may include exploring it in chain graph like structures, and extending it to dynamic modeling.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (Grant No. 61003181, 61175116).

REFRENCES:

[1] D. Koller and N. Friedman, *Probabilistic Graphical Models*, MIT Press, 2009.

[2] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan and Kaufmann, 1988.

[3] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum product algorithm", *IEEE Transactions on Information Theory*, Vol. 42, No. 2, 2001, pp. 498-519.

[4] D. Heckerman, "Causal independence for knowledge acquisition and inference", *Proceedings of Ninth Conference on Uncertainty in Artificial Intelligence*, 1993.

[5] F. J. Diez, "Parameter adjustment in Bayes networks. the generalized noisy or gate", *Proceedings of Ninth Conference on Uncertainty in Artificial Intelligence*, 1993.

[6] M. Henrion, "Some practical issues in constructing belief networks", *Proceedings of Third Conference on Uncertainty in Artificial Intelligence*, 1987.

[7] B. J. Frey, "Extending factor graphs so as to unify directed and undirected graphical models", *Proceedings of Nineteenth Conference on Uncertainty in Artificial Intelligence*, 2003.

[8] Y. Weiss, "Correctness of local probability propagation in graphical models with loops", *Neural Computation*, Vol. 12, 2000, pp. 1-41.

[9] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in gaussian graphical models of arbitrary topology", *Neural Computation*, Vol. 13, 2001, pp. 2173-2200.

[10] J. Yedidia, W. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms", *IEEE Transactions on Information Theory*, Vol. 51, 2005, pp. 2282-2312.

[11] K. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: an empirical study", *Proceedings of Fifteenth Conference on Uncertainty in Artificial Intelligence*, 1999.

[12] Y. Weiss, "Comparing the mean field method and belief propagation for approximate inference in MRFs", *Advanced Mean Field Methods*, MIT Press, 2001.

[13] C. Boutiler, N. Friedman, M. Goldszmidt, and D. Koller, "Context-specific independence in Bayesian networks", *Proceedings of Twelfth Conference on Uncertainty in Artificial Intelligence*, 1996.

© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[14] D. Geiger and D. Heckerman, "Knowledge representation and inference in similarity networks and Bayesian multinets", *Artificial Intelligence*, Vol. 82, No 1-2, 1996, pp. 45-74.

[15] D. Poole and N. Zhang, "Exploiting contextual independence in probabilistic inference", *Journal of Artficial Intelligence Research*, Vol. 18, 2003, pp. 263-313.

[16] T. Minka and J. Winn, "Gates", *Advances in Neural Information Processing Systems*, 2008.