* Q9.3 – Compare and contrast *associative classification* and *discriminative frequent pattern-based classification*. Why is classification based on frequent patterns able to achieve higher classification accuracy in many cases than a classic decision tree method?

Associative classification will mine data for frequent itemsets, which result from looking for reoccurring attribute-value pairs that occur throughout the data set. Then association rules are generated from these itemsets and reviewed to meet user-defined support and confidence levels, and then use these rules to create a classifier.

Discriminative frequent pattern-based classification performs a similar process but considers frequent patterns as single features and combined features. The data set is partitioned based on class label, and then then frequent patterns are found for each partition (based on user-defined support levels). Next, feature selection is utilized to determine the more discriminating patterns via information gain, Fisher score, or numerous other methods. Lastly, a classifier is constructed based on the new data set of discriminative patterns. Classification based on frequent patterns is able to achieve higher classification accuracy than a classic decision tree, because they consider association rules with high confidence across multiple attributes. Whereas, decision trees can only consider one attribute at a time throughout its induction method.

* Q9.4 – Compare the advantages and disadvantages of *eager* classification (e.g. decision tree, Bayesian, neural network) versus *lazy* classification (e.g. k-nearest neighbor, case-based reasoning).

Eager classification (such as decision tree, Bayesian, and neural network) has the advantage of being faster than lazy classification due to the method it utilizes to add/classify new data. In addition to being faster, it also has the advantage of adding weight/importance to specific attributes based on domain expert knowledge or other means. Disadvantages of eager classification include increased training time and lack of flexibility because the model can only look at one hypothesis at a time.

Lazy classification (such as k-nearest neighbor and case-based reasoning) has the advantages of increased classification accuracy based on the hypotheses, and as previously stated, less time is required for training these models. However, the disadvantages include storage costs of data and slower classification times based on the method of constructing new classifiers every time new data is added. Additionally, different weights/importance cannot be assigned to individual attributes, resulting in a less flexible/accurate model for certain applications.

* Q9.5 – Write an algorithm for *k-nearest-neighbor classification* given *k*, the nearest number of neighbors, and *n*, the number of attributes describing each tuple.

Optional for Non-CS Students (See Q9.8 as replacement question)

* Q9.6 – Briefly describe the classification process using (a) *genetic algorithms*, (b) *rough sets*, and (c) *fuzzy sets.*

1. Genetic algorithm utilize a sort of “survival of the fittest” method to creating classification rules. A random set of rules is created based on the number of attributes and class in a data set. Then the rules are reviewed against a set of data, and the rules with the strongest accuracy, or fitness, levels are kept. This is repeated until the entire population of rules meet a desired threshold of accuracy of fitness. Mutation of rules can also be created from crossover between multiple rules and tested in a similar fashion as previously described.
2. Rough sets can only be used on discrete-valued attributes and is useful with noisy or imprecise data. Briefly described, this method equivalence classes are created based on the available attributes. Within these equivalence classes, there is no difference between the samples in regard to the attributes describing them, resulting in “rough sets” among the data. Then an upper and lower approximation is taken for each class to describe what is certain to be included/excluded from the class. From here, a decision table is created to show all the decision rules based on the rough sets.
3. Fuzzy sets take a slightly different approach than the other methods previously described. Instead of a sharp boundary line between categories, “fuzzy” boundaries are calculated. Attributes can be discretized into multiple levels (such as small, medium, large) and then the fuzzy logic will output a truth value between 0.0 and 1.0 to describe the strength of membership to each discretized level. Typically, this method is accompanied by a visual representation for a user to review. This allows a user to work with vague data resulting from imprecise data collection.

* Q9.7 – Example 9.3 showed a use of error-correcting codes for a *multiclass classification* problem having four classes.

Optional for Non-CS Students (See Q9.8 as replacement question)

* Q9.8 – *Semi-supervised classification, active learning,* and *transfer learning* are useful for situations in which unlabeled data are abundant.
  + (a) Describe *semi-supervised, active learning,* and *transfer learning*. Elaborate on applications for which they are useful, as well as the challenges of these approaches to classification.

Semi-supervised classification can be split into two forms – self-training and co-training. This method, as well as the others that will be described are useful when there is a large amount of un-labeled data being collected. In self-training, a classifier is created based on a labeled set of data, and then this is used to create class labels for an un-labeled set of data. The tuple with the highest confidence is added to the labeled data set and the process repeats until all are classified. The disadvantage of this method is that it is susceptible to compounding/increasing errors within the data set. Co-training will utilize two separate feature sets from the label data set to teach each other. It will create a classifier for both of these sets and utilize each classifier on unlabeled data tuples. The tuple with the highest confidence for each classifier will be added to the opposite classifier’s data set. This is repeated until all have been labeled. This disadvantage with this method is that it may not be possible to create two mutually exclusive sets from the given data.

Active learning is useful in situations where there is large amounts of data, but class labels are not available for the majority of them or they are difficult to obtain. This method places class labels on unlabeled data with the assistance of a human oracle. The algorithm with select tuples from the unlabeled set of data and ask a human oracle to provide a class label for them. This will then be added to the labeled data set, and the algorithm will use this updated information to classify other un-labeled tuples. This process will continue until all tuples are labeled. Various methods can be used to determine which tuples to ask the user to provide a class label for. Some will focus on uncertainty, while others will focus on reducing the total number of incorrect predictions. Depending on the selected method, the cost and computational expense can be significantly higher. However, this method typically requires a significantly smaller number of tuples to learn a concept than supervised learning.

Transfer learning aims to take a classifier used in one application, known as the source task, and apply it to a new, similar application, known as a target task. This is useful in applications where data become outdated or the distribution significantly changes. The main advantage in this method of learning is the amount of time and money saved by transferring knowledge from one algorithm to another. This means that less training data is required to teach the algorithm. One method does this by applying weights to the “old” data to match that of the “new” data in the target application by reviewing all the tuples that are deemed to have been misclassified and adjusting the weights. The goal is to filter out the influence of the old data. The main challenge with this type of learning is the possibility of negative learning, which is when the new classifier is worse than the original. This is still a topic of discussion and new methods are being reviewed to eliminate this.

* + (b) Research and describe an approach to semi-supervised classification other than self-training and co-training.

Another approach to semi-supervised classification (other than self-training and co-training) utilizes an algorithm known as transductive support machine vector (TSVM). The goal of this method is to set a boundary for the classifier at regions where there are minimal data points, whether they be labeled or unlabeled. Once these boundaries are defined, then unlabeled tuples can be labeled according to the classifier that maximizes the boundary margin.

* + (c) Research and describe an approach to active learning other than pool-based learning.

Another method for active learning is known as stream-based selective sampling. In this method, the algorithm will analyze every all tuples in the data set and determine the information gain for each one. It will decide to add a class label to the tuple or ask the human oracle for a class label. This will be much more computationally expensive, as well as increased time for classification, since the system has to review every single point.

* + (d) Research and describe an alternative approach to instance-based transfer learning.

An alternative approach to instance-base transfer learning is the pre-trained model approach for transfer learning. This method will choose an already trained classifier from a source task, typically one that was created for a set of complex, challenging data. Next, specific portions of the model are removed or altered for the new, target task. Lastly, the classifier will be tested for accuracy and tweaked as needed to meet the requirements of the new task.

