**9.** Unlike naïve Bayesian classification (which assumes class conditional independence),

**Bayesian belief networks** allow class conditional independencies to be defined between subsets of variables. They provide a graphical model of causal relationships, on which learning can be performed. Trained Bayesian belief networks can be used for classification.

**Backpropagation** is a neural network algorithm for classification that employs a method of gradient descent. It searches for a set of weights that can model the data so as to minimize the mean-squared distance between the network’s class prediction and the actual class label of data tuples. Rules may be extracted from trained neural networks to help improve the interpretability of the learned network.

A **support vector machine** is an algorithm for the classification of both linear and nonlinear data. It transforms the original data into a higher dimension, from where it can find a hyperplane for data separation using essential training tuples called **support vectors**.

*Frequent patterns* reflect strong associations between attribute–value pairs (or items) in data and are used in **classification based on frequent patterns**. Approaches to this methodology include associative classification and discriminant frequent pattern–based classification. In **associative classification**, a classifier is built from association rules generated from frequent patterns.

In **discriminative frequent pattern–based** **classification**, frequent patterns serve as combined features, which are considered in addition to single features when building a classification model.

Decision tree classifiers, Bayesian classifiers, classification by backpropagation, support vector machines, and classification based on frequent patterns are all example of **eager learners** in that they use training tuples to construct a generalization model and in this way are ready for classifying new tuples. This contrasts with **lazy learners** or **instance-based** methods of classification, such as nearest-neighbor classifiers and case-based reasoning classifiers, which store all of the training tuples in pattern space and wait until presented with a test tuple before performing generalization. Hence, lazy learners require efficient indexing techniques.

In **genetic algorithms**, populations of rules “evolve” via operations of crossover and mutation until all rules within a population satisfy a specified threshold.

**Rough set** **theory** can be used to approximately define classes that are not distinguishable based on the available attributes.

**Fuzzy set** approaches replace “brittle” threshold cutoffs for continuous-valued attributes with membership degree functions.

Binary classification schemes, such as support vector machines, can be adapted to handle **multiclass classification**. This involves constructing an ensemble of binary classifiers. Error-correcting codes can be used to increase the accuracy of the ensemble.

**Semi-supervised classification** is useful when large amounts of unlabeled data exist. It builds a classifier using both labeled and unlabeled data.

Examples of semi-supervised classification include *self-training* and *cotraining*.

**Active learning** is a form of supervised learning that is also suitable for situations where data are abundant, yet the class labels are scarce or expensive to obtain. The learning algorithm can actively query a user (e.g., a human oracle) for labels. To keep costs down, the active learner aims to achieve high accuracy using as few labeled instances as possible.

**Transfer learning** aims to extract the knowledge from one or more *source tasks* and apply the knowledge to a *target task*. **TrAdaBoost** is an example of the *instance-based approach* to transfer learning, which reweights some of the data from the source task and uses it to learn the target task, thereby requiring fewer labeled target-task tuples.