



$$P(X | Y) = \frac{P(Y | X) P(X)}{P(Y)} \quad (2.6)$$

Bayes' theorem allows to convert an abductive reasoning task into a deductive one and vice versa. It plays a fundamental role in Bayesian networks, which model uncertain causal relationships among random variables. Applying Bayes' theorem allows to turn a causal model into a diagnostic reasoning task.

 **Inference** It is obviously straightforward to convert a propositional logic KB into a probabilistic KB by introducing a Boolean random variable for every atomic proposition in Σ . The joint probability distribution $P(\Sigma)$ then can be stored in a contingency table holding the co-occurrences of the respective atomic events. Making use of the chain rule, the law of total probability and Bayes' theorem, one can compute the posterior probability $P(Q | E)$ of any arbitrary query $Q \subseteq \Sigma$ given any arbitrary evidence $E \subseteq \Sigma$. Such a posterior belief can be computed by the canonical inference equation (cf. Jain, 2012)

$$\begin{aligned} P(Q = q | E = e) &= \frac{P(Q = q, E = e)}{P(E = e)} \\ &= \frac{\sum_{u \in \text{dom}(U)} P(Q = q, E = e, U = u)}{\sum_{q' \in \text{dom}(Q)} \sum_{u \in \text{dom}(U)} P(Q = q', E = e, U = u)} \end{aligned}$$

where $U = \Sigma \setminus Q \setminus E$ is the set of all hidden variables. However, as there must be an entry for every combination of atomic events, such a full joint probability table is hopelessly infeasible to represent for practical applications. Not only does the table grow exponentially in the number of variables, but, in order to be of statistical significance, the amount of data to be collected also grows exponentially.

 **Most Probable Explanation** Computing the posterior distribution over a set of query variables Q in the light of observed evidence variables E is only one kind of inference that is typically carried out in probabilistic KBs. Sometimes, however, one is not necessarily interested in the complete posterior distribution over the queries, but only in the *most probable* variable assignment given the evidence. In this kind of inference, also called most probable explanation (MPE) or maximum a-posteriori (MAP) inference, thus the most probable possible world \hat{x} is computed that is compatible with the observations E , i.e.

$$\hat{x} = \arg \max_{x \in \mathcal{X}} P(x | E).$$

As $P(x|E) \propto P(x, E)$, it is not required to compute a normalization constant, and it is often computationally more appealing to compute the MPE state of a posterior, when the complete distribution is not necessarily required. A very common class of such inferences are classification problems, where one is typically interested in assigning an entity the most probable class.

Learning There are, in principle, two ways for determining the quantitative specification of probabilities of specific random events: First, a knowledge engineer can enter manually probabilities into the model that expresses their belief in and knowledge about the domain of discourse. Such probabilities are called *subjective* probabilities. Another, much more frequent way of obtaining probabilities is to derive them from previously made observations. This special kind of inference, also called *inductive reasoning*, aims at determining the most suitable model parameters $\hat{\theta}$, in some model space Θ , such that the model best explains the observed data set. The training set \mathcal{D} comprises N examples of complete assignments of all variables under consideration, $\mathcal{D} = \{d_1, \dots, d_N\}$. It is typically assumed that the individual examples d_i represent samples that have been drawn independently of each other from the same ‘true’, problem-intrinsic distribution which is to be approximated by the learning procedure. This assumption is also called the *i.i.d.* (independent, identically distributed) assumption. Given the data at hand and variable model parameters θ , the *likelihood function* \mathcal{L} measures the congruence of the model defined by θ and the observations by applying the model under θ to \mathcal{D} , i.e.

$$\mathcal{L}(\theta | \mathcal{D}) = \prod_{i=1}^N P(d_i; \theta). \quad (2.7)$$

The problem of *parameter estimation* or *learning* is now to find the parameters $\hat{\theta}$ that maximize the likelihood function (2.7) with respect to θ ,

$$\hat{\theta}_{MLE} = \arg \max_{\theta \in \Theta} \mathcal{L}(\theta) = \arg \max_{\theta \in \Theta} \prod_{i=1}^N P(d_i; \theta),$$

to yield the parameters $\hat{\theta}_{MLE}$ that best explain the observations, the so-called maximum likelihood estimate (MLE). The maximum likelihood principle is the most fundamental and perhaps most commonly used technique to fit probabilistic models to observed training data. Often, it is more convenient to maximize the natural logarithm of the likelihood function. The so-called *log-likelihood* does have its maxima at



Maximum
Likelihood

same positions as the logarithm is a strictly monotonically increasing function. The log likelihood has the advantage that the logarithm of the product in (2.7) turns into a sum of the individual logarithms, which can be conveniently maximized when the model's functional form is chosen appropriately.

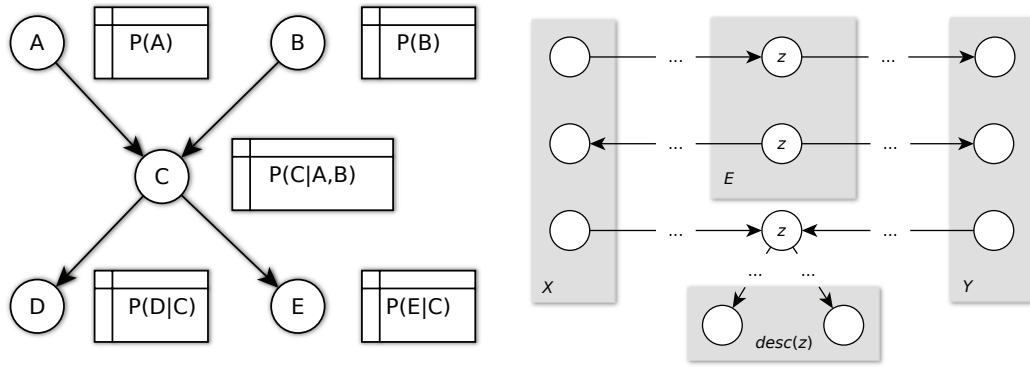
Complexity Both the problem of parameter learning and inference can be shown to be NP-complete (Russell and Norvig, 2003). Indeed, it can be shown that for many probabilistic models, the optimal model parameters can be obtained by combining model inference with a numeric optimization technique. It is therefore unlikely that there exist algorithms to compute the respective problems efficiently. Unfortunately, it seems that the only effective way to reduce the representational complexity of probabilistic KBs is rigorous exploitation of domain knowledge about independencies of variables (Koller and Friedman, 2009). Probabilistic graphical models (PGMs) are representation formalisms that make use of directed or undirected graph structures to represent the (in-)dependencies among random variables.

2.2.2 Bayesian Networks

In order to mitigate the computational complexity of probabilistic methods, Bayesian networks (BNs) provide a formalism to specify dependencies in terms of a causal relationships of variables. A Bayesian network (BN) consists of a directed, acyclic graph (DAG) structure, where there is one node for every random variable under consideration, and the (directed) edges among the nodes determine the dependency structure of the distribution. A node with an outgoing edge is called a *parent* of the node that the respective edge points at. Every node X_i with incoming edges directly depends on all of its parents, which are denoted by the set $Par(X_i)$. Every node has attached a distribution that determines the probability of the respective variable conditioned on all its direct parents. The joint distribution over all variables X_1, \dots, X_n is then given by

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | Par(X_i)) \quad (2.8)$$

The graph structure of a BN thus directly determines the factorization of the joint distribution. BNs have been successfully used for modeling causal relationships among variables, such as in expert systems for deductive and abductive reasoning in medical diagnosis (Uebersax, 2004) and risk management (Cárdenas et al., 2013),



(a) Exemplary Bayesian network with five nodes A, B, C, D, E and their conditional probability tables.

(b) Graphical representation of the three cases of path blocking in the d-separation criterion.

Figure 2.2: Exemplary Bayesian network structures

among many others.

Knowledge engineering with BNs appears particularly intuitive and natural since the graph structure directly determines the joint distribution as a normalized factorization of conditionals reflecting the graph structure. In addition, the principle of a causes-and-effects model allows very intuitive top-down construction of BNs.

An example of a BN is shown in Figure 2.2a. It consists of five random variables/nodes A, B, C, D and E , and each node has attached a conditional probability table conditioned on its direct parents storing the respective conditional distributions. According to (2.8), the distribution in turn factorizes as



Example

$$P(A, B, C, D, E) = P(D|C) \cdot P(E|C) \cdot P(C|A, B) \cdot P(B) \cdot P(A).$$

The structure of this exemplary BN is adapted from the well-known ‘alarm example’ from Russell and Norvig (2003). The random event C can have one of two causes A and B , which are independent of each other. The common effect C of A and B can itself trigger two possible events D and E . Although very simplistic, the example exposes a couple of very interesting and fundamental properties of BNs.

Parameter Estimation Parameter estimation boils down to counting relative frequencies of events in the training data and storing them in conditional probability tables, which can be shown to be the maximum likelihood estimate of the conditional probabilities. There are, however, a few subtleties that need to be taken into account when working with BNs, which I address in the following.

Conditional Independence The perhaps most subtle property of variables in a BN is the phenomenon of conditional dependence, i.e. two a-priori independent variables in a BN can become dependent given a third variable is observed. In the BN in Figure 2.2a, A and B are a-priori independent. If C is observed, however, they become dependent as knowledge about either of them might ‘explain away’ the respective other. This phenomenon of inter-causal inference makes determination of independence in BN structures particularly cumbersome.

The concept of *d-separation* defines a couple of criteria that enable to tell whether or not two sets of variables in a BN are conditionally independent by simple graph properties: Two sets of variables X and Y are conditionally independent given a set of observations E iff all (undirected) paths between X and Y are being blocked. A path between X and Y is being blocked by a node z on that path iff one of the following criteria applies:

1. there is a node z on the path with one incoming and one outgoing edge that is observed, i.e. $z \in E$,
2. there is a node z on the path with two outgoing edges that is observed, i.e. $z \in E$,
3. there is a node z on the path with two incoming edges and neither z nor any of its descendants $desc(z)$ are observed, i.e. $z \notin E$ and $desc(z) \cap E = \emptyset$.

The three criteria are depicted in Figure 2.2b. It can be shown that d-separation is equivalent to conditional independence. It remains, however, an unhandy and peculiar way of investigating dependencies in BNs.

Directedness and Acyclicity The directedness of edges conveys that dependencies are unilateral, i.e. one variable influences another variable but not vice versa. However, this is not possible in Bayesian probability theory: Let, w.l.o.g. be $P(X|Y) > P(X)$. Then, according to Bayes’ theorem, we have

$$P(X|Y) > P(X) \Leftrightarrow \frac{P(X,Y)}{P(Y)} > P(X) \Leftrightarrow \frac{P(Y,X)}{P(X)} > P(Y) \Leftrightarrow P(Y|X) > P(Y).$$

Consequently, (in-)dependence among any pair of random variables is always bilateral and the directedness of edges in a BN does not seem to have a natural correspondence in terms of probabilistic dependence.

Correlation and Causality In statistical learning and inductive reasoning, the use of causal models may be misleading and only serve better interpretability. Despite the

presence of many variables whose causations are undoubted, many others are very hard if not impossible to determine: For example, although it is commonly assumed that smoking can be a cause of lung cancer, a correlation between health and mood in humans is less evident: Does improved health lead to improved mood, or does good health lead to good mood? Or do both factors influence each other mutually? Creating a BN requires a knowledge engineer to decide on either direction of causality. If there is no or only little evidence for causality, it is hard to come up with resilient BN structures. In the worst case, unfortunately designed BN structures even leverage misinterpretations of correlations to exhibit causation. From a purely statistical point of view, for instance, it is not possible to tell if obese people tend to consume diet drinks in order to loose weight, or if the consumption of such drinks causes weight gains (“consuming diet drinks leads to obesity”).

The directed nature of knowledge bases encoded as BNs has consequences that make them practically less straightforwardly applicable as they, at first, may have seemed.

2.2.3 Markov Random Fields

As a trade-off between the generality of full joint distributions and strongly restricted BN structures, undirected graphical models are another family of probabilistic formalisms, whose semantics are more straightforward.

Undirected graphical models, also known as Markov random fields (MRFs) or Markov networks (MNs) are an alternative representation of probability distributions over complex structures of random variables. As opposed to BNs, MRFs use *undirected* graph structures for representing the dependency model of the variables. In an MRF, a pair of nodes in the network is connected by an undirected edge if the respective variables are conditionally dependent given all other nodes in the network. The factorization of the joint distribution over the variables $X = \langle X_1, \dots, X_n \rangle$ of the network G for a specific possible world x is defined as the Gibbs distribution

$$P(X = x) = \frac{1}{Z} \prod_{c \in G} \psi_c(x_{\{c\}}), \quad (2.9)$$

where the cs denote all *maximal cliques* in G , ψ_c denotes a *potential function* attached to c , and $x_{\{c\}}$ denotes a projection of the values of all variables in the specific world x that are part of c . The clique potentials ψ_c in an MRF constitute the quantitative representational part of a distribution. A potential maps every state of a clique to a non-negative real-valued measure that multiplicatively contributes to the overall

probability mass of a possible world:

$$\psi_c : X_{\{c\}} \mapsto \mathbb{R}^+$$

As the probability masses of all possible worlds may not sum to 1, it is required to normalize them by the so-called *partition function* Z ,

$$Z = \sum_{x' \in \mathcal{X}} \prod_{c \in G} \psi_c(x'_{\{c\}}),$$

in order to form a proper probability distribution. The perhaps most important observations are firstly, whereas in BNs, the individual factors in the joint distribution are local, normalized conditional distributions, in an MRF, the factors are (not necessarily normalized) clique potentials that contribute in some way to the overall probability of a possible world. Consequently, the numeric values of the potential functions in isolation do not have a direct probabilistic interpretation, but they only gain their probabilistic semantics when the contributions of *all* potentials are taken into account. Secondly, since the partition function sums over the probability masses of all possible worlds, exact inference in MRFs is intractable for all but the smallest problem instances. Therefore, approximate algorithms are typically applied in parameter learning and inference.

Independence As opposed to BNs, independence of sets of variables in MRFs can be determined by simple graph separation instead of d -separation. Two sets of variables X and Y of an MRF G are conditionally independent of each other given a third set of variables Z (with X , Y and Z being pairwise disjoint) if the removal of Z from G would separate X and Y , i.e. the removal of Z would remove any connection between X and Y . Figure 2.3 shows the exemplary probabilistic model from Figure 2.2a as an

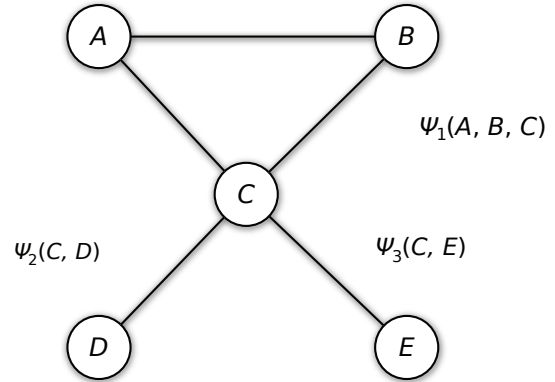


Figure 2.3: Exemplary Markov network with three maximal cliques $\{A, B, C\}$, $\{C, D\}$ and $\{C, E\}$ and their clique potentials ψ_1 , ψ_2 and ψ_3 .

MRF retaining the conditional (in)dependencies among the variables. The MRF comprises the same nodes A, B, C, D , and E . There are three maximal cliques in the network, one ternary connecting A, B , and C , and two binary connecting C and D , and C and E , respectively. Hence, there must be three clique potential functions

$\Psi_1 : A \times B \times C \mapsto \mathbb{R}$, $\Psi_2 : C \times D \mapsto \mathbb{R}$, and $\Psi_3 : C \times E \mapsto \mathbb{R}$. Note that, in order to retain the conditional dependence of A and B given C , there must be an additional edge connecting A and B . Without the edge $\{A, B\}$, A and B would be separated given C and hence would be independent. With the edge $\{A, B\}$, they can still be a priori independent as independence can also be expressed in terms of the potential values.

Log-linearity In practice, it is often beneficial to choose the clique potentials in an MRF such that they take the functional form of an exponentiated weighted sum of binary features, i.e.

$$\psi_c(x_{\{c\}}) = \exp\left(\sum_{i \in \mathcal{X}_{\{c\}}} w_{c,i} f_{c,i}(x_{\{c\}})\right),$$

where $\mathcal{X}_{\{c\}}$ denotes the set of possible configurations of the clique c and $f_{c,i}(x_{\{c\}})$ denotes a Boolean feature function returning 1 iff $x_{\{c\}}$ corresponds to the i -th configuration of the clique c . $w_{c,i}$ denotes a real-valued weight that is attached to the i -th configuration of clique c . If all clique potentials take this log-linear form, the MRF distribution has a couple of particularly appealing representational and computational properties. First, the product over all clique potentials in Equation (2.9) reduces to a sum over all weights attached to the respective clique configurations that hold in a specific possible world x . And second, the gradient with respect to a particular model parameter $w_{c,i}$ can be computed fairly efficiently, as I show below. A slightly more general representation of MRFs is obtained by relaxing the factorization of the joint distribution in (2.9) which is imposed by the topological constraints of the maximal cliques and allowing features to represent aspects of a possible world on a global scope. Such an MRF distribution, also called a *Boltzmann distribution* is given by

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i f_i(x)\right), \quad (2.10)$$

where Z denotes again the partition function given by

$$Z = \sum_{x' \in \mathcal{X}} \exp\left(\sum_i w_i f_i(x')\right),$$

f_i is a feature function that is 1 iff the respective feature is present in the world x and 0 otherwise, and w_i is a real-valued weight attached to the i -th feature. The representation in (2.10) is both more compact and more general than the original definition of the MRF distribution in (2.9), since it considers features more ‘global’

characteristics of a possible world rather than a ‘local’ configuration of a clique. It also makes the MRF design more convenient as a knowledge engineer does not need to be attentive to the topological structure and the maximal cliques in a network, but can concentrate on the variables and features of the probabilistic KB.

Inference Considering an MRF representing a probability distribution over a KB in propositional logic, a canonical inference problem for computing the posterior probability of an arbitrary sentence ϕ given another sentence ψ is given by

$$P(\phi | \psi) = \frac{\sum_{x' \models \phi \wedge \psi} \mu(x')}{\sum_{x \models \psi} \mu(x)}. \quad (2.11)$$

$\mu(x)$ is called the *probability mass* function and assigns a positive real-valued measure to every possible world x , which determines x ’s portion of the available total mass of probability given by $\sum_{x \in \mathcal{X}} \mu(x)$. It is evident that exact inference according to (2.11) requires enumerating all models of particular logical sentences, which is known to be #P-complete (Roth, 1996) and thus intractable for real-world applications. In practice, approximate algorithms therefore need to be applied. Among those, Markov chain Monte Carlo (MCMC) based methods such as Gibbs sampling (Russell and Norvig, 2003) are perhaps the most popular ones. Alternatively, there are subsets of general MRFs that impose constraints on the structural design of a model that is less expressive but allows for specialized more efficient algorithms. Examples of such MRFs include logistic regression models (Bishop, 2006) or conditional random fields (Lafferty et al., 2001).

Parameter Estimation The parameters w_i of features in an MRF taking the functional form of a log-linear Boltzmann distribution as in (2.10) are particularly elegant to obtain. Considering a set \mathcal{D} of independently drawn, identically distributed fully observed examples, the log-likelihood function is given by

$$\begin{aligned} \mathcal{L}(\mathbf{w}) &= \log P(\mathbf{w} | \mathcal{D}) \propto \log \prod_{d \in \mathcal{D}} P(X = d | \mathbf{w}) \\ &= \sum_{d \in \mathcal{D}} \sum_i w_i f_i(d) - \log Z. \end{aligned} \quad (2.12)$$

As maximizing Equation (2.12) does not have a closed form solution, numeric optimization methods, such as Newton or quasi-Newton methods need to be applied in order to find the parameters \mathbf{w} that maximize the log-likelihood. To this end, its partial derivative with respect to a particular component w_i of the weight vector \mathbf{w}

can be obtained by computing

$$\begin{aligned} \frac{\partial \mathcal{L}(w)}{\partial w_i} &= \frac{\partial}{\partial w_i} \sum_{d \in \mathcal{D}} \left(\sum_i w_i f_i(d) - \frac{\partial}{\partial w_i} \log \left(\sum_{x' \in \mathcal{X}} \exp \left(\sum_i w_i f_i(x') \right) \right) \right) \\ &= \sum_{d \in \mathcal{D}} \left(f_i(d) - \sum_{x' \in \mathcal{X}} f_i(x') P(X = x') \right). \end{aligned} \quad (2.13)$$

The gradient in (2.13) intuitively represents for every feature the difference between the value of the feature in the data and the expectation of the feature with respect to all possible worlds subject to the model parameters. Consequently, if model parameters have been found, such that this difference disappears (equals 0), the prediction of the model for a feature and the actual value of the respective feature in the data coincide, which in turn means that the model is perfectly fitted to the training data. The likelihood is maximal in this particular set of parameters. However, as the computation of the gradient requires inference over the model as a whole, exact learning is intractable for all but the smallest examples.

It is not always necessary to represent a full joint distribution over all variables in an MRF. Instead, in many real-world applications, a knowledge engineer can identify random variables that will never appear in any query but can in any case be observed as evidence. Examples of such models are, for instance, optical character recognition systems, where the input pixels of a document are always observed but are never subject to reasoning. Such kinds of models, which are commonly referred to as *discriminative* models, are typically computationally cheaper than their *generative* counterparts as they require enumeration over fewer possible worlds.



Generative vs.
Discriminative
Learning

2.3 Probabilistic Relational Models

Probabilistic graphical models, such as Bayesian networks and Markov random fields, provide very elegant means for compactly representing joint probability distributions over large sets of random variables. As opposed to a naive table-based representation of random events, which was hopelessly infeasible, these models carry a graph-based structure that makes independencies among variables explicit and thus allow more compact representations and more efficient computations. In addition, these graphical structures are fairly easily interpretable and thus can be straightforwardly designed by domain experts and knowledge engineers.

However, the number of random variables in PGMs is fixed and must be known beforehand in order to instantiate a respective network. In many practical applications, however, it is desirable for KBs to support varying numbers of random variables. As an example, consider the perception component of a household robot that is to categorize different items that its sensors have identified in the environment (cf. Nyga et al. (2014)). Ideally, such a classification system should support inputs of arbitrary lengths since the number of entities to classify cannot be known at compile time. Their propositional nature thus limits the expressiveness of classical PGMs.

The field of PRMs is a research direction that aims at overcoming the obvious limitations of logics and PGMs by lifting propositional probabilistic models to first-order representations and thereby provide knowledge representation formalisms that are as expressive as FOL, but still able to deal with uncertainty and inconsistency in a meaningful way. It has been a long-standing goal in the AI and machine learning community to combine probability theory and logic in a single, coherent representation and many attempts towards this direction have been published in the recent two decades. An excellent and comprehensive survey of techniques is given by Getoor (2007a). One of the methods that have garnered most attention in recent years is perhaps Markov logic networks (MLNs), a formalism combining the expressiveness of FOL with the probabilistic semantics of MRFs.

2.3.1 Markov Logic Networks

In this section, I introduce the concept of Markov logic networks, following mainly the definitions of Richardson and Domingos (2006) and Jain (2012).

A Markov logic network L consists of an indexed set of pairs $\langle F_i, w_i \rangle$, where F_i is a formula in FOL and w_i is a real-valued weight attached to the i -th formula. Intuitively, a formula's weight determines the 'hardness' or 'correctness' of the respective formula, i.e. the larger the weight of a formula is, the more important the formula is supposed to be; or the larger the penalty for violation of the formula is, respectively. In most implementations of Markov logic, predicate arguments are *typed*, i.e. all arguments of any predicate are bound to a dedicated named set of values, their *domain*. This is in contrast to original FOL, where constant predicate arguments are in one universal set of constant symbols. Given such a finite domain of discourse, a ground Markov random field (ground MRF) can be instantiated by introducing to the MRF one Boolean variable for each ground atom reflecting the truth of the respective proposition. The set of possible worlds \mathcal{X} is in turn given by the set of