BAYESIAN BELIEF NETWORKS

INTRODUCTION

- Bayesian learning methods are relevant to 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.
- The second reason that Bayesian methods are important to 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

- 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.

Features of Bayesian learning methods

- Bayesian methods can accommodate hypotheses that make probabilistic predictions (e.g., hypotheses such as "this pneumonia patient has a 93% chance of complete recovery").
- 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.

Difficulties of Bayesian learning methods

- One practical difficulty in applying Bayesian methods is that they typically 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.
- A second practical difficulty is the significant computational cost required to determine the Bayes optimal hypothesis in the general case.

BAYESIAN BELIEF NETWORKS

- A Bayesian belief network describes the probability distribution governing a set of variables by specifying a set of conditional independence assumptions along with a set of conditional probabilities.

- In contrast to the naive Bayes classifier, which assumes that ***all*** the variables are conditionally independent given the value of the target variable.

BAYESIAN BELIEF NETWORKS

- Bayesian belief networks allow stating conditional independence assumptions that apply to ***subsets*** of the variables.
- Bayesian belief networks provide an intermediate approach that is less constraining than the global assumption of conditional independence made by the naive Bayes classifier.
- Bayesian belief networks are an active focus of current research, and a variety of algorithms have been proposed for learning them and for using them for inference.

BAYESIAN BELIEF NETWORKS

- In general, a Bayesian belief network describes the probability distribution over a set of variables. Consider an arbitrary set of random variables Yl . . . Y,, where each variable Yi can take on the set of possible values V(Yi).
- the ***joint space*** of the set of variables Y to be the cross product V(Yl) x V(Y2) x . . . V(Y,).

- The probability distribution over this joint' space is called the ***joint probability distribution.***

BAYESIAN BELIEF NETWORKS

- The joint probability distribution specifies the probability for each of the possible variable bindings for the tuple (Yl . . . Y,).
- A Bayesian belief network describes the joint probability distribution for a set of variables.

Conditional Independence

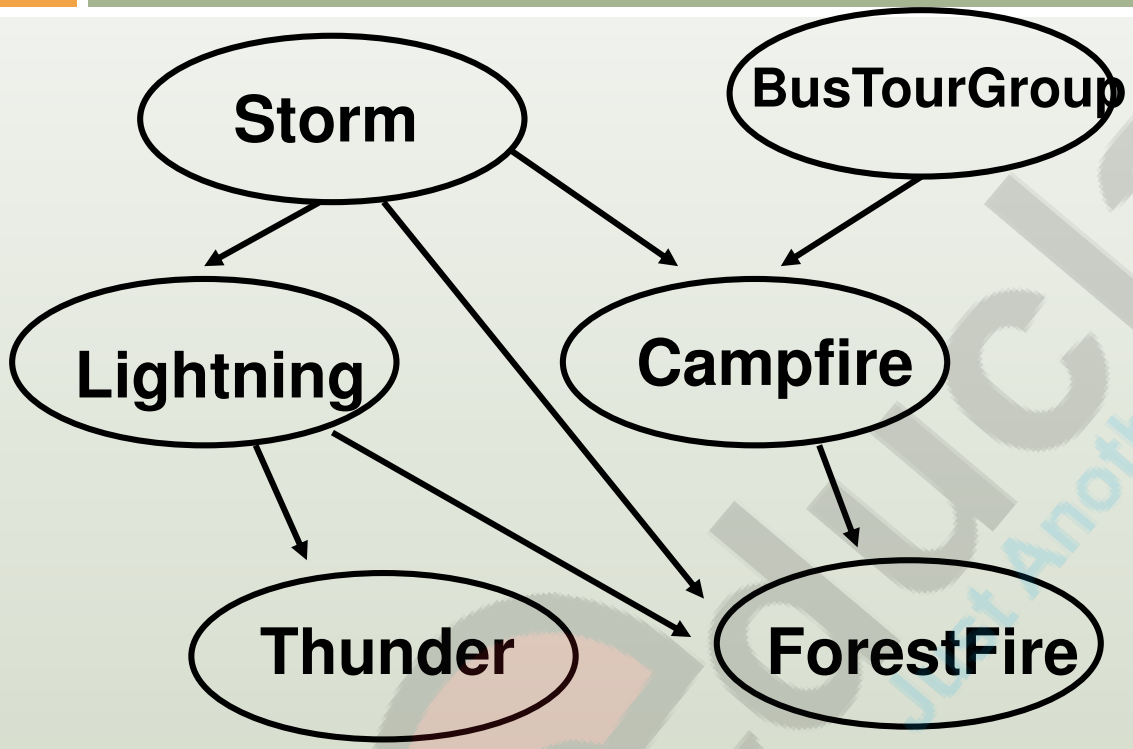
- Bayesian belief networks by defining precisely the notion of conditional independence.
- Let X, Y, and Z be three discrete-valued random variables.
- It says that X is ***conditionally independent*** of Y given Z if the probability distribution governing X is independent of the value of Y given a value for **Z.**

Conditional Independence

- We say that X is ***conditionally independent*** of Y given Z if the probability distribution governing X is independent of the value of Y given a value for Z.
- i.e., $(\forall x_i, y_j, z_k)\ P(X=x_i|Y=y_j,Z=z_k)=P(X=x_i|Z=z_k)$
- or, $P(X|Y,Z)=P(X|Z)$
- This definition can be extended to sets of variables as well: we say that the set of variables $X_1 \ldots X_l$ is conditionally independent of the set of variables $Y_1 \ldots Y_m$ given the set of variables $Z_1 \ldots Z_n$, if

$$P(X_1 \ldots X_l|Y_1 \ldots Y_m,Z_1 \ldots Z_n(=P(X_1 \ldots X_l|Z_1 \ldots Z_n)$$

Representation in Bayesian Belief Networks



Associated with each node is a conditional probability table, which specifies the conditiona distribution for the variable given its immediate parents in the graph

Each node is asserted to be conditionally independent of its non-descendants, given its immediate parents

Inference in Bayesian Belief Networks

- A Bayesian Network can be used to compute the probability distribution for any subset of network variables given the values or distributions for any subset of the remaining variables.
- Unfortunately, exact inference of probabilities in general for an arbitrary Bayesian Network is known to be NP-hard.
- In theory, approximate techniques (such as Monte Carlo Methods) can also be NP-hard, though in practice, many such methods were shown to be useful.

Learning Bayesian Belief Networks

3 Cases:

1. The network structure is given in advance and all the variables are fully observable in the training examples. ==> Trivial Case: just estimate the conditional probabilities.
2. The network structure is given in advance but only some of the variables are observable in the training data. ==> Similar to learning the weights for the hidden units of a Neural Net: Gradient Ascent Procedure
3. The network structure is not known in advance. ==> Use a heuristic search or constraint-based technique to search through potential structures.

The EM Algorithm: Learning with unobservable relevant variables.

- **Example:** Assume that data points have been uniformly generated from k distinct Gaussian with the same known variance. The problem is to output a hypothesis $h=<\mu_1, \mu_2 ,.., \mu_k>$ that describes the means of each of the k distributions. In particular, we are looking for a maximum likelihood hypothesis for these means.
- We extend the problem description as follows: for each point x_i, there are k hidden variables $z_{i1},..,z_{ik}$ such that $z_{il}=1$ if x_i was generated by normal distribution l and $z_{iq}=0$ for all $q \neq l$.

The EM Algorithm (Cont'd)

- An arbitrary initial hypothesis $h=<\mu_1, \mu_2, .., \mu_k>$ is chosen.
- The EM Algorithm iterates over two steps:
  - ***Step 1 (Estimation, E):*** Calculate the expected value E[zij] of each hidden variable zij, assuming that the current hypothesis $h=<\mu_1, \mu_2, .., \mu_k>$ holds.
  - ***Step 2 (Maximization, M):*** Calculate a new maximum likelihood hypothesis $h'=<\mu_1', \mu_2', .., \mu_k'>$, assuming the value taken on by each hidden variable z_{ij} is its expected value $E[z_{ij}]$ calculated in step 1. Then replace the hypothesis $h=<\mu_1, \mu_2, .., \mu_k>$ by the new hypothesis $h'=<\mu_1', \mu_2', .., \mu_k'>$ and iterate.

The EM Algorithm can be applied to more general problems