IJCAT International Journal of Computing and Technology
www.IJCAT.org
ISSN : 2348 - 6090

Recognizing Character using Bayesian Network in Data Mining

[1]Anver Muhammed.K.M, [2] Radhakrishnan B, [3] Shine Raj G, [4]Jayakrishnan R

[1, 2, 3, 4] Baselios Mathews II College of Engineering, Sasthamcotta, Kollam, Kerala, India.

Abstract - Bayesian Networks are used to represent knowledge about an uncertain domain. Searching a data or word from a database is a random process. Probabilistic theories can be used to analyze the data. We use Bayes theorm for recognizing a word. The frequency in which the word occurs is the criteria for search.

***Keywords* - Bayes theorm, hypothesis, probability.**

I. Introduction

Bayesian networks (BNs), also known as *belief networks*, belong to the family of probabilistic *graphical models* (GMs). These graphical structures are used to represent knowledge about an uncertain domain. In particular, each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding random variables. These conditional dependencies in the graph are often estimated by using known statistical and computational methods. Hence, BNs combine principles from graph theory, **probability theory**, computer science, and statistics. The structure of a DAG is defined by two sets: the set of nodes (vertices) and the set of directed edges.

The nodes represent random variables and are drawn as circles labeled by the variable names. The edges represent direct dependence among the variables and are drawn by arrows between nodes. In particular, an edge from node *Xi* to node *Xj* represents a statistical dependence between the corresponding variables. Thus, the arrow indicates that a value taken by variable *Xj* depends on the value taken by variable *Xi* , or roughly speaking that variable *Xi* "influences" *Xj* . Node *Xi* is then referred to as a *parent* of *Xj* and, similarly, *Xj* is referred to as the *child* of *Xi* . An extension of these genealogical terms is often used to define the sets of "descendants" – the set of nodes that can be reached on a direct path from the node, or "ancestor" nodes – the set of nodes from which the node can be reached on a direct path. The structure of the acyclic graph guarantees that there is no node that can be its own ancestor or its own descendent. Such a condition is of vital importance to the factorization of the joint probability of a collection of nodes. Note that although the arrows represent direct causal connection between the variables, the *reasoning process* can operate on BNs by propagating information in any direction.

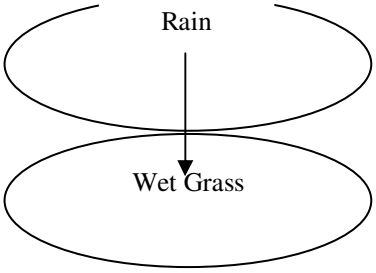

Fig. 1. Bayesian network modeling that rain is the cause of wet grass

2. Bayes Theorem

Data comes from a process that is not completely known. This lack of knowledge is indicated by modeling the process as a random process. Maybe the process is actually deterministic, but because we do not have access to complete knowledge about it, we model it as random and use probability theory to analyze it. The extra pieces of knowledge that we do not have access to are named the *unobservable variables.* In the coin tossing example, the only *observable variable* is the outcome of the toss. Denoting the unobservables by z and the observable as *x,* in reality we have $x = f(z)$ where $f(\cdot)$ is the deterministic function that defines the outcome from the unobservable pieces of knowledge. Because we cannot model the process this way, we define the outcome X as a random variable drawn from a probability distribution $P(X = x)$ that specifies the process. Tossing a coin is a random process because we cannot predict at any toss whether the outcome will be heads or tails-that is why we toss . coins, or buy lottery tickets, or get insurance. Bayes rule is used to calculate the probabilities of classes.

Bayesian learning methods are relevant in machine learning for two different reasons. First, Bayesian learning

algorithms that calculate explicit probabilities for hypotheses, such as the naive Bayes classifier, are among the most practical approaches to certain types of learning problems and advantageous than other learning algorithms, including decision tree and neural network algorithms. The naive Bayes classifier is competitive with other learning algorithms in many cases and that in some cases it outperforms those methods. It is mainly used to classify text documents such as electronic news articles.

Second reason that Bayesian methods are important in machine learning is that they provide a useful perspective for understanding many learning algorithms that do not explicitly manipulate probabilities.

Features of Bayesian learning methods include:

- Each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct. This provides a more flexible approach to learning than algorithms that completely eliminate a hypothesis if it is found to be inconsistent with any single example.
- Prior knowledge can be combined with observed data to determine the final probability of a hypothesis. In Bayesian learning, prior knowledge is provided by asserting (1) a prior probability for each candidate hypothesis, and (2) a probability distribution over observed data for each possible hypothesis.
- Bayesian methods can accommodate hypotheses that make probabilistic predictions.
- New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities.
- Even in cases where Bayesian methods prove computationally intractable, they can provide a standard of optimal decision making against which other practical methods can be measured.

One practical difficulty in applying Bayesian methods is that they require initial knowledge of many probabilities. When these probabilities are not known in advance they are often estimated based on background knowledge, previously available data, and assumptions about the form of the underlying distributions. A second practical difficulty is the significant computational cost required to determine the Bayes optimal hypothesis.

In machine learning we are often interested in determining the best hypothesis from some space H, given the observed training data D. One way to specify what we mean by the ***best*** hypothesis is to say that we demand the ***most probable*** hypothesis, given the data ***D*** plus any initial knowledge about the prior probabilities of the various hypotheses in H. Bayes theorem provides a direct method for calculating such probabilities. More precisely, Bayes theorem provides a way to calculate the probability of a hypothesis based on its prior probability, the probabilities of observing various data given the hypothesis, and the observed data itself. To define Bayes theorem precisely, let us first introduce a little notation. We shall write P(h) to denote the initial probability that hypothesis h holds, before we have observed the training data. *P(h)* is often called the prior probability of h and may reflect any background knowledge we have about the chance that h is a correct hypothesis. If we have no such prior knowledge, then we might simply assign the same prior probability to each candidate hypothesis. Similarly, we will write P(D) to denote the prior probability that training data D will be observed (i.e., the probability of D given no knowledge about which hypothesis holds). Next, we will write ***P(D/h)*** to denote the probability of observing data ***D*** given some world in which hypothesis ***h*** holds. More generally, we write ***P(x/y)*** to denote the probability of x given ***y.*** In machine learning problems we are interested in the probability ***P (h/ D)*** that ***h*** holds given the observed training data ***D. P (h/ D)*** is called the ***posterior probability*** of ***h,*** because it reflects our confidence that ***h*** holds after we have seen the training data ***D.*** Notice the posterior probability ***P(h/D)*** reflects the influence of the training data ***D,*** in contrast to the prior probability ***P(h)***, which is independent of D.

Bayes theorem is the cornerstone of Bayesian learning methods because it provides a way to calculate the posterior probability ***P(h/D),*** from the prior probability ***P(h),*** together with ***P(D)*** and ***P(D(h).***

$$P(h/d)=\frac{P(D/h)P(h)}{P(D)} \quad (1)$$

$$h_{MAP} \equiv \arg\max_{h \in H} P(h/D) \quad (2)$$

$$= \arg\max_{h \in H} \frac{P(D/h)P(h)}{P(D)}$$

$$= \arg\max_{h \in H} P(D/h)P(h)$$

3. Bayesian Recognizer

Words that appear frequently in the language are recognized more easily than words that appear less frequently. With words of equal frequency, participants should have to choose the alternative that best matches the target. In the case of low frequency words a random choice

has to be done. So in the case of low frequency words more random guesses need to be done.

In cases if input is completely unambiguous, its easy to select the entry whose representation matches the input. The actual implementation of the matching process might be very complex, but the optimal strategy is just to select the matching word from the lexicon. An optimally designed system would be able to match the input against all words in the lexicon in parallel, and there would be no effect of word-frequency. However, if the input is ambiguous, the requirements are different. With ambiguous input it is no longer sufficient simply to select the best matching lexical entry. Under these circumstances we must also take the prior probabilities of the words into account.

Starting from knowledge of the prior probabilities with which events or hypotheses (H) occur, Bayes theorem indicates how those probabilities should be revised in the light of new evidence (E). Given the prior probabilities of the possible hypotheses P(Hi), and the likelihood that the evidence is consistent with each of those hypotheses P(E|Hi), Bayes theorem can be used to calculate P(Hi|E), the revised, or posterior probability of each hypothesis, given the evidence.

$$P(W/I) = \frac{P(W) \times P(I/W)}{\sum_{i=0}^{n} P(w_i) \times P(I/W_i)} \qquad (3)$$

In the case of word recognition, the hypotheses will correspond to words, and P(H) is given by the frequency of the word.The probability that the input corresponds to a particular word is then given by the probability that the input was generated by that word, divided by the probability of observing that particular input. Any particular input could potentially be produced by many different words. The probability of generating a particular input is therefore obtained by summing the probabilities that each word might have generated the input.

In the absence of any knowledge of the prior probabilities of the words, the best strategy will always simply be to choose the word that best matches the input. However, if prior probabilities are available, these should be taken into account by the application of Bayes theorem.

Words are generally presented clearly under circumstances where the participant will have no difficulty in identifying the stimulus accurately. However, the critical question is not whether the stimulus itself could potentially be identified unambiguously, but whether there is ambiguity at the point where a decision is made. Participants are expected to respond as quickly as possible. Indeed, participants generally respond so quickly that they make errors. If they respond before they have reached a completely definitive interpretation of the input, there will necessarily be some residual ambiguity at the point where the decision is made. Under these circumstances, frequency should still influence responses, even though the stimulus itself is presented clearly.

If the recognizer is presented with a noisy representation of a word there will be some probability that the word that most closely matches the input will not be the correct word. If the input most closely matches a low frequency word, there is some probability that the input was actually produced by another word. If the other word is much more frequent than the low frequency word, it may be more likely that the input was produced by the less well matching high frequency word than the more closely matching low frequency word. That is, information about word frequency effectively alters the weighting of perceptual evidence.

In Bayes theorem the measure of the evidence for a word contains the term P(I|W), that is, the probability of observing the perceptual input I, given that the word W has been presented. P(I|W) could be determined by experience. For example, each time a word is encountered it will produce some representation Ix at the input of the recognizer. The recognizer could learn the probability density function (pdf) representing the probability of receiving a particular input, given that a particular word was presented. For any new input, the system would then be able to look up the probability that that input came from presentation of a particular word (P(I|W). This is effectively how some automatic speech recognition systems are trained to recognize words using hidden Markov models.

In fact, P(I|W) could be computed from the products of P(I|Letter) over letter positions, so there is no need for the system to have extensive experience of every word in the lexicon.

In effect, frequency in the Bayesian approach acts as a bias. For example, in a perceptual identification task, a Bayesian recognizer should respond with the word with the largest posterior probability. Other things being equal, high frequency words would therefore tend to be generated as responses more often than low frequency words. However, frequency would not improve the perceptual sensitivity of the system in terms of its ability to discriminate between a pair of words in a forced-choice task.

High frequency words just require fewer features to exceed threshold than do low frequency words If P(I|correct word) is 1.0, and all other P(I|W) are 0.0, then frequency P(W) cancels out. The better the perceptual evidence, the smaller will be the influence of frequency. Frequency can never override reliable perceptual evidence. This is clearly a desirable property. No matter how large the frequency difference between two similar words, a Bayesian decision process will always select the correct word when the input is unambiguous.

4. Bayesian Approach in Visual Recognition

The goal is to establish whether some of the most important features of visual word recognition can be explained simply by assuming that readers approximate ideal observers operating on a noisy perceptual input. All of the important features of the model follow from this simple assumption The Bayesian Reader is the first step in this process.

As a first step in assessing the behavior of an optimal word recognizer, it is necessary to have an estimate of P(I|W). Although P(I|W) could be learned, for the purposes of the simulations presented here, we will take a rather different approach and assume that P(I|W) can be estimated from the current input. This depends on three assumptions:

1. All words can be represented as points in a multidimensional perceptual space.
2. Perceptual evidence is accumulated over time by successively sampling from a distribution centered on the true perceptual co-ordinates of the input with samples being accumulated at a constant rate.
3. P(I|W) for all words can be computed from an estimate of variance of the input distribution

The first assumption follows simply from the fact that some words are more easily confused than others. The second assumption is that perceptual information were completely unambiguous, word frequency should have no influence on recognition. The third assumption avoids the need to learn the probability density function of P(I|W) for individual words. It also helps to keep the model simple and general, and avoids making any arbitrary assumptions about learning. As successive samples arrive, it is possible to compute the mean location and the standard error of the mean (SEM) of all samples received so far.

$$\sigma_M = \sigma / \sqrt{N} \qquad (4)$$

The mean represents a point in perceptual and the SEM is computed from the distances between each sample and the sample mean. The SEM is measured in units corresponding to Euclidean distance in perceptual space. Given the form of input being assumed here, I is a continuous valued variable whose probability distribution is then correctly represented as a density function f(I|W). Under these assumptions the equivalent Bayes equation is:

$$P(W/I) = \frac{P(W) x f(I/W)}{\sum_{i=0}^{n} P(w_i) x f(I/Wi)} \qquad (5)$$

Where f(I|W) corresponds to the height of the pdf at I. For a given I, f(I|W) is called the likelihood function of W. When comparing different candidate W's on the basis of input I, it is the ratio of the likelihoods that influence the revision of the prior probabilities Words that are far away from the mean of the input distribution will tend to become less and less likely as more samples are accumulated. Consequently, P(W|I) of the word actually presented will tend to increase, while the P(W|I) of all other words will decrease. One noteworthy feature of this kind of sampling model is that, given enough samples, there is no limit as to how small the SEM can become. In the absence of any restrictions on the amount of data available (i.e. number of samples), the P(W|I) of a clearly presented word will always approach 1.0 in the limit.

For some purposes it may be possible to derive P(I|W), or P(Input | Letter) from perceptual confusion matrices, rather than estimating them from the input. Note that as more perceptual information arrives, P(W) will have less and less influence on P(W|I). In the limit, P(I|W) for all but the word actually presented will approach 0, and P(W) will have no effect whatsoever. However, in general, as P(W) gets lower, the number of samples required to reach a given P(W|I) will increase. That is, high frequency words will be identified more quickly than low frequency words.

It is important to bear in mind that the posterior probabilities being calculated here are the probabilities that the input is a particular word, given that the input really is a word. Because of the properties of the normal distribution, the closest word to the input mean will always have a probability approaching 1.0 in the limit, even if the input does not correspond exactly to any particular word. The decision being made is: given that the input is a word, which word is it? Even an unknown word will produce a high P(W|I) for one existing word in the lexicon. When simulating identification of known words, this limitation is not a problem. However, consideration of how to handle unknown words will become important later when modeling lexical decision.

5. Conclusion

Bayesian Networks became popular models in the last decade. They have been used for applications in various areas, such as machine learning, text mining, natural language processing, speech recognition, signal processing, bioinformatics, error-control codes, medical diagnosis, weather forecasting, and cellular networks. In a general form of the graph, the nodes can represent not only random variables but also hypotheses, beliefs, and latent variables. Such a structure is intuitively appealing and convenient for the representation of both causal and probabilistic semantics. This structure is ideal for combining prior knowledge, which often comes in causal form, and observed data. Bayesian Network can be used, even in the case of missing data, to learn the causal relationships and gain an understanding of the various problem domains and to predict future events.

References

[1] Ehsan Hajizadeh*, Hamed Davari Ardakani and Jamal Shahrabi Application of data mining techniques in stock markets:A survey.

[2] Yue, X., Wu, Y., Wang, Y. L., & Chu, C. (2007). A review of data mining-based financial fraud detection research, international conference on wireless communications Sep, Networking and Mobile Computing (2007) 5519–5522.

[3] Oxford Concise English Dictionary, 11th Edition, Oxford University Press, 2009.

[4] Phua, C., Lee, V., Smith, K. & Gayler, R. (2005). A comprehensive survey of data mining-based fraud detection research, Artificial Intelligence Review (2005) 1–14.

[5] Wang, J., Liao, Y., Tsai, T. & Hung, G. (2006). Technology-based financial frauds in Taiwan: issue and approaches, IEEE Conference on: Systems, Man and Cyberspace Oct (2006) 1120–1124.

[6] Wang, S. (2010). A Comprehensive Survey of Data Mining-Based Accounting-Fraud Detection Research. International Conference on Intelligent Computation Technology and Automation, vol. 1, pp.50-53, 2010.

[7] Accounting Fraud Definition and Examples retrieved from http://www.accountingelite.com/accountingtips/accounting-fraud-definition-and-examples-freeaccounting-fraud-article/.

[8] Ngai, E.W.T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2010). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature, Decision Support System(2010), doi:10.1016/j.dss.2010.08.006.

[9] Kirkos, E., Spathis, C., & Manolopoulos, Y. (2007). Data mining techniques for the detection of fraudulent financial statements, Expert Systems with Applications 32 (4) (2007) 995–1003.

[10] Fanning, K., Cogger, K., & Srivastava, R. (1995). Detection of management fraud: a neural network approach. International Journal of Intelligent Systems in Accounting, Finance & Management, vol. 4, no. 2, pp. 113– 26, June 1995.

[11] Fanning, K., & Cogger, K. (1998). Neural network detection of management fraud using published financial data. International Journal of Intelligent Systems in Accounting, Finance & Management, vol. 7, no. 1, pp. 21-24, 1998.