K Best cluster based neighbour Classification using Improved DDG approaches

Basheera.Shaik#1, G.Vijayadeep#2

1 M.Tech (CSE), Gudlavalleru Engineering College, Gudlavalleru
2 Associate Professor, Gudlavalleru Engineering College, Gudlavalleru.

ABSTRACT:

Data classification has attracted considerable research relating to computational statistics and data mining from its wide range of applications. k Nearest Neighbor (KNN) is widely used in classification datasets for dealing with the more difficult problems such as large-scale or different data categories. But KNN classifier may have a problem when training samples are uneven. The problem is that KNN classifier may decrease the precision of classification due to uneven density of training data. Clustering is applied as a possible initial step within each class to discover the n-class grouping within the dataset. Different data clustering techniques use different similarity measures which maintains its own strength and weakness. Thus, utilizing the three cluster measures may be helped by the effectiveness of each of them and eliminate problems of using an individual cluster measure. To resolve the problem, a new K Best Cluster Based Neighbour (KB-CB-N) is our present classification procedure based upon the implementation of three different similarity measures for cluster based classification. The basic is to use unsupervised learning by the instances for each class supplied in the dataset and after that use the output as an input for the classification algorithm to discover the K best neighbours of groups direct from density, gravity and distance view points. Experimental results will shows proposed algorithm gives better performance when compare to existing knn approach.

I INTRODUCTION

Data mining involves the utilization of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data set. These power tools can include statistical models, mathematical algorithm and machine learning methods. Consequently, data mining consists of more than collection and managing data, it also includes analysis and prediction.[3,5]. Most classification algorithms perform batch classification, that is, they use the dataset directly in memory. More intricate methods often build large data structures which give them a greater memory footprint. This larger footprint prevents them away from being put on the majority of the larger datasets. Even though these datasets can be utilized, the complexity of one's algorithm are able to make the classification task spend time in an inordinately period of time. There are resolutions to those problems as adding more memory or waiting longer for experiments to finish. However both feature a cost in money and time and both will possibly not ultimately solve the challenge in the event the dataset cannot in memory as soon as the memory is at a maximum. A feature of most large datasets is the lot of of data points that contain similar information. So one option would be to remove redundant information from the dataset to get a classifier to focus on data points that represent a larger group of points. This allows the building of a classier that correctly models the relationship expressed by the results in the dataset. Random sampling is not too difficult way to remove redundancy and could be very effective on uniformly distributed data[6].

Setting up a target data set: selecting a data set, or specializing in a subset of variables, or data samples, on which discovery is going to be performed. Data cleaning and preprocessing.

Removal of noise or outliers. Collecting required information to model or account for noise.
Strategies for handling missing data fields.
Discussing time sequence information and known changes[4].

Data reduction and projection.
Finding useful features to represent the results depending on the goal of the task. Using dimensionality reduction or transformation techniques reduce the effective number of variables into account and to find invariant representations for the data.

Choosing the data mining task

Deciding whether the goal of the KDD process is classification, regression, clustering, etc. Making sure you pick the data mining algorithm(s). Selecting method(s) to be utilized for searching for patterns within the data. Deciding which models and parameters can be appropriate. Matching an exact data mining task with the overall criteria as to the KDD process.

Data mining

Searching for patterns of curiosity in a particular representational form or possibly a set of such representations as classification rules or trees, regression, clustering, and so forth. Interpreting mined patterns. Consolidating discovered knowledge. Classification techniques have attracted the interest of researchers on account of the significance of their applications. Various methods encompassing decision trees, rule based methods, and neural networks are employed for the classification problems. KNN is not a demanding, but yet effective classification method. The most ideal idea is finding K nearest instances within the training sample to classify any unlabelled data instance. KNN has also been chosen by the data mining community on the list of top 10 data mining algorithms. However, there are a number of problems that decrease overall performance KNN. One of these problems that has got a clear negative impact on the classification performance of KNN is considered the utilization of standard Euclidean distance in finding the nearest neighbours. The classification decision for unlabelled samples relies upon sub-clusters in comparison to instances. That directly assists enhancing the classification performance and reducing the effect of noisy data. Moreover, that helps discover hidden patterns and categories within individual classes. The moment contribution is based on the development of all the similarity measure in KNN. While traditional KNN classification techniques typically employ Euclidean distance to assess pattern similarity and choose the nearest neighbour, other measures may be utilised to further improve the accuracy and realize the best neighbour among sub-clusters for all labelled classes in contrast to nearest one. We coined our novel technique as K Best Cluster Based Neighbour (KB-CB-N)[1].

II BACKGROUND AND RELATED WORK

Many techniques have been applied for classification, including decision trees, neural network (NN), support vector machine (SVM), K nearest neighbour (KNN) and many other techniques. K nearest neighbour has been widely used as an efficient classification model; however it has many shortcomings. Many methods have been developed to improve the KNN performance, including Weight Adjusted K-Nearest-Neighbor (WAKNN), Dynamic KNearest Neighbor (DKN), K-Nearest-Neighbor with Distance Weighted (KNNNDW)[2,5], and K-Nearest-Neighbour Naive Bayes (KNNNB). The main contributions of the above techniques are how to improve the distance similarity measure function, select neighbour size, and enhance voting system. KNN is considered an Instance Based Learning method IBL which predicts the label of little test samples by voting among K individual training instances. Alternatively, some techniques appear to have been developed to exchange the individual training samples by multitude of clusters as with [9] in an effort to improve the classification performance. However, the similarity measure applied to this class-based clustering algorithm is just distance but not considering merging other similarity measures like density and gravity. The blending of density, gravity and distance similarity measures have been first introduced in the following earlier perform [2], but particularly for clustering purposes.

Existing system Problems:

- Existing knn approach takes user defined k values in order to get k nearest dataobjects.
- Existing algorithm does not give effective results if the data contains different datatypes with missing values.
- Existing knn approach does not use clustering algorithm before classification.
- Existing clustering approaches does not use three similarity measures distance,density and gravity based measures.
- Existing algorithms does not classify if the data is unevenly distributed.

3. PROPOSED FRAMEWORK

Data Preprocessing:
Incomplete, noisy, and inconsistent data are commonplace properties of huge realworld databases and data warehouses. Incomplete data can occur and get a number of reasons. Attributes of interest might not constantly be available. Other data will not be included just because it was not considered important at the time of entry. Relevant data will not be recorded mainly because of an misunderstanding, or on account of equipment malfunctions. Data which were inconsistent with other recorded data could have been deleted[8-10]. Furthermore, the recording of one’s history or modifications to the data could possibly have been overlooked. Missing data, particularly for tuples with missing values for some attributes, may have to remain inferred. There are many possible causes for noisy data (having incorrect attribute values). The comprehensive data collection instruments used may be faulty. There might are now human or computer errors occurring at boring tasks.
Errors in data transmission can even occur. There could be technology limitations, an example would be limited buffer size for coordinating synchronized data transfer and consumption. Incorrect data may also result from inconsistencies in naming conventions or data codes used, or inconsistent formats for input fields, such as date. Duplicate tuples also require data cleaning. Data cleaning routines act to “clean” the results by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies.

Data cleaning (or data cleansing) routines effort to complete missing values, reduce noise while identifying outliers, and point out inconsistencies in the data

KB-CB-N classification algorithm is composed of two principle phases: clustering and prediction. The contributions as to the proposed algorithm can be summarized over the next couple of:

1) The notion of cluster-based classification greatly improves accuracy of prediction decision just like the classification decision is based on a set of instances instead of individual samples.
2) The clustering technique applied within each class makes a significant effect to find hidden patterns and related features among the many class objects and such consequently leads to higher classification performance.
3) Applying various similarity measures supplied in the prediction phase gives more accurate anticipation by considering not only in the distance metric, but also the distribution and size of the candidate class.
4) Voting among different candidates is over by weighted voting system which considers the rank of candidates by various similarity measures beyond just the count and size of candidate sub-clusters.

The two phases of all the proposed algorithm will just be described separately as follow:

1) Clustering phase: In the first phase, EM clustering technique is applied regarding the parts of each class. How many sub-clusters produced in each class and of course the size of each are totally depending on how distinguished are the data objects within each labelled class. The cluster-based classification technique is thought to be very efficient in detecting hidden patterns most especially when there will be definite distinguished features into each class.

2) Prediction Phase: The 2nd phase of your novel technique is the classification of this particular data objects driven by different sets of sub-clusters produced from the initial phase and presenting different classes. Within this step, the classification process is applied by using mixture of three different similarity measures (distance, density and gravity). Applying different measures when it comes to the classification purpose in contrast to only using an individual is highly significant for producing more efficient classification results.

Similarity measurements among sub-clusters KB-CB-N uses a combination of three different similarity measures for the classification purpose. For each measure, a set of candidate sub-clusters have been chosen and ranked according to each measure. Then the ranked sub-clusters from each measure are merged together and re-ranked according to their standing among different measures[1].

```plaintext
GetDist( DataObj[], SubClusters)
for i = 1 to n SubClusters do
   Calculate Euclidean distance ED between DataObj[] and SubClusters
   Assign ED to DistanceArray[i]/2
end for

GetDensity( DataObj[], SubClusters)
for i = 1 to n SubClusters do
   Calculate CurrDens of SubClusters
   Add DataObj[] to SubClusters
   Calculate ExpDens of SubClusters
   Calculate DensityGain = e^log(ExpDens - CurrDens)*10
   Assign DensityGain to DensityArray[i]
   Remove DataObj[] from SubClusters
end for
```
GetGravity(DataObj[], SubClusters)
for i = 1 to n SubClusters do
Calculate Distance Dis between DataObj[] and SubClusters centre
Calculate GravitationalForce Fg of SubClusters i Add Fg to ModGravityArr
end for

BAGGING( K Candidate Sub-Clusters)
for i = 1 to n Classes do
Count J SubClusters among k candidates assigned to class i Calculate averageRank avgRankj Of J subclusters
Calculate weightedRank WRank = avgRankj / J
if WRank ≤ tempWeight then
BestCandidate BCand = i
tempWeight = WRank
else
end if
end for
Return BestCandidate BCand.

5. EXPERIMENTAL RESULTS

All experiments were performed with the configurations Intel(R) Core(TM)2 CPU 2.13GHz, 2 GB RAM, and the operating system platform is Microsoft Windows XP Professional (SP2).

KBCBN APPROACH nearest neighbour(s) for classification

Time taken to build model: 0.3 seconds
Time taken to test model on training data: 0.13 seconds

=== Error on training data ===
Correctly Classified Instances 57 100 %
Incorrectly Classified Instances 0 0 %
Kappa statistic 1
Mean absolute error 0.0169
Root mean squared error 0.0169
Relative absolute error 3.7085 %
Root relative squared error 3.5513 %
Coverage of cases (0.95 level) 100 %
Mean rel. region size (0.95 level) 50 %
Total Number of Instances 57

=== Confusion Matrix ===
a b <-- classified as
20 0 | a = bad
0 37 | b = good

Improved KBCBN

Classifier Model

Time taken to build model: 0.31 seconds
Time taken to test model on training data: 0.29 seconds

=== Error on training data ===
Correctly Classified Instances 57 100 %
Incorrectly Classified Instances 0 0 %
Kappa statistic 1
Mean absolute error 0
Root mean squared error 0
Relative absolute error 0 %
Root relative squared error 0 %
Coverage of cases (0.95 level) 100 %
Mean rel. region size (0.95 level) 50 %
Total Number of Instances 57

=== Stratified cross-validation ===
Correctly Classified Instances 47 82.4561 %
=== Detailed Accuracy By Class ===

<table>
<thead>
<tr>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROC Area Class</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0 1 1 1 1</td>
<td>bad</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0 1 1 1 1</td>
<td>good</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Weighted Avg. 1 0 1 1 1 1

=== Confusion Matrix ===

a b <-- classified as
20 0 | a = bad
0.37 | b = good

=== Stratified cross-validation ===

Correctly Classified Instances 53 92.9825 %
Incorrectly Classified Instances 4 7.0175 %
Kappa statistic 0.8423
Mean absolute error 0.1386
Root mean squared error 0.2459
Relative absolute error 30.3197 %
Root relative squared error 51.5168 %
Coverage of cases (0.95 level) 98.2456 %
Mean rel. region size (0.95 level) 76.3158 %
Total Number of Instances 57

=== Detailed Accuracy By Class ===

<table>
<thead>
<tr>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROC Area Class</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0.108 0.833 1 0.909 0.974</td>
<td>bad</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.892 0 1 0.892 0.943 0.974</td>
<td>good</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Weighted Avg. 0.93 0.038 0.942 0.93 0.931 0.974

=== Confusion Matrix ===

a b <-- classified as
20 0 | a = bad
4.33 | b = good

Improved REPTree

@attribute duration numeric
@attribute wage-increase-first-year numeric
@attribute wage-increase-second-year numeric
@attribute wage-increase-third-year numeric
@attribute cost-of-living-adjustment {none,tcf,tc}
@attribute working-hours numeric
@attribute pension {none,ret_allw,empl_contr}
@attribute standby-pay numeric
@attribute shift-differential numeric
@attribute education-allowance {yes,no}
@attribute statutory-holidays numeric
### Classifier Model

**REPTree**

```plaintext
@attribute wage-first-year < 2.9 : bad (12/2) [5.32/1.32]
wage-first-year >= 2.9
  | cluster = cluster1 : good (13/0) [5/0]
  | cluster = cluster2 : bad (0/0) [1/0]
  | cluster = cluster3 : bad (1/0) [0/0]
  | cluster = cluster4 : good (8/0) [3/0]
  | cluster = cluster5
  | education-allowance = yes : good (2/0) [1.84/0.5]
  | education-allowance = no : bad (2/0) [2.84/1.34]
```

Size of the tree: 10

Time taken to build model: 0.28 seconds
Time taken to test model on training data: 0.04 seconds

#### Error on training data ===

- Correctly Classified Instances: 52, 91.2281%
- Incorrectly Classified Instances: 5, 8.7719%
- Kappa statistic: 0.8138
- Mean absolute error: 0.1434
- Root mean squared error: 0.2558
- Relative absolute error: 31.3686%
- Coverage of cases (0.95 level): 100%
- Mean rel. region size (0.95 level): 72.807%
- Total Number of Instances: 57

#### Confusion Matrix ===

- a b <- classified as
  - 19 1 | a = bad
  - 4 33 | b = good

#### Stratified cross-validation ===

- Correctly Classified Instances: 49, 85.9649%
- Incorrectly Classified Instances: 8, 14.0351%
- Kappa statistic: 0.6846
- Mean absolute error: 0.2091
- Root mean squared error: 0.3559
- Relative absolute error: 45.72%

### 6. CONCLUSION AND FUTURE WORK

In this paper, we have proposed, developed and evaluated our novel cluster-based K best neighbor classification method based on three new different similarity measures, namely, Moddistance, Moddensity and Modgravity. Using this unsupervised learning method for labeled classes before applying supervised learning improves detection of hidden features inside each class. Experimental results show that Robust KB-CB-N and Robust REPTree achieved better classification accuracy than several efficient classification methods. However, our New KB-CB-N uses the clustering results to perform classification.

### REFERENCES:


