Data Mining

CS475

Spring 2019

Project

Name: Youssef Ali Fayed

ID: 162085

- **Approiri:**
  - **Fundamentals:**

    The algorithm can be divided into two main steps:

    Step 1: Apply minimum support to find all the frequent sets with k items in a database.

    Step 2: Use the self-join rule to find the frequent sets with k+1 items with the help of frequent k-itemsets. Repeat this process from k=1 to the point when we are unable to apply the self-join rule.
  - **Pros:**
    - It is an easy-to-implement and easy-to-understand algorithm.
    - It can be used on large item sets.
  - **Cons:**
    - Sometimes, it may need to find a large number of candidate rules which can be computationally expensive.
    - Calculating support is also expensive because it has to go through the entire database.
  - **Parameters:**
    - Support
    - Confidence

- **Filtered Association:**
  - **Fundamentals:**

    It works similar to the approiri algorithm.
  - **Pros:**
    - It is an easy-to-implement and easy-to-understand algorithm.
  - **Cons:**
    - It works by using another algorithm which is not really innovative.
  - **Parameters:**
    - Class Index
    - Assosiator

- **K-means:**
  - **Fundamentals:**

    used to simplify large datasets into smaller and simple datasets. Distinct patterns are evaluated and similar data sets are grouped together. The variable K represents the number of groups in the data.
    Arbitrarily choose k objects from D as the initial cluster centers.
    Repeat.
    (Re) assign each object to the cluster to which the object is most similar based on the mean value of the objects in the cluster.
    Update the cluster means.
    Until no change.
  - **Pros:**
    - Simple: It is easy to implement k-means and identify unknown groups of data from complex data sets. The results are presented in an easy and simple manner.
    - Flexible: K-means algorithm can easily adjust to the changes. If there are any problems, adjusting the cluster segment will allow changes to easily occur on the algorithm.
    - Suitable in a large dataset: K-means is suitable for a large number of datasets and it's computed much faster than the smaller dataset. It can also produce higher clusters.
    - Efficient: The algorithm used is good at segmenting the large data set. Its efficiency depends on the shape of the clusters. K-means work well in hyper-spherical clusters.
  - **Cons:**
    - No-optimal set of clusters: K-means doesn't allow development of an optimal set of clusters and for effective results, you should decide on the clusters before.
    - Lacks consistency: K-means clustering gives varying results on different runs of an algorithm. A random choice of cluster patterns yields different clustering results resulting in inconsistency.
    - Uniform effect: It produces cluster with uniform size even when the input data has different sizes.
    - Order of values: The way in which data is ordered in building the algorithm affects the final results of the data set.
  - **Parameters:**
    - Number of clusters
    - Distance function

- **EM:**
  - **Fundamentals:**
    Start with 1 point (singleton).
    2) Recursively adds two or more appropriate clusters.
    3) Stop when k number of clusters is achieved.
    Divisive (top down)
    1) Start with a big cluster.
    2) Recursively divides into smaller clusters.
    3) Stop when k number of clusters is achieved.
  - **Pros:**
    - No apriority information about the number of clusters required.
    - Easy to implement and gives best result in some cases.
  - **Cons:**
    - It may not scale well
    - No automatic discovering of optimum clusters.
  - **Parameters:**
    - Number of clusters (optional)

- **Cobweb:**
  - **Fundamentals:**
    Core points: These are points that are at the interior of a cluster.
    2) Border points: These points' falls within the neighborhood of a core point.
    3) Noise points: A noise point is any point that is not a core point or a border point.
  - **Pros:**
    - does not require the knowledge of the number of clusters in the data a priori, as opposed to k-means.
    - can find arbitrarily shaped clusters.
    - has a notion of noise.
  - **Cons:**
    - can only provide good result in the case of well-formed clusters.
    - cannot cluster data sets well with large differences in densities.
  - **Parameters:**
    - Number of clusters (optional)

- **One-R:**
  - **Fundamentals:**
    OneR, short for "One Rule", is a simple, yet accurate, classification algorithm that generates one rule for each predictor in the data, then selects the rule with the smallest total error as its "one rule". To create a rule for a predictor, we construct a frequency table for each predictor against the target. It has been shown that OneR produces rules only slightly less accurate than state-of-the-art classification algorithms while producing rules that are simple for humans to interpret.
  - **Pros:**
    - Computationally fast
  - **Cons:**
    - The main disadvantage is that rule-based methods are usually not the best performers in terms of prediction quality. Other methods (forests, SVM, deep nets) tends to be better.
  - **Parameters:**
    - -

- **J48:**
  - **Fundamentals:**
    J48 is an extension of ID3. The additional features of J48 are accounting for missing values, decision trees pruning, continuous attribute value ranges, derivation of rules, etc. In the WEKA data mining tool.
    In case the instances belong to the same class the tree represents a leaf so the leaf is returned by labeling with the same class.
    The potential information is calculated for every attribute, given by a test on the attribute. Then the gain in information is calculated that would result from a test on the attribute.
  - **Pros:**
    - Easier to interpret results
    - Helps to visualise through a decision tree
  - **Cons:**
    - Run complexity of algorithm depends on the depth of the tree(i.e the no of
    - attributes in the data set)
    - Space complexity is large as values need to be stored in arrays repeatedly
  - **Parameters:**
    - -

- **Naïve Bayes:**
  - **Fundamentals:**
    Core points: These are points that are at the interior of a cluster.
    2) Border points: These points' falls within the neighborhood of a core point.
    3) Noise points: A noise point is any point that is not a core point or a border point.
  - **Pros:**
    - Computationally fast
    - Simple to implement
    - Works well with high dimensions
  - **Cons:**
    - Relies on independence assumption and will perform badly if this assumption is not met
  - **Parameters:**
    - -

- **Logistic:**
  - **Fundamentals:**
    Core points: These are points that are at the interior of a cluster.
    2) Border points: These points' falls within the neighborhood of a core point.
    3) Noise points: A noise point is any point that is not a core point or a border point.
  - **Pros:**
    - low variance
    - provides probabilities for outcomes
    - works well with diagonal (feature) decision boundaries
    - NOTE: logistic regression can also be used with kernel methods
  - **Cons:**
    - high bias
  - **Parameters:**
    - -
