Weekly report of lessons

Name: Mayank Kumar
Roll No: 19CS30029
The week: 13-09-2021 to 19-09-2021

The topics covered:

Minimum Description Length Principle in Bayesian Learning, Optimal Classifier, Gibbs Algorithm, Discriminant Functions, Challenges in computing; Naive Bayes Classifier: Likelihood Estimation, Pros and Cons; Bayesian Network: Formation of a graphical model, Conditional Independence, Larger graphs from simpler graphs, Computation, Inference; Bayesian Decision Making - Losses and Risks, Generalization to utility theory, Mining association rule, Measures of association rule, Apriori Algorithm, Association and causality

Summary topic wise:

Minimum Description Length Principle in Bayesian Learning: The principle states to choose that hypothesis H which minimizes L(H) along with LH(D), where L(H) is length of hypothesis, and LH(D) is length of description of data D using hypothesis H.

$$h_{MAP} = argmax_{h \in H} P(D|h)\ P(h)$$
$$= argmax_{h \in H} (log_2 P(D|h) + log_2 P(h))$$
$$= argmin_{h \in H} (- log_2 P(D|h) - log_2 P(h))$$

Here, $- \log_2 P(D|h)$ is length of information data and $-\log_2(P(h))$ is length of optimal encoding for H.

Optimal Classifier: For all h in hypothesis space for decision making weighted by their posterior, target function for x: $c(x) = argmax_{v \in V} \{\sum_{h \in H} (P(v \mid h)\ P(h \mid D))\}$, where V is the set of class labels

Gibbs Algorithm: Applies on the instance x a hypothesis h, randomly chosen according to P(h | D).

$$E[error_{Gibbs}] \leq 2\ E[error_{BayesOptimal}]$$

Discriminant Functions: We can express Bayesian Classifier based on a set of discriminant functions. The target class C_i is assigned if $g_i(x) > g_k(x)\ \forall$ k except i, where $g_i(x)$ and $g_k(x)$ are discriminant functions. For two classes, $g(x) = g_1(x) - g_2(x)$

Challenges in computing: Getting prior knowledge of probabilities of classes and probability distribution in multidimensional feature spaces.

Naive Bayes Classifier

It is based on a property that every pair of features being classified is independent of each other. It is best suited for a moderate or large training set.

Likelihood Estimation: If the attributes that describe instances are conditionally independent, the likelihood is estimated as $P(X \mid C_i) = \Pi_{k=1 \text{ to } n} P(x_k \mid C_i)$

$P(x_i \mid C_k)$ = fraction of occurrence in a class (for categorical variable)
= use parametric modeling of Gaussian Distribution (for continuous variable)

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$P(X \mid C_i) = g(x_k, \mu_i, \sigma_i)$

Avoiding Zero-Probability: By using Laplacian correction, i.e., add 1 to each case

Pros and Cons: Easy to implement and yield good results, but accuracy is lost due to attributes being conditionally independent.

Bayesian Network

It represents the set of conditional independence assertions in directed acyclic graph where each node is conditionally independent of its nondescendants and represents the probability of random variable X.

Formation of a graphical model: Add nodes, and arcs between any two nodes if they are not independent.

Conditional Independence: For X and Y to be independent, P(Y|X) = P(Y), P(X|Y) = P(X), P(X,Y) = P(X)P(Y). Conditional Independence between X and Y given occurrence of third event Z:

- $P(X,Y \mid Z) = P(X \mid Z)\ P(Y \mid Z)$
- $P(X, Y, Z) = P(Y)\ P(Z \mid Y)\ P(X \mid Z)$ (Head to Tail connection)
- $P(X, Y, Z) = P(Z)\ P(X \mid Z)\ P(Y \mid Z)$ (Tail to Tail connection)

In Head-to-Head connection, X and Y are initially independent but become dependent when Z is known.

Larger graphs from simpler graphs: Independencies are explicitly encoded and inference is broken into calculations over small groups. Then, we propagate from evidence nodes to query nodes.

Computation on Bayesian Network: The probabilities of a set of variables is inferred when another set of variables is given as evidence.
Inference through Bayesian Networks: For any subset of X_i, $P(X_1, X_2, ..., X_d) = \Pi_{i=1 \text{ to } d} P(X_i \mid \text{parent of } X_i)$

Bayesian Decision Making - Losses and Risks: If action a_i assigns x to class C_i and l_{ik} is the loss if x belongs to C_k, then Expected risk for taking a_i, $R(a_i \mid x) = \Sigma_k (l_{ik} P(C_k \mid x))$, and $P(C_i \mid x) = 1 - R(a_i \mid x)$
The chosen a_i should minimize R(.) or maximize $P(C_i \mid x)$ the most.

Generalization to utility theory: Gain U_{ik} is considered instead of the loss l_{ik}. The chosen a_i should have maximum $EU(a_i \mid x)$, where $EU(a_i \mid x)$ or Expected Utility = $\Sigma_k (U_{ik} P(C_k \mid x))$

Mining Association Rules: For implication $X \rightarrow Y$, X is the antecedent and Y is the consequent.
Measures: Support(X,Y): $P(X,Y)$, shows statistical significance
Confidence (X→Y): $P(Y|X) = P(X,Y) / P(X)$, indicates strength of rule
Lift(X,Y): $P(Y|X) / P(Y)$, close to 1 if X and Y are independent

Apriori Algorithm: Gets association rules using high support and confidence. It finds frequent item sets with enough support and converts them to rules with high confidence by splitting them into antecedent and consequent item sets.

Association and casuality: X→Y indicates an association. The variables acting in the unidentified process might be hidden.

Concepts challenging to comprehend:
None

Interesting and exciting concepts:
The graphical model formed through Bayesian Network and its significance was quite exciting. Another interesting point to infer is that being a probabilistic classifier, the Naive Bayes Classifier will work really well even for less training dataset, if we assume that the attributes are independent.

Concepts not understood:
None.

Any novel idea of yours out of the lessons:
As mentioned in Association and casuality, and in real-world learning tasks as well, there may be some variables or instances which remain hidden. So, there is a need for such an algorithm that has an approach robust enough to learn in presence of hidden variables. To address this issue, we can roughly think of some procedure that will start with any arbitrary hypothesis and repeatedly calculate the expected values for the hidden variables. If we assume that the calculated expected values for hidden variables were correct, we can now calculate the maximum likelihood hypothesis.
