Threshold Computation to Discover Cluster Structure, a New Approach

Preeti Mulay
Department of CS & IT, Symbiosis Institute of Technology, Symbiosis International University, Pune, India

Article Info

Article history:

Received Feb 6, 2015
Revised Nov 18, 2015
Accepted Dec 2, 2015

Keyword:

Closness factor
Cluster
Evaluation measures
Incremental-clustering
Threshold

ABSTRACT

Cluster structure formation is still one of the research areas open for researchers. The "NoSQL" concept opened a new arena for innovations. In this paper a comparative study about forming cluster structure is discussed and this discussion is based on selecting an optimal threshold value to form a cluster. Threshold selection continues to play important role in post cluster phase as well, to accommodate influx of new data. This paper consists of a new incremental-clustering approach (ICNBCF), various possibilities of threshold computation and evaluation measures.

Corresponding Author:

Preeti Mulay,
Department of CS & IT, Symbiosis Institute of Technology,
Symbiosis International University, Lavale, Pune 412 115, India
Email: Preeti.mulay@sitpune.edu.in

1. INTRODUCTION

With the spurt of data in all domains (almost), it is essential to have modernized data exploratory methods, like incremental-clustering, cluster analysis, incremental-learning etc. to name a few. These methods are useful in varied applications [1] which require handling influx of new data consistently and to perform forecasting, decision making and predictions. These application domains include finance [2], biology [3], feedback analysis [4], sensitivity analysis [5], electricity power consumption etc.

The purpose of this research paper is to broaden the abilities of "Incremental clustering using Naïve Bays and Closeness-Factor" (ICNBCF) [6] algorithm, and introduce set of activities at post-clustering phase. These activities include validating cluster structures. These modifications proved the enhancements in resulting cluster structures. ICNBCF has already proved by implementing parameter-free data-clustering algorithm. UCI repository's Wine, WineQuality, Electricity, Software Project and Zenni Optics datasets are used to perform experiments. The resulting clusters are evaluated by four analytical measures: f-measure, Rand index, Variance and Dunn index.

2. RESEARCH METHOD

ICNBCF incremental clustering algorithms is executed by using Microsoft Visual Studio 2005 and Eclipse Java platform on an Intel® CoreTM i5 CPU M 450 @ 2.40GHz, 4GB RAM computer sytem.

The novel method based on "cluster-first" approach, almost parameter-free, error-based statistical incremental clustering method [1] is extended in this paper. To execute ICNBCF for the first time on new data set, pre-processing or pre-clustering step may be computed, (PCA). ICNBCF generates basic clusters, at first and then either updates clusters or generates new clusters based on inflow of new data. ICNBCF works in three phases. In the first phase initial clusters are built. Once the basic clusters are ready, with influx of new information, existing cluster is updated or new cluster is formed. A new cluster is generated if the behavior of input data is entirely different from the previously generated clusters and their members, that means there exists threshold difference. ICNBCF does not use the (external) distance measure for determining the relative closeness between the series. It makes use of the novel approach “closeness factor”. The method needs threshold value to be set which is used as limit while forming clusters. The clusters are generated dynamically and results show that with good choice of threshold value, ICNBCF has assets that are valuable for analyst. ICNBCF is an extension of original CFBA algorithm [1]. The statistical details of CFBA algorithm are as shown below:

CFBA algorithm:

In this algorithm capital C and G are used interchangeably. Read the input csv file, using a CSV Parser – store the contents in a Vector of “Dataset” objects

```
1. Read all the closeness parameters:
   a. New or incremental
   b. Number of Series to use to calculate the “g” values – default is 2
2. If method is New, then
   a. Method = New
      i. CalculateG()
         ➢ For all the series,
            a. S1=S(i), S2=S(i+1)
            b. Calculate the sum of each column $T \ni T(j) = S_1(j) + S_2(j)$
            c. Calculate the sum of each series – this will be used to calculate the Probability ratio.
               $p = \frac{\sum_{j=1}^{n} S_1(j)}{\sum_{j=1}^{n} T(j)}$
            d. Calculate the Error for each series, $c(j) = \frac{p \times T(j) - S_i(j)}{\sqrt{T(j) \times p \times (1-p)}}$
            e. Calculate the Weight of each series, $w(j) = \sqrt{T(j)}$
            f. Calculate G for these two series, $G(j) = \frac{\sum_{j=1}^{n} c(j)^2 \times w(j)}{\sum_{j=1}^{n} w(j)}$
            g. Store the p, G values for each series by adding two columns to the end of the series.
            h. Next i
      ii. CreateClusters()
         ➢ Create clusters using the closeness method
         ➢ For all the series “i” to “n”,
            a. Get the g value of S(i)
            b. Check for Series_processed_flag(Boolean) –
               i. if flag=true, ignore series, as its already part of a cluster. Continue the for-loop
                  for the next element
               ii. Else CreateCluster and add this series as part of the cluster
            c. For all the series “j=i+1” to “n”
               i. Get the g value of S(j)
               ii. If (S(i) – S(j) < closeness_factor)
                  1. Add S(j) to the Cluster
                  2. Set Series_processed_flag to true for Series(j)
               iii. Continue for next element of Series(j)
            d. Continue for the next element of Series(i)
         ➢ At the end of the for loop, all series should be part of some cluster, and all the series
           should have the “Series_processed_flag” to true.

   b. Else, Method = Incremental
      i. Read the Results file, which contains the clusters and its elements, in memory
      ii. CalculateG() – step 3.a.i
```

```
        iii. UpdateClusers()
            ➢ For each existing clusters,
                a. Get each Series in this cluster, S(i)
                b. Get the g value of S(i)
                c. For each newly added Series
                    i. Get the Series S(j)
                    ii. Get Series_processed_flag. If true, ignore series, and go to step v.
                    iii. Get the g value of S(j)
                    iv. If ( S(i) – S(j) < closeness_factor)
                        1. Add S(j) to cluster,
                        2. Set the Series_processed_flag to true
                    v. Continue to the next Series
                d. Continue to the next cluster.
            ➢ Check if the Series_process_flag is set for all the incremental elements
            ➢ For all the incremental series
                a. Follow steps in 3.a.ii
3. Write Output in the output file
4. End.
```

ICNBCF is an incremental clustering method most suitable for quantitative datasets only. The essential statistical computations involved are as follows:

1. Accept raw numeric dataset
2. Apply pre-processing like remove zeros, PCA etc. if required
3. Select two dataseries say DS_1 and DS_2
4. Compute their row-total T(j), where j=1,2
5. Compute every column-total / attribute total $T_i(j)$, where i varies from 1 to N, N is number of attributes.
6. Compute grand-total = $T(DS_1)$ and $T(DS_2)$
7. The probability that DS_1 and DS_2 will be part of same cluster is computed using
8. $\mathrm{P(j)} = \frac{\sum_{j=1}^{2} \mathrm{DS(j)}}{\mathrm{grand{-}total}}$ $\quad \rho = \frac{\mathrm{P(j)}}{1-\mathrm{P(j)}}$
9. Expected value of dataseries is given *by* $< \mathrm{Ds(j)} >= \ \rho \ * \mathrm{grand} - \mathrm{total}$
10. Error is computed by following formula: $\mathrm{error} = \frac{<Ds(\mathrm{j})> -DS(j)}{\sqrt{\mathrm{column{-}total}* \rho*(1-\rho)^2}}$
11. Weights of individual attributes is computed using
12. $\mathrm{w(i)} = \sqrt{\mathrm{column}} - \mathrm{total}$
13. The closeness factor [2] is computed using error and weights, as
14. $\mathrm{closeness} - \mathrm{factor\ (j)} = \frac{\sum_{j=1}^{2} \mathrm{DS(j)}}{\sum_{i=1}^{N} \mathrm{w(i)}}$

This computed closeness-factor is referred hereafter as CF. Once the algorithm calculates all CF values; it is the time to decide the cluster structure. CF values guide users which data series are close to each other and can be a part of same cluster. Once the decision of two close data series is made, it is also essential to know whether the selected data series have matching events based on selected features or not? To achieve this modified Naïve Bayes method is applied based on CF values. The combination of CF and Naïve Bayes method proved to be another way to confirm Principal Components of given dataset.

Naïve Bayes classifier is a term in Bayesian statistics dealing with simple probabilities based clustering based on applying Bayes' Theorem with naïve independent assumptions. "Independent feature-based model" is more suitable similar term used for probability based models. Naïve Bayes method assumes that the presence or absence of a particular feature of a class is unrelated to presence or absence of any other feature. In many practical applications maximum likelihood approach is used for parameter-based estimations using Naïve Bayes'. An advantage of using Naïve Bayes' method is that it requires a small amount of training data to estimate the parameters necessary for finding whether the two given data series are close to each other based on particularly selected feature or not. In ICNBCF, CF values are used instead of mean and variance of variables, as in original method, as the CF value is computed based on entire set of attributes. Naïve Bayes' based method is fast and incremental and deal with discrete and continuous attributes. The comparison of performance in various domains confirms the advantages of successive learning and suggests its application to other learning algorithms. In the Bayesian approach, the task corresponds to finding class label y that based on selected impactful features that maximizes the probability that the two data series are found close to each other and their specific events also match.

Let x=(x1,x2,…..xd) be the set of attribute values for an unlabeled instance z=(x,y). The posterior / matching probabilities for y given x can be computed using the Bayes theorem as:

$$P(y|x) = P(Y|x1, x2, \ldots..xd) = \frac{P(x1, x2 \ldots..xd) * P(y)}{P(x1, x2 \ldots..xd)}$$

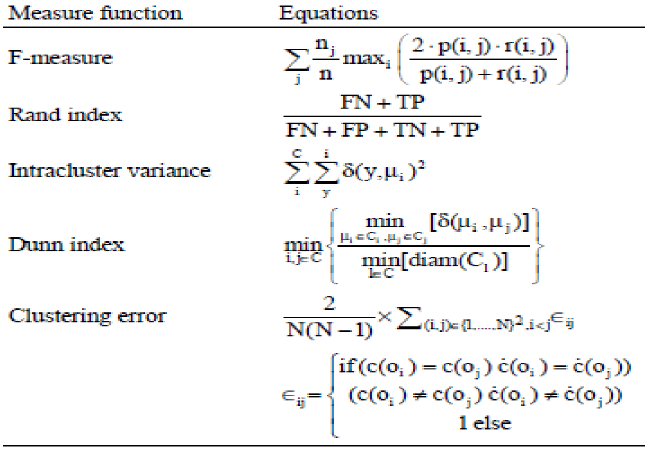
Since we are interested in comparing the posterior or matching probabilities for different values of y, we can simply ignore the denominator term. The difficult part is to determine the conditional probabilities P(x1,x2,…….xd|y) for every possible cluster. A Naïve Bayes' method attempts to resolve this by making additional assumptions regarding the nature of relationships among the given attributes. It assumes that attributes are conditionally independent of each other when class label y is known. In other words $P(a_i\ a_j\,|y) = P(a_i|y) * P(\ a_j\,|y)$ for all i's and j's, therefore, $P(x1, x2, \ldots\ldots..xd|y) = \prod_{i=1}^{d} P(\ x_i\quad |y)$ This equation is more practical because instead of computing the conditional probabilities for every possible combination of x given y, we only have to estimate the conditional probabilities for each pair P(Xi|y). The CF values calculation has already taken care of finding data series which are close to each other, only left out part is to concentrate on specific feature or event for more effectual closeness based incremental-clustering.

To cluster based on instance z=(x,y) the naïve bayes method computes the posterior probability of y given x using $\prod_{i=1}^{d} P(x_i|y)P(y)$ and selects the value of y that maximizes this product. This way incremental clustering is obtained using the combination of closeness factor and Bays theorem.

The crucial task involved while designing new incremental clustering algorithm ICNBCF based on CF and Naive Bays approach [6] is to decide the threshold for forming cluster structures. In the initial first phase cluster structure is created from completely raw data and by performing Principal Component Analysis (PCA) as pre-clustering stage.

In the second phase of forming cluster structure, analytical evaluation measures such as maximum Dunn Index, f-measure, variance, mean (□) and standard deviation (□) is computed. The table 7 shows the complete comparison using these evaluation measures on various standard quantitative datasets such as UCI Machine Learning Repository's Wine, Wine Quality, Zenni Optics, Software Project and Electricity. The table no. 3 shows the evaluation measure used, and their formulas.

Table 3. Evaluations measures



Measure function	Equations
F-measure	$\sum_j \frac{n_j}{n} \max_i \left(\frac{2 \cdot p(i,j) \cdot r(i,j)}{p(i,j) + r(i,j)} \right)$
Rand index	$\frac{FN + TP}{FN + FP + TN + TP}$
Intracluster variance	$\sum_i^C \sum_y^i \delta(y, \mu_i)^2$
Dunn index	$\min_{i,j \in C} \left\{ \frac{\min_{\mu_i \in C_i, \mu_j \in C_j} [\delta(\mu_i, \mu_j)]}{\min_{l \in C}[\mathrm{diam}(C_l)]} \right\}$
Clustering error	$\frac{2}{N(N-1)} \times \sum_{(i,j) \in \{1,\ldots,N\}^2, i<j} \in_{ij}$
	$\in_{ij} = \begin{cases} \text{if}(c(o_i) = c(o_j)\ \dot{c}(o_i) = \dot{c}(o_j)) \\ (c(o_i) \neq c(o_j)\ \dot{c}(o_i) \neq \dot{c}(o_j)) \\ 1 \text{ else} \end{cases}$

External evaluation measures include number of clusters having maximum f-measure and Rand index and internal measures are variance and the Dunn index. In addition, the number of clusters and cluster error rate are also obtained to measure cluster results.

The test experiment of the ICNBCF incremental cluster algorithm is conducted by running the algorithm fifty times simulating influx of data, to validate incremental-clustering, for each of the five real datasets: Wine, WineQuality, Software Projects, Zenni Optics and Electricity.

The reason behind using UCI Machine Learning Repositories is multifold, one is classes are already known and all of them are validated datasets. As the classes are already known, it becomes easier to compare the results given by researched algorithm, for ex. Wine dataset contains three classes and WineQuality contains six classes etc. The evaluation measures witnessed Wine achieving a mean of 2.98 for three clusters and WineQuality 4.26 for six clusters. ICNBCF is competent to find outliers as well as duplicates, almost duplicates from given data. Hence it was observed that WineQuality dataset also contains 10 wines of 9th quality, and four impactful features, instead of three, which was published. Evaluation results also witnessed worst results while clustering Zenni Optics, with a mean of 1.06, and detected only one cluster initially. Linear or semi-correlated dataset perform best with ICNBCF, while highly correlated classes shows degraded performance.

Table 4. Results showing clusters found, cluster error and total execution time.

Datasets	Clusters found		Clustering Error		Total execution Time	
	μ	σ	μ	σ	μ	σ
Wine	2,98	0,25	0,14	0,07	0,74	0,44
Electricity	3,98	0,77	0,32	0,05	0,48	0,38
Software Project	2,36	0,48	0,37	0,05	2,32	0,63
Zenni Optics	1,06	0,24	0,88	0,03	1,4	0,56
WineQuality	4,26	0,44	0,01	0,01	0,52	0,54

The total execution time recorded for individual dataset depends on size of dataset given as input for finding appropriate cluster and for accommodating new data series with influx of data. As shown in the table no. 4, runtime for small electricity data is 0.48 and 2.32 for largest software project dataset. It is also proved that this algorithm is stable while processing single record from input dataset. The average difference recorded is □0.0052 ((∑μ_time) / Nmax) / (DatasetNumber). The maximum average distribution equals 0.63, recorded for the Software Project dataset and the minimum mean distribution, recorded for the Electricity dataset, is 0.38.

Table 5. Comparison of wine and Zenni Optics by external measures

Datasets	F-measure		Rand index	
	μ	σ	μ	σ
Wine	0,84	0,0503	0,84	0,03
Zenni Optics	0,7	0,0127	0,56	0,05

The evaluation measures, namely f-measure and Rand Index using Wine and Zenni Optics dataset is given in table no. 5 f-measure using Wine dataset is 0.84 which shows the compactness of cluster members. The Rand Index value of 0.84 shows different perspective due to dense data distribution related to some data series and varied distribution at other end of data file. Zenni Optics is a non-linear dataset and records f-measure (μ= 0.7) and the Rand index (μ = 0.56).

Table 6. Comparison of Electricity and Software Project by intra measures

Datasets	Variance		Dunn index	
Software Project	0.85	0.07	11.94	2.87
Electricity	0.38	0.01	9.68	4.81

Variance and Dunn Index computation based on Software Project and Electricity data set is shown in table 6. It is observed that the variance computed of Software Project data is high almost equal to computed for Wine dataset. By observing the values computed using Electricity data set, it is clear that most data is gathered around mean closely. In addition, the average Dunn index value for the Electricity data is 9.68 with a high separation rate of 4.81. The algorithm attains the optimum run for Electricity as it achieves a zero error rate for about 80% of fifty independent runs and the associated variance and Dunn index have values of 0.38 and 12.71 respectively. It is observed that the computations given by evaluation measures indicate successful combining of similar data members in the form of clusters.

To further evaluate the performance of ICNBCF, in addition to evaluation measures, other incremental clustering algorithms namely COBWEB, I k-means is used [7].

From the comparison table no. 7, it is visible that ICNBCF achieved best value of correct clusters as given in published results and one additional cluster showing higher quality wine in WineQuality dataset. Variance value=0.35, cluster error=0.08, given by ICNBCF shows inner cluster compactness. The f-measure = 0.96 which is obtained in 50% of total iterations, again one more check for proposed incremental clustering. As shown in the table no. 7, I k-means, COBWEB and k-means have f-measures=0.96, 0.85 and 0.96 respectively, the worst error value is that of COBWEB (0.2969). ICNBCF achieves highest intra-cluster variance which is 0.77, and low cluster error value, only just 0.22. Incremental version of k-means provides 0.24 intra-cluster variance, which is lowest, still does not provide correct set of cluster members. The cluster error computed for I k-means is very high, .089. Hence, it is proved that ICNBCF is most suitable incremental-clustering method for Wine dataset. The graphical comparison using f-measure, variance and clustering error, along with 2% moving average based on f-measure is shown in figure 1 below.

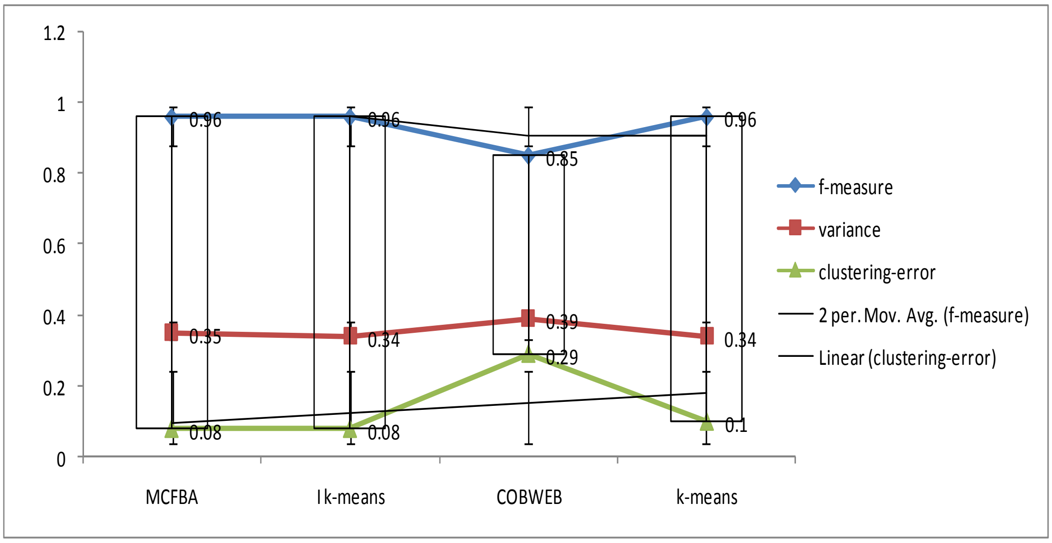


Figure 1. Showing comparison of four data-clustering algorithms

The best Dunn index value obtained for the Software Project dataset by ICNBCF clustering was 13.83. The I k-means, COBWEB and k-means, on the other hand, failed to attain the value in any of their runs. For the Software Project dataset, ICNBCF clustering was able to obtain the lowest clustering error value (0.29) corresponding to two clusters and a value of 0.34 corresponding to three-cluster groups; while the other algorithms recorded clustering errors near to a value of one such as the I k-means and thus this model generated too many clusters in most of the runs.

Table 7. Comparison of Performance

[redacted]

The results using Zenni Optics data is as shown in Table 7; the f-measure = 0.67, Rand index = 0.55, variance = 1.12, Dunn index=15.42 and clustering error=0.89 using ICNBCF algorithm. I k-means has visibly obtained better results than ICNBCF, where the f-measure was 0.70; but there were poorer results by k-means f-measure= 0.60 and COBWEB: f-measure=0.66.

For Electricity, a small data set composed of 75 data objects with four clusters, all algorithms gave optimal results, mostly similar. Their f-measure is one, clustering error is zero and variance is 0.37 (Table 7). The I k-means, however, attains inferior results since its f-measure= 0.46, has a variance=0.63 and clustering error=0.68.

3. CONCLUSION

In this paper various computations using evaluation measures such as f-measure, Dunn Index, Rand Index along with cluster error is discussed. This discussion was based on various incremental clustering algorithms such as ICNBCF, Incremental k-means, COBWEB and k-means being most pioneered data-clustering algorithm. Experimental results showed that ICNBCF is comparable to the other clustering algorithms in terms of validity measure. Moreover, the method has achieved a higher degree of clustering accuracy for some datasets.

In the initial part of this paper, computation of cluster threshold practice is discussed based on normalization. The solutions discussed in this paper provide another option for manual “threshold selection” by user having knowledge of entire data set.

As a part of future work, ICNBCF is achieving interesting behavioural results using various datasets including world-wide ice-cream dataset, beer dataset, wine dataset revisited from perspective of combining wine with food intake together, cosmetics dataset based on skin-type, age, allergies, fat contents etc., cheese dataset, to name a few as primary products which can be consumed directly, along with some secondary datasets like tea and coffee datasets which needs mediums to consume and taste.

REFERENCES

[1] Mooi, E.A. and M. Sarstedt, 2011. A Concise Guide to Market Research: The Process, Data and Methods Using IBM SPSS Statistics. 1st Edn., Springer, Berlin, ISBN-10: 3642125417, pp: 307.

[2] Cai, F., N.A. Le-Khac and M.T. Kechadi, 2012.Clustering approaches for financial dataanalysis: A survey. Proceedings of the 8th International Conference on Data Mining, (DM’ 12), Las Vegas, Nevada, USA, pp: 105-111.

[3] Nazeer, K.A., M. Sebastian and S.M. Kumar, 2013. A novel harmony search-K means hybrid algorithm for clustering gene expression data. Bioinformation, 9:84-88. DOI: 10.6026/97320630009084.

[4] Inkaya, T., 2011. A methodology of swarm intelligence application in clustering based on neighbourhood construction. The Graduate School of Natural and Applied Sciences of Middle East Technical University.
[5] William Claster, "Wine Tasting and a Novel Approach to Cluster Analysis", 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation.
[6] "Evolve systems using incremental clustering approach", Preeti Mulay, Dr.Parag A. Kulkarni Evolving Systems, An Interdisciplinary Journal for Advanced Science and Technology, Journal No. 12530 by Springer, Oct 2012.
[7] Jain, A.K. and S. Maheswari, 2012. Survey of recent clustering techniques in data mining. Int. J. Comput.Sci. Manage. Res., 1: 72-78.

BIOGRAPHY OF AUTHOR

Preeti Mulay did her M.S from WSU, MI, USA, M.Tech from JNTU, Hyderabad India and PhD from BVU, Pune. She is associated with Symbiosis International University from 2013. Her areas of interest include machine learning, data mining, software engineering and knowledge augmentation.