Fuzzy Bio-Inspired K-Nearest Neighbor Techniques for Spatial Data Analysis in Coimbatore Region

B. Murugesakumar
Research Scholar,
Bharathiar University,
Coimbatore, Tamilnadu

Dr. K. Anandakumar
Assistant Professor(Sl.Grade),
Department of Computer Science and Engineering,
Bannari Amman Institute of Technology,
Sathyamangalam, Tamilnadu

Dr. A. Bharathi
Professor,
Department of Information Technology,
Bannari Amman Institute of Technology,
Sathyamangalam, Tamilnadu

Abstract: In this research work, agricultural Data Mining data are summarized. An improved Soil Data Prediction Model is developed to estimate the above parameters at locations for Coimbatore city. 142 locations were investigated for the development of the model. The model involves multiple regression equation, chi-square test and Bio inspired K-Nearest Neighbor classification. The correlation analysis measures the degree of association between two sets of quantitative data, while regression analysis explains the variation in one variable, based on the variation in one or more of these variables.

Keywords *: KNN, k-means, Clustering, mining algorithm, FP growth*

*****

I. INTRODUCTION

The soils of Coimbatore district are predominantly of red Loamy and black types. It is estimated that 82 percent of the area is of red soil and 18 percent of the area is a black soil. The soils of the district have been classified according to soil-taxonomy based on study of physiographic drainage, litho logy and landforms and their relationship using remote sensing data. Weather forecasting system provides us the information about future weather conditions for a particular region, locality over a specified time period. Weather directly depends upon the air molecules which can absorb high frequency sunrays. The air molecules data is collected by the system periodically after every one hour. Typically, soil temperature and moisture forecasting has been performed using a land-surface model. This is a physically-based approach that models heat transfer and moisture flow between the atmosphere and the soil subsurface. It is usually initialized with current subsurface and atmospheric conditions. The Weka tool uses these raw data to bound large information to find the “Structural weather Database”. Then the collected weather data sets are classified main database into four regions that are grouped according to the direction of wind flow over the year. Data mining involves the use of erudite data analysis tools to discover previously unidentified, suitable patterns and relationships in large data sets. Data mining tools can include statistical models, machine learning methods such as neural networks or decision trees, and mathematical algorithms. Nowadays technology enhancements improve the decision process and increases yielding in agriculture lands. But still due to huge data prediction and decision support system need more attention. However, the proposed soil and weather data analysis model predicts exact results than other model; also the proposed models are compared with exiting techniques in this research.

II. RELATED WORK

k-means Variants

Due to its versatility and simplicity, k-means has been widely adopted in different contexts. In the era of big-data, k-means has been used as a basic tool to process large-scale data of various forms. Unfortunately, as discussed in Section1, the computational cost could be prohibitively high as the scale of data increases to extraordinarily large, i.e. billion level. Recently, several k-means variants are proposed to either enhance its clustering quality or scalability. In terms of the clustering quality, one of the important work comes from S. Vassilvitskii et al. [1], [2]. The motivation is based on the observation that k-means converges to a better local optima if the initial clustering centroids are carefully selected. According to [3], k-means iteration also converges faster due to the careful selection on the initial cluster centroids. However, in order to adapt the initial centroids to the data distribution, k rounds of scanning over the data are necessary. Although the number of scanning rounds has been reduced to a few in [1], the extra computational cost is still inevitable.

K-Nearest Neighbor Graph Construction

KNN graph is primarily built to support nearest neighbor search [4], [5]. It is also the key data structure in the manifold learning and machine learning, etc [5]. Basically, it tries to find the top-_ nearest neighbors for each data point. When it is built in brute-force way, its time complexity is O(d_n2), where both d and n could be very large. As a result, it is computationally expensive to build an exact KNN graph. For this reason, recent works [6], [7], [42], [8] aim to search for an approximate but efficient solution. In [9], an approximate KNN graph is built efficiently by divide-and-conquer strategy. In this algorithm, the original dataset is partitioned into thousands of small subsets by KD trees. KNN list is built by exhaustive comparison within each subset. However, the recall of KNN graph turns out to be very low. Recent works [10], [11] could be viewed as improvements over this work. In 2011, a very successful KNN graph construction algorithm called NN Descent/KGraph [12] has been proposed. This algorithm is proposed based on the observation that "a neighbor of a neighbor is also likely to be a neighbor". According to [13], its empirical time complexity is only O(n1:14). Unfortunately, according to our observation, its recall drops dramatically as the scale of data increases to very large, i.e. 10M. Algorithm presented in [14] faces similar problem. In this paper, a novel KNN graph construction algorithm is proposed and used to support the fast k-means clustering. To the best of our knowledge, this is the first piece of work that KNN graph is used to speed-up k-means clustering. In addition, comparing with other KNN graph construction algorithms, our algorithm is computationally efficient and leads to lowest clustering distortion. Furthermore, when it is applied in ANNS problem, it shows satisfactory performance across different datasets.

III. METHODOLOGY

Expansion of Neighborhood and Distance or Similarity Metrics

We begin searching for nearest neighbors by finding the exact matches. If the number of exact matches is less than *k*, we expand the neighborhood. The expansion of the neighborhood in each dimension are done simultaneously, and continued until the number pixels in the neighborhood is greater than or equal to *k*. We develop the following two different mechanisms, corresponding to max distance and our newly defined HOB distance, for expanding the neighborhood. They have trade offs between execution time and classification accuracy.

Higher Order Bit Similarity (HOBS): We propose a new similarity metric where we consider similarity in the most significant consecutive bit positions starting from the left most bit, the highest order bit. Consider the following two values, *x*1 and *y*1, represented in binary. The 1st bit is the most significant bit and 8th bit is the least significant bit.

Bit position: 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
*x*1: 0 1 1 0 1 0 0 1 *x*1: 0 1 1 0 1 0 0 1
*y*1: 0 1 1 1 1 1 0 1 *y*2: 0 1 1 0 0 1 0 0

These two values are similar in the three most significant bit positions, 1st, 2nd and 3rd bits (011). After they differ (4th bit), we don't consider anymore lower order bit positions though *x*1 and *y*1 have identical bits in the 5th, 7th and 8th positions. Since we are looking for closeness in values, after differing in some higher order bit positions, similarity in some lower order bit is meaningless with respect to our purpose. Similarly, *x*1 and *y*2 are identical in the 4 most significant bits (0110). Therefore, according to our definition, *x*1 is closer or similar to *y*2 than to *y*1.

Definition 2. The similarity between two integers *A* and *B* is defined by

HOBS(*A*, *B*) = max{*s* | 0 < *i* < *s* -> *ai* = *bi*}

in other words, HOBS(*A*, *B*) = *s, where for all i* £ *s and* 0 £ *i, ai* = *bi and as*+1 ¹ *bs*+1.
ai and *bi* are the ith bits of *A* and *B* respectively.

Definition 3. The distance between the values *A* and *B* is defined by

$$dv(A, B) = m - \text{HOBS}(A, B)$$

where *m* is the number of bits in binary representations of the values. All values must
be represented using the same number of bits.

Definition 4. The distance between two pixels *X* and *Y* is defined by

$$d_p(X,Y) = \max_{i=1}^{n-1}\{d_v(x_i,y_i)\} = \max_{i=1}^{n-1}\{m - \text{HOBS}(x_i,y_i)\}$$

where n is the total number of bands; one of them (the last band) is the class attribute that we don't use for measuring similarity. To find the closed –KNN set, first we look for the pixels, which are identical to the target pixel in all 8 bits of all bands i.e. the pixels, *X*, having distance from the target *T*, *dp*(*X*,*T*) = 0. If, for instance, *x*1=105 (0101001*b* = 105*d*) is the target pixel, the initial neighborhood is [105, 105] ([01101001, 01101001]). If the number of matches is less than *k*, we look for the pixels, which are identical in the 7 most significant bits, not caring about the 8th bit, i.e. pixels having *dp*(*X*,*T*) 1. Therefore our expanded neighborhood is [104,105] ([01101000, 01101001] or [0110100-, 0110100-] - don't care about the 8th bit). Removing one more bit from the right, the neighborhood is [104, 107] ([011010--, 011010--] - don't care about the 7th or the 8th bit). Continuing to remove bits from the right we get intervals, [104, 111], then [96, 111] and so on. Computationally this method is very cheap. However, the expansion does not occur evenly on both sides of the target value (note: the center of the neighborhood [104,111] is (104 + 111) /2 = 107.5 but the target value is 105). Another observation is that the size of the neighborhood is expanded by powers of 2. These uneven and jump expansions include some not so similar pixels in the neighborhood keeping the classification accuracy lower. But P-tree-based closed-KNN method using this HOBS metric still outperforms KNN methods using any distance metric as well as becomes the fastest among all of these methods.

Computing the Nearest Neighbors

For HOBS: *Pi,j* is the basic P-tree for bit *j* of band *i* and *P*€ *i,j* is the complement of *Pi,j*.

Let, *bi,j* = *jth* bit of the *ith* band of the target pixel, and define *Pti,j* = *Pi,j*, if *bi,j* = 1, = *P* € *i,j*, otherwise.

We can say that the root count of *Pti,j* is the number of pixels in the training dataset having as same value as the *jth* bit of the *ith* band of the target pixel. Let, *Pvi*,1-*j* = *Pti*,1 & *Pti*,2 & *Pti*,3 & … & *Pti,j*, and *Pd*(*j*) = *Pv*1,1-*j* & *Pv*2,1- *j* & *Pv*3,1- *j* & … & *Pvn*-1,1- *j* where & is the P-tree AND operator and n is the number of bands. *Pvi*,1-*j* counts the pixels having as same bit values as the target pixel in the higher order *j* bits of *ith* band. We calculate the initial neighborhood P-tree, *Pnn* = *Pd*(8), the exact matching, considering 8-bit values. Then we calculate *Pnn* = *Pd*(7), matching in 7 higher order bits; then Then *Pnn* = *Pd*(6) and so on. We continue as long as root count of *Pnn* is less than *k*. *Pnn* represents closed-KNN set and the root count of *Pnn* is the number of the nearest pixels. A 1 bit in *Pnn* for a pixel means that pixel is in closed-KNN set.

The algorithm for finding nearest neighbors is given below.

```
Input: Pi,j for all bit i and band j, the basic P-trees
and bi,j for all i and j, the bits for the target pixels
Output: Pnn, the P-tree representing closed-KNN
// n - # of bands, m - # of bits in each band
FOR i = 1 TO n-1 D
FOR j = 1 TO m DO
IF bi,j = 1 Ptij ← Pi,j
ELSE Pti,j ←P€i,j
FOR i = 1 TO n-1 DO
Pvi,1 ←Pti,1
FOR j = 2 TO m DO
Pvi,j Pvi,j-1 & Pti,j
s ←m
REPEAT
Pnn ←Pv1,s
FOR r = 2 TO n-1 DO
Pnn _ Pnn & Pvr,s
s ←s - 1
UNTIL RootCount(Pnn) ³ k
```

Improved k-means Driven by Bio-Inspired KNN Graph

Given a KNN graph is ready, Bio-inspired k-means procedure presented is revised as Alg. 2. At the beginning of the clustering, 2M tree (Alg. 1) is called to produce k clusters. The initial clusters will be incrementally optimized in the later steps. In each step of the optimization iteration, one sample is randomly selected. Thereafter, all the clusters in which its k neighbors reside are collected. The selected sample is therefore checked with these clusters to seek for the best move. The iteration terminates until convergence condition is reached.

```
Algorithm 2. BKNA-means($X_{nxd}$, k, $G_{nxn}$)
1: Input: matrix Xn_d, k, KNN graph $G_{nxn}$
2: Output: S1; …; Sr;….Sk
3: cLabel = TwoMeans($X_{nxd}$, k);
4: Q ←Q;
5: while not convergence do
6: for each xi € X do
7: for j = 1; j <k; do
8: b = G[i][j];
9: Q ⟶ Q ʋ[ cLabel[b];
10: j = j + 1;
11: end for
12: Seek v in Q that maximizes ΔI(xi);
13: if ΔI(xi) > 0 then
14: Move xi from current cluster to Sv;
15: end if
16: Q ←Q;
17: end for
18: end while
```

It is considerably faster than traditional k-means initialization. Secondly, as shown from Line 6-12, only clusters that keep the first Δ neighbors of x_i are visited, the number of which is much smaller than k. Furthermore, it is possible that several neighbors of xi may live in the same cluster. As a consequence, the number of clusters that one sample visits is even smaller than k.

Accuracy of the Proposed Bio-Inspired K-Nearest Neighbor Prediction Algorithm (BKNA)

The accuracy of the bio-inspired k-nearest neighbor predictive algorithm provides the amount of generated prediction is similar to actual outcomes. Therefore that can also be defines as the amount of correctly recognized patterns among the total samples produces to test. That can also be evaluated using the following formula:

$$\text{Accuracy} = \frac{Total\ \ correctly\ \ identified\ \ patterns\ \ of\ datasets}{Total\ \ input\ \ data} \times 100$$

The figure 1 and table 1 contains the evaluated performance of the system in terms of bio-inspired k-nearest neighbor algorithm accuracy of prediction. In this diagram the amount of accurate pattern recognition is given using Y axis and the X axis shows the amount of data to be train for the predictor. According to the comparative results the performance of the proposed algorithms remains much consistent and increasing as compared to the traditional algorithm. In order to evaluate the accuracy of algorithm a fixed amount random patterns are extracted from database and the next values are used as the actual class label during evaluation of data. Finally, the proposed technique predicts best accuracy when compared to existing methods.

Table 1: Accuracy of Comparative data sets

Dataset size	BKNA(Proposed)	FP Growth Existing Techniques
100	76.63	66.17
300	79.25	69.54
600	80.36	71.45
1000	82.29	72.48
1500	84.51	74.51
2500	86.28	76.77
3000	89.74	79.85

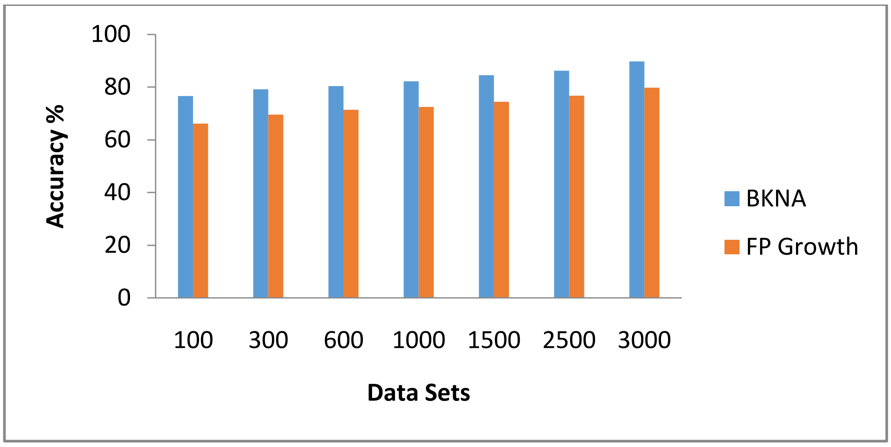
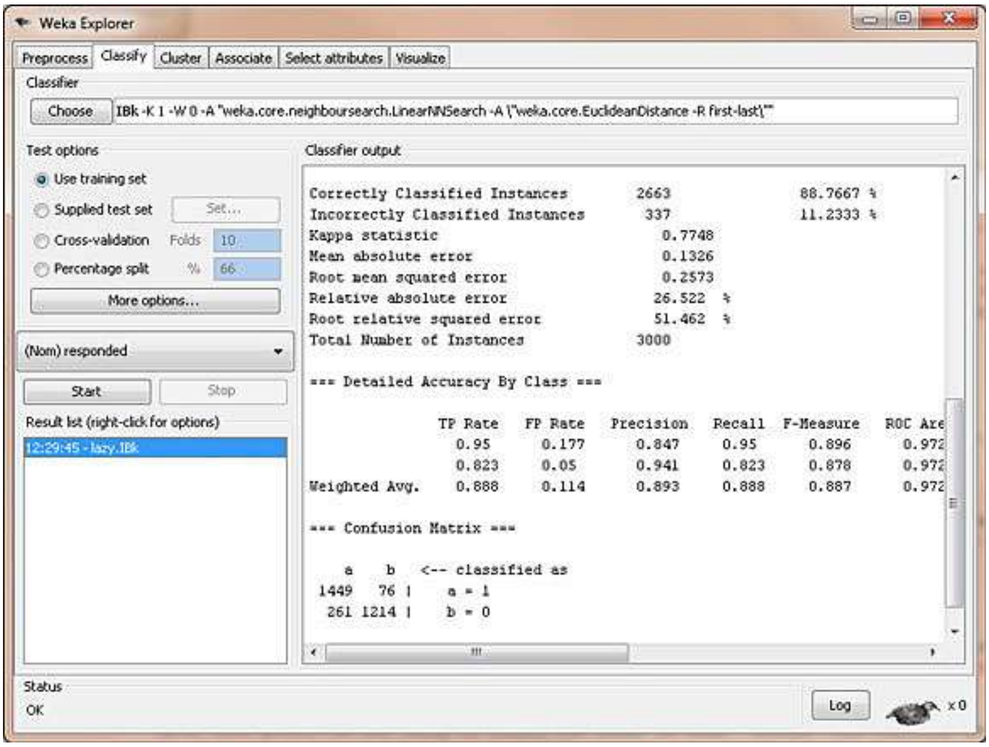


Figure 1: Accuracy of Comparative data sets of BKNA vs FP Growth



IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Test options

● Use training set

○ Supplied test set Set...

○ Cross-validation Folds 10

○ Percentage split % 66

Result list (right-click for options)

12:29:45 - lazy.IBk

Classifier output

```
Correctly Classified Instances        2663               88.7667 %
Incorrectly Classified Instances       337               11.2333 %
Kappa statistic                          0.7748
Mean absolute error                      0.1326
Root mean squared error                  0.2573
Relative absolute error                 26.522  %
Root relative squared error             51.462  %
Total Number of Instances             3000

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Are
                 0.95      0.177      0.847     0.95      0.896      0.972
                 0.823     0.05       0.941     0.823     0.878      0.972
Weighted Avg.    0.888     0.114      0.893     0.888     0.887      0.972

=== Confusion Matrix ===

    a    b   <-- classified as
 1449   76 |    a = 1
  261 1214 |    b = 0
```

Status

OK

Figure 2: Accuracy rate of Cluster data sets

IV. CONCLUSION

From the obtained results, This research work implemented various data mining techniques in agricultural data sets and implementation results will helpful for predicting the soil and weather condition. The bio-inspired k-nearest neighbor prediction classification algorithm for agricultural data compared with existing FP Growth cluster algorithm, the proposed technique predicts accurate soil and weather condition in the Coimbatore region. Thus, the study showed that, obtained prediction rate is very high in the proposed method compared to the existing methods.

References

[1] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii,“Scalable k-means++,” In Proceedings of the VLDB Endowment,vol. 5, no. 7, pp. 622–633, 2012.

[2] A. Broder, L. Garcia-Pueyo, V. Josifovski, S. Vassilvitskii, and S. Venkatesan, “Scalable k-means by ranked retrieval,” in Proceedings of the 7th ACM international conference on Web search and data mining, pp. 233–242, 2014.

[3] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching,” inCVPR, pp. 1–8, Jun. 2007.

[4] Y. Avrithis, “Quantize and conquer: A dimensionality-recursive solution to clustering, vector quantization, and image retrieval,” in Proceedings of the IEEE International Conference on Computer Vision, pp. 3024–3031, Dec. 2013.

[5] Y. Avrithis and Y. Kalantidis, “Approximate gaussian mixtures for large scale vocabularies,” in ECCV, pp. 15–28, 2012.

[6] Y. Avrithis, Y. Kalantidis, E. Anagnostopoulos, and I. Z. Emiris, “Web-scale image clustering revisited,” in ICCV, pp. 1502–1510, Dec. 2015.

[7] J. Wang, J. Wang, Q. Ke, G. Zeng, and S. Li, “Fast approximate k-means via cluster closures,” in CVPR, pp. 3037–3044, Jun. 2012.

[8] C. Elkan, “Using the triangle inequality to accelerate,” in ICML, 2013.

[9] H. J´egou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” Trans. PAMI, vol. 33, pp. 117–128, Jan. 2011.

[10] N. Verma, S. Kpotufe, and S. Dasgupta, “Which spatial partition trees are adaptive to intrinsic dimension?,” in Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 565–574, Jun. 2009.

[11] W. Dong, C. Moses, and K. Li, “Efficient k-nearest neighbor graph construction for generic similarity measures,” in WWW, pp. 577– 586, Mar. 2011.

[12] C. Fu and D. Cai, “EFANNA: An extremely fast approximate nearest neighbor search algorithm based on knn graph,” arXiv preprint arXiv:1609.07228, 2016.

[13] Y. A. Malkov and D. Yashunin, “Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs,” arXiv preprint arXiv:1603.09320, 2016.

[14] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y.Wu, “An efficient k-means clustering algorithm: analysis and implementation,” Trans. PAMI, vol. 24, pp. 881–892, Jun. 2002.

[15] A. Broder, L. Garcia-Pueyo, V. Josifovski, S. Vassilvitskii, and S. Venkatesan, “Scalable k-means by ranked retrieval,” in Proceedings of the 7th ACM international conference on Web search and data mining, pp. 233–242, Feb. 2014.

[16] A. Goswami, R. Jin, and G. Agrawal, “Fast and exact out-of-core kmeans clustering,” in Fourth IEEE International Conference on Data Mining, pp. 83–90, Nov. 2004.

[17] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data. 1988.

[18] Y. Zhao and G. Karypis, “Hierarchical clustering algorithms for document datasets,” Data Mining and Knowledge Discovery, vol. 10, pp. 141–168, Mar. 2005.

[19] J. Chen, H. ren Fang, and Yousef, “Fast approximate knn graph construction for high dimensional data via recursive lanczos bisection,” Journal of Machine Learning Research, vol. 10, pp. 1989–2012, Dec. 2009.