CMPUT 566: Introduction to Machine Learning
A Self-Study Report on: Probabilistic Graphical Models (PGMs)

Islam A. Ali
Student ID: 1633813
iaali@ualberta.ca

Abstract

Due to the success achieved by machine learning algorithms in a wide range of applications, the need to have models that can accommodate complex tasks and being able to infer results in an efficient way became a necessity. These complex tasks come equipped with an extended number of variables and a complex network of relations that can results in inefficient modeling in case of using conventional learning and inference models. Probabilistic Graphical Models (PGMs) represents an efficient method for dealing with such complex tasks by combining concepts from probability theory with graph theory. They depend on modeling random variables and the dependencies among them in the form of either a directional acyclic graph (in case of Bayesian Networks) or as a indirected graphs (in case of Markov Networks). The main advantage of such models is the utilization of the conditional independence property that results in an efficient structure for inference. In this report, we present a summary of PGMs and their usage in machine learning tasks with a focus on the Bayesian Networks. The report starts by giving a brief intro about the background needed for this topic. Then, The representation of PGMs, specially Bayesian Networks is presented in detail. Following that, the concepts of learning in Bayesian Networks as well as inference is discussed. The report concluded by mentioning some of the application of PGMs such as expert systems, and provided a brief discussion PGMs pros and cons.

1 What are Probabilistic Graphical Models (PGMs)?

Probabilistic Graphical Models (PGMs) are statistical model that can be used to represent complex dependencies between random variables efficiently using graphs data structure. This dependencies are modeled as joint distributions. However, with the increase of the number of random variables involved, the need to have a more efficient way of representation is needed. The graph representation provide efficiency in terms of time complexity as well as scale-ability to more complex systems as it models random variables and the conditional dependencies among them allowing for scale-ability in terms of problems studied.

2 Motivation

To motivate the idea of having a more complexity-efficient statistical model for dependencies representation, an example of industrial defects in products diagnosis system is discussed. Let's say a product in a production line can either be malfunctioned electrically or mechanically or both. This can be represented by a joint distribution of two random variables where each is representing a certain malfunction. Further, we can expand the diagnosis system to the reason for defect. For instance in the electrical defects, it is because of PCBs manufacturing or due to components defects, on the other hand, the mechanical malfunction can be either from the material used or the methodology utilized. And the relations goes on, with extended number of random variables, that can be very expensive to represent or to query using regular mathematical methods. In this case, the rise of the PGMs makes sense, where such a system can be represented as a graph with nodes representing each random variable in the system, and with directed edges representing the dependencies and their directions.

Other motivation comes from the graph theory itself, where a wide range of research was done and was proven to be correct for data dependency representation, query, and manipulation. A wide range of efficient algorithms are available for all these purposes and even beyond.

3 Mathematical Background

In this section, we briefly review the main concepts that will be used throughout this report. For this purpose, a review of both probability theory and the graph theory are reviewed briefly by mentioning their most famous/important rules and concepts. This quick review intentions is to act as a reminder and not an extensive review of theories.

3.1 Probability Theory

Probability theory's main objective is to systematically study uncertainty or the degree of confidence in quantities or measurements. The following are the three main axioms of probability:

1. **Sample Space:** It is the set of all possible outcomes of a certain experiments, it is denoted by Ω.
2. **Event Space:** It is a subset of the sample space and represents a limited set of possible outcomes of the experiment, it is denoted by F.
3. **Probability Measure:** It is a measure that maps the event space to real numbers, it is denoted by P, and it satisfies a number of rules:
   - $P(A) \geq 0$
   - $P(\Omega) = 1$
   - The union of disjoint events is the summation of their probabilities.

3.1.1 Conditional Probability and Independence

The conditional probability is the probability of the occurrence of an event after observing another event. This is given by the following formula:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{1}$$

Independence, on the other hand, is mainly defined as the first event having no impact on the second event and can be mathematically represented as:

$$P(A \cap B) = P(A)P(B) \tag{2}$$

3.1.2 Results of Conditional Probability: Chain Rule and Bayes Rule

Based on the conditional probability, we can deduce the following relation that is called the chain rule:

$$P(A \cap B) = P(A|B)P(B) \tag{3}$$

More generally, if a series of events $A_1, A_2, ..., A_n$, the probability of a certain event can be given by:

$$P(A_1 \cap ... \cap A_n) = P(A_1)P(A_1|A_2)...P(A_1|A_2 \cap ... \cap A_n) \tag{4}$$

Another result of the conditional probability, is the Bayes Rule which allows to compute a certain conditional probability from its inverse with some knowledge of probability priors. The mathematical representation is defined as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{5}$$

3.1.3 Random Variables, Marginal, and Joint Distributions

A random variable is defined by a function that assigns a value for all possible outcome of a certain measurable experiment. It is a very useful when dealing with complex and dependent relations of events where each has its own probabilistic properties.

One of the associated concepts to random variables is marginal distribution. The marginal distribution of random variable X is defined as the distribution over the events in which X is involved, and is denoted by $P(X)$. Another concept associated with random variable is joint distributions, which is defined as the distribution over events in which two or more random variables are involved, and is denoted by $P(X, Y)$. It can also be mathematically represented by:

$$P_{XY}(x, y) = P_{XY}(y|x)P_Y(x) \tag{6}$$

3.1.4 Mean/Expectation and Variance

Two other concepts related to probability theory are mean and variance which can be used to describe a probability distribution. The mathematical formula for the expectation in both the discrete and continuous cases are given by:

$$\mathbb{E}[X] = \sum_x x.P(x) \tag{7}$$

$$\mathbb{E}[X] = \int_x x.P(x)\ dx \tag{8}$$

While the variance is defined as:

$$\mathbb{V}ar[X] = \mathbb{E}[X^2] - \mathbb{E}^2[X] \tag{9}$$

3.2 Graph Theory

A graph is a data structure that is structured from two elements that are: nodes and edges. It is mainly used to represented relations between generic data elements via connections among nodes via edges. Graphs can have multiple structure and can be categorized based on many aspects such as:

1. What nodes represent?
2. Is it a directed or undirected graph?
3. Is it a fully or partially directed graph?
4. Are the edges weighted or not?
5. Are cycles and loops allowed or not?
6. What are the traversal techniques allowed?

In the context of this report, nodes represent random variables while edges represent the conditional relationship between random variables. Edges in this context can either be directed or undirected, and can allow loops and cycles. The following is an example of a directed graph [1]:

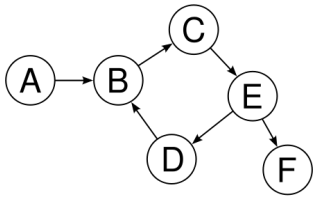

Figure 1: Example of Directed Graph

[1] Image was downloaded from: `https://computersciencewiki.org/index.php/The_web_as_a_directed_grap`

4 PGMs Representation

PGMs has many representations such as *Bayesian Networks* where edges are directed and are indicated by an arrow in the graphical representation. Another type is the *Markov Random Fields* in which edges are undirected and are represented by lines. The first method is useful for conditional dependency representation, while later one is useful in representing soft constraints. In this report, we focus on directed probabilistic graphical models, known as **"Bayesian Networks"**.

4.1 Bayesian Networks Representation

Bayesian Networks, a.k.a Bayesian Belief Networks (BBNs), are PGMs that represent the conditional relations between random variables by directed acyclic graph (DAG).
Formally, a Bayesian Network BN is represented by:

$$BN = (g, \{PX_1, ..., PX_N\}) \tag{10}$$

Where: g is a directed acyclic graph (DAG) represented as $g = (X, E)$. From the definition of the DAG, X are the graph nodes representing random variables, while E are the edges connecting nodes to each other representing conditional independence between random variables. $PX_1, ..., PX_N$ are the conditional probability distributions associated with each node/random variable. It is represented as a table indicating the relations between ancestors of a certain node and the resulting probability for each outcome of the same node. In the following diagrams we present the probability distribution table associated with a certain node in the BN context.

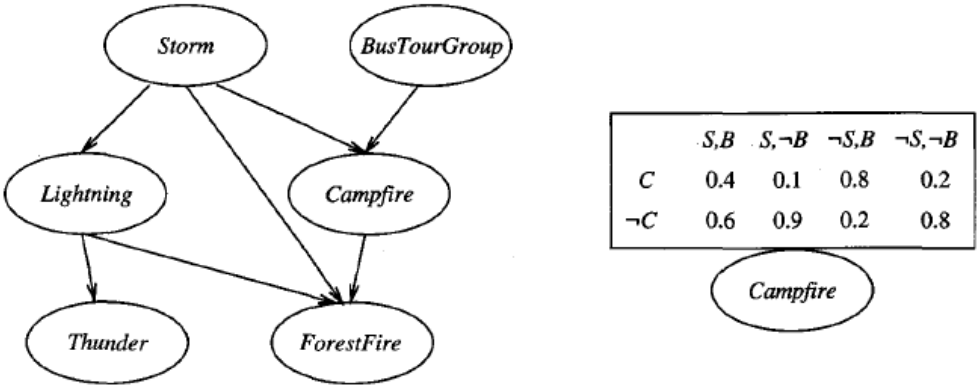

Figure 2: BN with Conditional Probability Table

For the sake of formulation, we also define the parameter $\Theta = \{\theta_1, ..., \theta_k\}$, which are the parameters which define the probability distribution for each node. For instance, in case of a Gaussian distribution, for the node X_i, the set of parameters defining it are given by:

$$\Theta_i = (\mu_i, \sigma_i) \tag{11}$$

Where: μ_i and σ_i^2 are the mean and variance of the Gaussian distribution describing node X_i. Accordingly, the BN can be defined in terms of the parameters defining the probability distributions as follows:

$$BN = (g, \{\Theta_1, ..., \Theta_N\}) \tag{12}$$

It worth mentioning that in some of the references, the parameter Θ refers to the weights and bias for each probability distribution rather than the mean and variance or the parameters defining the distribution. The definition of weight and bias and their impact on the distributions are not provided, unless they refer to the weight and bias governing the mean definition, which will then make $\Theta = (w_i, b_i, \sigma_i)$.

4.1.1 Factorization Properties

The factorization property allow for defining the joint distribution as a multiplication of conditional probabilities governed by the conditional independence implied in the graph structure. Each of the conditional probability distributions are denoted by:

$$P_{X_i} = (X_i|P_g(Xi)) \tag{13}$$

Which means that the probability of a certain random variable is defined by the conditional probability relative to its parents/ancestors in the DAG.
Based on the previous relation and the definition of a joint probability given in the introduction, the complete joint probability of the BN $P_{BN}(X_i, ..., X_N)$ (Where N is the number of random variable in the BN), is given by:

$$P_{BN}(X_i, ..., X_N) = \prod_{i=1}^{N}(X_i|Parent_g(Xi)) \tag{14}$$

Applying this rule on the example BN given in Fig. 2, we can end up with the following joint distribution (note that for each node, the initials are used as random variable names. e.g. Storm –¿ S).

$$P_{BN}(X_i, ..., X_N) = P(S).P(B).P(L|S).P(C|S, B).P(T|L).P(F|L, S, C) \tag{15}$$

Based on the joint distribution factorization, one can easily proof that conditional probability in case of parameters independence. For this, we seek to get the conditional probability of a certain node given all other node available in the graph. For example, we seek to get $P(C|S, B, L, T, F)$ in the previously mentioned example in Fig. 2. According to the relation between conditional probability and the factorization principle illustrated, the conditional probability can be re-written as:

$$P(C|S, B, L, T, F) = \frac{P(C, S, B, L, T, F)}{P(S, B, L, T, F)} \tag{16}$$

The joint distributions can be further expanded as:

$$P(C, S, B, L, T, F) = P(S)P(B)P(L|S)P(C|S, B)P(T|L)P(F|L, S, C) \tag{17}$$

$$P(S, B, L, T, F) = \sum_{C} P(C, S, B, L, T, F) \tag{18}$$

$$P(S, B, L, T, F) = \sum_{C} P(S)P(B)P(L|S)P(C|S, B)P(T|L)P(F|L, S, C) \tag{19}$$

$$P(S, B, L, T, F) = P(S)P(B)P(L|S)(\sum_{C} P(C|S, B))P(T|L)P(F|L, S, C) \tag{20}$$

The summation over all values of a random variable =1, from the basic probability axioms, which imply:

$$P(S, B, L, T, F) = P(S)P(B)P(L|S)P(T|L)P(F|L, S, C) \tag{21}$$

Therefore, the conditional probability can be re-written as:

$$P(C|S, B, L, T, F) = \frac{P(S)P(B)P(L|S)P(C|S, B)P(T|L)P(F|L, S, C)}{P(S)P(B)P(L|S)P(T|L)P(F|L, S, C)} \tag{22}$$

After canceling out equal terms in the fraction, we end up with:

$$P(C|S, B, L, T, F) = P(C|S, B) \tag{23}$$

Which imply the independence of the node of any non-parent node.

4.1.2 Conditional Independence Property

Conditional independence is a very important property when dealing with BNs that can substantially simplify the structure of the joint distributions and can provide a concise presentation of it when factorized. As suggested by the literature, conditional independence is considered the backbone of BNs due to the face that the structure of the graph and the complexity reduction advantage are mainly basing on this property.
Consider the case when we have a BN with three nodes, where nodes c-¿a and c-¿b are connected. The independence of random variable a and b are coming from the definition of the c and its relation to both random variables. The following is the definition of conditional independence:

$$P(a, b|c) = P(a|c)P(b|C) \tag{24}$$

In this case, a and b are declared conditionally independent given c and this can be denoted by the perpendicular sign $\perp$ such that:

$$P(a \perp b|c) = P(a|c) \tag{25}$$

$$P(b \perp a|c) = P(b|c) \tag{26}$$

This definition stems from the definition of the joint distribution of independent random variables given by (in case of a and b begin independent):

$$P(a, b) = P(a)P(b) \tag{27}$$

Alternatively, the definition of conditional independence can be given by:

$$P(a|b, c) = P(a|c) \tag{28}$$

The proof for this relation comes from the joint distribution definition and its relation to the conditional probability definition, and it goes as follows:

$$P(a|b, c) = \frac{P(a, b, c)}{P(b, c)} \tag{29}$$

$$P(a|b, c) = \frac{P(a, b|c)P(c)}{P(b|c)P(c)} \tag{30}$$

From the first definition of conditional independence:

$$P(a|b, c) = \frac{P(a|c)P(b|c)P(c)}{P(b|c)P(c)} \tag{31}$$

Canceling out equivalent terms:

$$P(a|b, c) = P(a|c) \tag{32}$$

To show the impact of the conditional independence on the complexity reduction, we consider the following network structure. Assume we have $n = 20$ binary random variables, with a structure having each node in the network to have a maximum of 4 parent nodes. The number of probability values needed to be calculated in order to decide on an outcome in case of having no conditional independence assumption is given by:

$$count(P)_{total} = 2^n = 2^{20} \tag{33}$$

While in case of utilizing the reduction from the conditional independence the number goes down to:

$$count(P)_{total} = n * (2^k) = 20 * (2^4) = 20 * 16 = 320 \approxeq 2^9 \tag{34}$$

Which is a huge improvement in terms of complexity.

4.1.3 Proof for The Conditional Independence Property

In order to prove the conditional independence property, we define three types of nodes with respect to a certain X_i node:

1. Descendant Node: Which is one of the children of X_i, either directly or non-directly.
2. Parent Node: Which are nodes having X_i as one of their descendants.
3. Non Descendant Node: Which are node that are neither descendants nor parents.

Intuitively, a node have dependency on descent node, then the following relation is sufficient to prove the conditional dependency:

$$P(X_i|non-desc(X_i), Parent(X_i)) = P(X_i|Parent(X_i)) \tag{35}$$

Again, using the joint probability and the relation to conditional probability, we get:

$$P(X_i|non-desc(X_i), Parent(X_i)) = \frac{P(X_i, non-desc(X_i), Parent(X_i))}{P(non-desc(X_i)P(Parent(Xi)))} \tag{36}$$

$$numerator = \sum_{desc(X_i)} P(X_i, non-desc(X_i), Parent(X_i), desc(X_i)) \tag{37}$$

Which can be factorized based on each node type as follows:

$$= \sum_{desc(X_i)} (P(X_i|Parent(X_i))) * (\prod_{X_j \in desc(X_i)} P(X_j|Parent(X_j))) * (\prod_{X_k \in non-desc(X_i) \bigcup parent(X_i)} P(X_k|Parent(X_k))) \tag{38}$$

Given the fact that the summation over all values of desc = 1 from the prob. axioms, the numerator can be reduced to:

$$= (P(X_i|Parent(X_i))) \sum_{desc(X_i)} \quad (\prod_{X_k \in non-desc(X_i) \bigcup parent(X_i)} P(X_k|Parent(X_k))) \tag{39}$$

Processing the denominator will be as follows:

$$denominator = \sum_{desc(X_i)} \quad (\prod_{X_k \in non-desc(X_i) \bigcup parent(X_i)} P(X_k|Parent(X_k))) \tag{40}$$

by dividing the above two equations, we get the final formula as:

$$P(X_i|non-desc(X_i), Parent(X_i)) = P(X_i|Parent(X_i)) \tag{41}$$

Which proves the conditional independence property.

5 Learning in PGMs

After illustrating the representation of BNs, and the significance of their usage in machine learning problems, the learning process of this model is now discussed. The objective of the learning process it to have a BN that we can use to get the probability distribution of a random variable over all possible values of this variable. In this context we do have three cases as follows:

1. Both BN structure and conditional probabilities are known, and in this case, the inference process is straight forward.
2. Only the structure of the BN is known, and the conditional probabilities are not defined. In this case, The learning process would be to define the probability distribution tables associated with each random variable. In this stated that, domain experts may define the optimal structure of the BN but the training data may not have enough observability over internal random variables.

3. Both the structure and the conditional probabilities are not known and the objective of the training process is to determine both the optimal structure and the conditional distributions given the training data.

In all cases, the learning process will depend on the the specific task and the available knowledge of the system and whether the domain experts are available for providing the optimal structure of the BN. Additionally, it worth mentioning that having a full automated model that can learn both the structure of the network and the probability distributions and the conditional relations is the optimal case and the ultimate objective in learning PGMs.

5.1 Learning The Conditional Distribution Tables of BNs

In the case where the structure of the network is defined, meaning that for each random variable/node in the BN X_i; the parents $Parent(X_i)$ and descendants $Desc(X_i)$ of this random variables X_i are defined and the conditional independences are known. In this case, we assume partial observability of conditional distribution tables associated with each node. In this section, we discuss the Gradient Ascent Algorithm that can be used to learn the entries of the tables given the structure and partial observability. From an abstract point of view, the algorithm depends on searching in hypotheses H and tries to maximize $P(D|h)$, where D is the available training data, and h is one hypothesis in the hypothesis space. More formally, we need to define the following terms:

- $P(D|h)$: is the probability of the training data given a certain hypothesis. It is also denoted by $P_h(D)$ for simplicity.
- $d \in D$: is a data sample inside the training data, the training data D is of size M.
- w_{ijk}: is an entry in one of the conditional distribution tables associated with each random variable Y_i.
- Y_i: is the random variable in hand to which the table is being calculated.
- U_i: is the set of parent nodes for the random variable Y_i.
- y_{ij}: is one value of possible values for random variable Y_i.
- u_{ik}: is one value of possible values for parent U_k or random variable Y_i.

The objective is to maximize $P(D|h)$ for each of the tables entries for each node in the network. The maximization is achieved by calculating the gradient at each w_ijk for the $lnP_h(D)$. The objective is to proof the following relation:

$$\frac{\partial lnP(D|h)}{\partial w_{ijk}} = \sum_{d \in D} \frac{P(Y_i = y_{ij}, U_i = u_{ik}|d)}{w_{ijk}} \tag{42}$$

Assuming the data $d \in D$ is iid:

$$\frac{\partial lnP(D|h)}{\partial w_{ijk}} = \frac{\partial}{w_{ijk}} ln \prod_{d \in D} P_h(d) \tag{43}$$

$$\frac{\partial lnP(D|h)}{\partial w_{ijk}} = \sum_{d \in D} \frac{\partial lnP_h(D)}{\partial w_{ijk}} \tag{44}$$

The partial derivative of ln is given by:

$$\frac{\partial lnf(x)}{\partial x} = \frac{1}{f(x)} . \frac{\partial f(x)}{\partial x} \tag{45}$$

Then, the equation will be:

$$\frac{\partial lnP(D|h)}{\partial w_{ijk}} = \sum_{d \in D} \frac{1}{P_h(D)} . \frac{\partial P_h(D)}{\partial w_{ijk}} \tag{46}$$

In order to determine the gradient, $P(D|h)$ needs to be represented using y_{ij} and u_{ik} as follows:

$$\frac{\partial lnP(D|h)}{\partial w_{ijk}} = \sum_{d \in D} \frac{1}{P_h(D)} . \frac{\partial}{\partial w_{ijk}} \sum_{j',k'} P_h(d|y_{ij'}, u_{ik'}) P_h(y_{ij'}, u_{ik'}) \tag{47}$$

Following the product rule, we can further expand the equation as follows:

$$\frac{\partial lnP(D|h)}{\partial w_{ijk}} = \sum_{d \in D} \frac{1}{P_h(D)} . \frac{\partial}{\partial w_{ijk}} \sum_{j',k'} P_h(d|y_{ij'}, u_{ik'}) P_h(y_{ij'}|u_{ik'}) P_h(u_{ik'}) \tag{48}$$

The summation is done over $j^{'}$ and $k^{'}$ as they represent all entries in the network, however, for the value to be non-zero, the following conditions must hold:

$$j^{'} = j \qquad k^{'} = k \tag{49}$$

Using this constraint over the equation, the equation can be reduced to the following form:

$$\frac{\partial lnP(D|h)}{\partial w_{ijk}} = \sum_{d \in D} \frac{1}{P_h(D)} . \frac{\partial}{\partial w_{ijk}} P_h(d|y_{ij}, u_{ik}) P_h(y_{ij}|u_{ik}) P_h(u_{ik}) \tag{50}$$

The entry w_{ijk} is essentially the prob. of the random variable given its parents, and can be re-written as:

$$w_{ijk} = P_h(y_{ij}|u_{ik}) \tag{51}$$

Therefore, the gradient can be expressed as:

$$\frac{\partial lnP(D|h)}{\partial w_{ijk}} = \sum_{d \in D} \frac{1}{P_h(D)} . \frac{\partial}{\partial w_{ijk}} P_h(d|y_{ij}, u_{ik}) w_{ijk} P_h(u_{ik}) \tag{52}$$

And the result of the partial derivative will result in excluding this term as:

$$\frac{\partial lnP(D|h)}{\partial w_{ijk}} = \sum_{d \in D} \frac{1}{P_h(D)} . P_h(d|y_{ij}, u_{ik}) P_h(u_{ik}) \tag{53}$$

The conditional prob. $P_h(d|y_{ij}, u_{ik})$ can be re-written by the Bayes rule as:

$$P_h(d|y_{ij}, u_{ik}) = \frac{P_h(y_{ij}, u_{ik}|d) P_h(d)}{P_h(y_{ij}, u_{ik})} \tag{54}$$

Plugging this relation into the gradient equation, we end up with the following relation:

$$\frac{\partial lnP(D|h)}{\partial w_{ijk}} = \sum_{d \in D} \frac{1}{P_h(D)} \frac{P_h(y_{ij}, u_{ik}|d) P_h(d)}{P_h(y_{ij}, u_{ik})} P_h(u_{ik}) \tag{55}$$

Canceling out $P_h(D)$:

$$\frac{\partial lnP(D|h)}{\partial w_{ijk}} = \sum_{d \in D} \frac{P_h(y_{ij}, u_{ik}|d)}{P_h(y_{ij}, u_{ik})} P_h(u_{ik}) \tag{56}$$

Using the relation between the joint probability and the conditional prob. we can further reduce the equation into:

$$\frac{\partial lnP(D|h)}{\partial w_{ijk}} = \sum_{d \in D} \frac{P_h(y_{ij}, u_{ik}|d)}{P_h(y_{ij}|u_{ik})} \tag{57}$$

Using the definition of $w_{ijk} = P_h(y_{ij}|u_{ik})$, we can re-write the gradient to be:

$$\frac{\partial lnP(D|h)}{\partial w_{ijk}} = \sum_{d \in D} \frac{P_h(y_{ij}, u_{ik}|d)}{w_{ijk}} \tag{58}$$

Which is the objective we started trying to prove.

5.1.1 Gradient Ascent Algorithm

For the learning to happen, we can use a gradient ascent algorithm that can be formed as:

$$w_{ijk} = w_{ijk} + \alpha \sum_{d \in D} \frac{P_h(y_{ij}, u_{ik}|d)}{w_{ijk}} \tag{59}$$

Where α is the learning rate and is considered a hyperparameter. The main constraint that must be respected is that the values will need to be normalized in order to have all conditional probabilities to = 1 for a certain given random variable Y_i, that's to satisfy the probabilities axioms mentioned earlier in this report.

5.2 Learning the Structure of the BN

The problem of deducing the structure of the BN as well as the conditional probability tables associated with it, is considered a PN-Hard problem due to the possible configurations of networks given a certain dataset. One solution to this problem an algorithm called K2 algorithm, which is essentially doing heuristic search over alternative forms of the BN, but the algorithm assumes full observability of the training data and the relations. Other algorithms are also available, and they differ in terms of the needed observability in the training data and the trade-off made between accuracy and efficiency. Another aspect to look after in the context of learning the BN structure is the scoring function or the measure by which a certain structure can be judged compared to others.

6 Inference in PGMs

Inference is the process of inferring the probability distribution of a certain query variable given some observable variables in the system. The assumption here is that the structure of the network as well as the values of the conditional probability tables are all defined either before hand in the experiment setup or by learning as illustrated in the previous section. In this context we define the following sets:

- O: The set of observed random variables.
- Q: The set of query random variables.
- H: The set of random variables that are not one of the above.

Together they construct the set of random variable in BN such that:

$$X = O \cup Q \cup H \tag{60}$$

Another condition on these sets is that they do not intersect with each other:

$$O \cap Q = Q \cap H = O \cap H = \emptyset \tag{61}$$

6.1 Inference Queries in PGMs

Queries can have one of two types in the context of PGMS:

(1) Marginalization Queries: Where the marginal distribution of a query variable is required given the observed variables, this is given by:

$$P(Q|O = o) = \frac{P(Q, O = o)}{P(O = o)} \tag{62}$$

The joint distribution $P(Q, O = o)$ can be calculated by marginalization over the H variables as follows:

$$P(Q, O = o) = \sum_{h \in val(H)} P(O = o, Q, H = h) \tag{63}$$

While the prior can be deduced from marginalization over the Q variables as follows:

$$P(O = o) = \sum_{q \in val(Q)} P(O = o, Q = q) \tag{64}$$

(2) Maximum A Posteriori Queries: Where the most likely instantiation of a variable is required given some observation, and is given by:

$$P^* = maximize(q)\ P(Q = q|O = o) \tag{65}$$

Which is equivalent to the maximization of the joint distribution by marginalization over the variables of H. The above equation can then be rewritten to be:

$$P^* = maximize(q) \sum_{h \in val(H)} P(Q = q, O = o, H = h) \tag{66}$$

6.2 Exact vs. Approximate Inference

The compromise between the accuracy and efficiency of the inference step is determined based on the structure/complexity of the network and on the task itself. For instance, in real-time applications, with a loose need for best accuracy, approximate inference can be used. In some cases, exact inference is not even feasible with an increased number of variables and a complex dependencies. Variable elimination is one example of how an exact inference can be done. It replies on defining factors that contain multiple variable, then marginalizing over variables inside factors so inference results can be deduced. For approximate inference, variational methods are stated in some of the literature with not much of an illustration as the main method for approximate inference.

7 Applications of PGMs

PGMs are used to model uncertainty in systems and to represent conditional dependency between random variables. The application at which such a scheme can be used are countless. For instance, they can be used in speech recognition, computer vision, or expert systems. Focusing on expert systems, an expert system is defined as a system that contains a knowledge base and an inference engine, where the inference can be done based on the knowledge base. The knowledge base can evolve over time and the inference engine must take that into consideration when producing new information. BNs and more generally PGMs are used to model uncertainty in such systems. Expert systems are very popular in the medical field where diagnosis assistance is needed. For instance, inferring the probability of a certain disease given a observable symptoms is one form of such systems. Other areas, such as insurance, stock market, among many other utilize such systems for the sake of modeling uncertainty due to its efficiency and applicability.

8 Discussion

8.1 PGMs Pros and Cons

As outlined multiple times during the course of this report, the pros of PGMs are mainly in the efficient representation and the utilization of the conditional independence for inference. Also, the applicability for scalability that PGMs have due to the same point of efficiency. However, during my reading in the materials available, I couldn't find any discussion of the memory requirements for maintaining a huge model, because this can be a bottleneck in some application where such availability of memory is not valid. Another point to mention is related to the learning process in BNs, specifically in the part where we need to learn the structure of the network by observable data samples. In this process, and due to the heuristic search approach adopted in the space of difference structures, the optimality of the solution is not guaranteed which means that the resulting structure may not the optimal one that can maximize $P(D|h)$.

8.2 PGMs vs. Deep Learning

Due to the network-like structure of both, the differences between them needs to be highlighted in order to get more understanding for the applicability of using any of them given a certain task. I believe the main difference is the observability of internal variables and nodes in PGMs which is not valid in case of neural networks and deep learning as all layers except for the input and output layers are considered hidden layers. However, deep learning have the ability to memorize data and relations much better than PGMs due to its structure that does not depend on random variables available in the system only. Also, the power to change internal structure based on the data in NNs gives it another advantage over PGMs where the structure can be pre-defined by domain experts.

Conclusion

In this report, PGMs were discussed in details. The report started by providing detailed description of PGMs representation in the case of Bayesian Networks which represents random variables and the dependency relations among them in the form of acyclic directed graph (DAG). Then, learning PGMs was discussed as well, where we presented the three cases where learning is required which are: having both the structure and the conditional probability table as known information, or having only the structure, or having neither the structure nor the conditional probability tables. Then, the derivation of a gradient ascent algorithm for determining the conditional probability tables in case of known network structure was provided. After that, concepts for exact and approximate inference was discussed briefly. Finally, the report concluded by discussing potential applications of PGMs, the pros and cons of PGMs, and the similarities and differences between PGMs and deep learning models.

References

[1] A. Maleki and T. Do, "Cs229: Machine learning, lecture notes." Online Lecture Notes, Department of Computer Science, Stanford University, 2020.

[2] T. M. Mitchell *et al.*, "Machine learning," 1997.

[3] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[4] F. Pernkopf, R. Peharz, and S. Tschiatschek, "Introduction to probabilistic graphical models," in *Academic Press Library in Signal Processing*, vol. 1, pp. 989–1064, Elsevier, 2014.

[5] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[6] A. Kim, "Conditional independence — the backbone of bayesian networks." `https://towardsdatascience.com/conditional-independence-the-backbone-of-bayesian-networks-85710f1b35b`, Oct. 2019.

[7] S. Beretta, M. Castelli, I. Gonçalves, R. Henriques, and D. Ramazzotti, "Learning the structure of bayesian networks: A quantitative assessment of the effect of different algorithmic schemes," *Complexity*, vol. 2018, 2018.