Clustering Embedded with Context Awareness using an Evolutionary Approach

Sanjeevani Dhaneshwar
Department of Computer Engineering,
mt. Kashibai Navale College of Engineering,
Vadgaon Budruk, Pune, India

Manisha R. Patil
Department of Computer Engineering,
Smt. Kashibai Navale College of Engineering,
Vadgaon Budruk, Pune, India

ABSTRACT

The research presented in this paper explores the embedding of context awareness into a data mining method called clustering. Adding context to traditional data mining methods has been known to improve results of information retrieval systems. The approach used for this task is that of Multi Objective Evolutionary Algorithms. Evolutionary algorithms imitate the biological process of natural selection, also known as survival of the fittest, to solve computational problems. It is a heuristic method that finds approximate solutions. The solutions are generally optimized with respect to some system objective. However, many practical problems require optimization in more than one and possibly conflicting objectives. Multi Objective Evolutionary Algorithms (MOEA) are used for this purpose.

General Terms

Data Mining, Clustering, Context Awareness

Keywords

Multi Objective Optimization, Evolutionary Algorithms, Data Mining, Clustering, Context Awareness

1. INTRODUCTION

With the explosion of data produced by organizations and individuals, large repositories of raw data have been created. Interesting patterns can be drawn from this raw data to find out business beneficial information that may provide the organization with competitive advantage. For example, the purchase history of users on an online shopping portal may be mined to recognize shopping trends among particular groups of users. Characteristics of such groups may be identified and harnessed for targeted marketing.

Clustering is a well-known data mining technique. It has a wide application in the fields that require grouping together of similar data. The research presented, attempts to approach the method of clustering in a heuristic manner. The aim is not just to find the best clusters, but to find a set of cluster configurations for the given data set that have additional information embedded in them. This additional information is called context. Context awareness is described in detail in the further paragraphs.

The upcoming sections describe related work, problem statement, proposed solution, mathematical modelling and proposed algorithm.

2. RELATED WORK

2.1 Context Awareness

The results of data mining can be further improved by exploring the relationship that is intrinsic between the data objects. Data generated as a result of any operation is always influenced by the environmental factors that it performs the operation in. The object producing data is said to be operating in a context. Adding context awareness to data mining processes has been extensively used to improve their accuracy. Adomavicius et al. (2011) describe the notion of context as a frame for a given object. This frame contains elements or factors that influence the object and the activities that it performs [1].

For example, the choice of a dress bought by a woman will depend on the occasion that she is buying it for. The factor influencing her choice in this case is the occasion for which she will be wearing it. The question asked is ‘why is the dress being bought?’ Some other questions that may be asked are ‘what’, ‘where’, ‘how’, ‘when’, ’who’ and so on. These translate into contextual factors like location, time of the day, day of the week, month, purpose of purchase, etc.

2.2 Evolutionary Algorithms

There have been numerous instances wherein certain hard computation problems have been solved by inspiration from nature. This is called as bio mimicry. A prominent example is neural networks that are modelled after the working of the human brain. Another example is evolutionary algorithms. They are modelled after Darwin’s theory of Survival of the Fittest.

The generic algorithm follows these basic steps: [2]

1. Randomly generate a population of N chromosomes. Evaluate the fitness of these chromosomes according to the pre-decided fitness function
2. Crossover: Generate new offspring by combining attributes from parents with high fitness value. This step helps in propagating the good features or traits of one generation to the next.
3. Mutation: Randomly modify the value of an attribute of a chromosome. The number of mutations is determined by the mutation rate. This step helps the algorithm escape getting stuck in a local optimum and expands the search area to obtain a globally optimal solution
4. Fitness Assignment: Evaluate each chromosome with respect to the objective function and assign a fitness value to it.
5. Select N best chromosomes (now solutions) and add them to the next generation.
6. If stopping criterion is satisfied terminate the algorithm. Else, mark the start of a new generation and repeat from step 2.

Using an evolutionary algorithm is like a black box approach to solving a problem. It is generally used when we do not have enough information about the problem. So instead of trying to solve it, we generate a number of possible solutions, and look for the best one.

2.3 Multi-Objective Optimization

The approach described in the section above finds solutions that are optimal only with respect to a single objective, like performance, accuracy, etc. [3]

However, more often than not, in many real-life problems, objectives under consideration conflict with each other. For example, there are 2 concepts of exploration and exploitation in the field of recommendation systems. Exploitation leverages known information about the user's choices to produce new recommendations. It focuses on accuracy of recommendations. Whereas exploitation brings about diversity. It introduces the user to new content, with the risk that the user may not like it. Thus, exploration focuses on novelty of recommendations. Both are conflicting goals - accuracy and novelty. Hence, optimizing a solution with respect to a single objective often results in unacceptable results with respect to the other objectives.

Therefore, a perfect multi-objective solution that simultaneously optimizes each objective function is almost impossible. A reasonable solution to a multi objective problem is to investigate a set of solutions, each of which satisfies the objectives at an acceptable level without being dominated by any other solution [4]. The solution presented here uses NSGA-II algorithm to pick out balanced solutions with respect to all objectives.

NSGA-2 uses an elitist approach, marking the non-dominated solutions with a higher ranking [5]. In the selection process, the evaluated population is divided into 'non-dominated fronts' and each solution in a given front has the same ranking. The fronts with higher ranking are selected first. The remaining solutions are again evaluated with respect to their crowding distance. Crowding distance indicates how far apart the given solution is from other solutions with respect to its score of objective functions. The solutions that are spread out and less crowded are preferred so that the search space is expanded.

3. PROBLEM FORMULATION

3.1 Problem Statement

To perform clustering on a data set containing data about users and their choices, using a heuristic method of evolutionary algorithms, and improving the quality of information retrieval on these clusters by adding information about the context that the objects in the clusters are operating in.

The output of the system will be a set of vectors, each containing a possible configuration of all clusters in the system along with the context information of the cluster.

3.2 Proposed Solution

The aim of this research is to improve the quality of clusters, and thereby also improve the quality of the information system that it is being used in. This is achieved by adding the aspect of context awareness to the clusters. In the solution proposed, the actual process of machine. Learning is performed offline using an evolutionary approach. It is a heuristic approach that provides optimal or near optimal solutions. The reason for picking this approach is the rich solution space that it generates. Multi-objective optimization techniques are used, as the two main objectives to be achieved are context awareness and cluster quality. Integrating multi-objective optimization with evolutionary approach provides a faster way of picking the best solutions from the vast population generated by the evolutionary process.

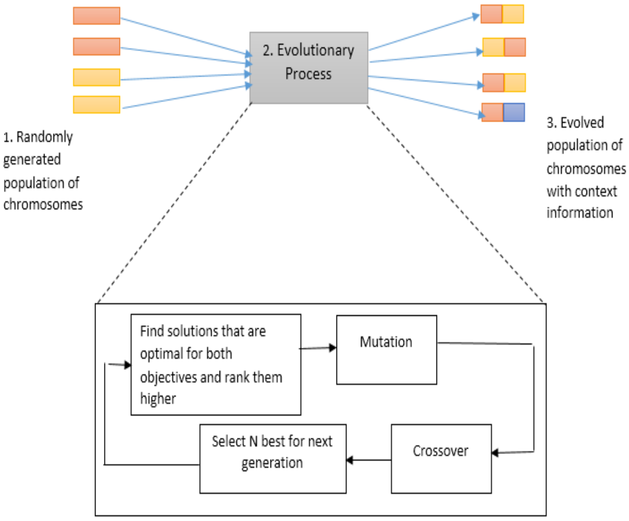
1. Randomly generated population of chromosomes
2. Evolutionary Process
3. Evolved population of chromosomes with context information
Find solutions that are optimal for both objectives and rank them higher
Mutation
Crossover
Select N best for next generation

Fig. 1: Architecture Diagram of Proposed solution

4. MATHEMATICAL MODELLING

The major steps in the evolutionary process for context aware clustering are modelled below.

4.1 Chromosome Representation

Let number of clusters be **m.**

Each chromosome will represent a unique cluster configuration of the total m clusters present in the system. Therefore each chromosome is made up of m components. Each component is made up of 2 parts:

1. The cluster center (K)
2. Context information of the cluster (C)

The cluster center is nothing but one object from the cluster which best represents the cluster properties. Each object is represented by its attributes, and the cluster center is no different. Therefore, let each cluster center K be represented by a vector of its attributes a1, a2, an.

K = [a1, a2... an]   … (1)

Context information is generally made up of categories. Each category may take on a fixed number of values. This can be converted into binary values (present or absent) for ease of further computation.

Eg. Mood happy, sad, mixed => [100]

This conversion is recommended only when the possible values that the context categories can take are limited. Else leaving them in their nominal form is a better option. Each component in the chromosome can now be represented as

Chromosome X = [(K1, C1), (K2, C2) … (Km, Cm)]

i.e. Chromosome X = [(a1,a2...an), (q1,q2...qv), … ,(Km ,Cm)] ...(2)

Where, q1, q2...qv are context attributes

4.2 Population Generation

For generating an initial population that will be evolved,

1. Generate **m** combinations of K and C randomly

2. do this **P** times (P is population size)

4.3 Objective Functions

Since K and C are generated randomly, there are 2 possibilities – the context C may not represent the context information in the cluster, and K may or may not be the best cluster representative. The problem now reduces to picking out, from the set of random solutions, those that satisfy both conditions reasonably well. Thus they lead to two objective functions to be optimized simultaneously. Every chromosome is considered as a possible solution. After each chromosome is evaluated for both objective functions then each solution will have 2 values associated with it – context similarity score and cluster validity score.

4.3.1 Context Similarity

The objects are evaluated to find the extent to which the randomly assigned context of the cluster in question matches with the actual context of the objects in that cluster. The data set required in this computation is generally nominal (categorical). Finding similarity between data objects with nominal attributes is not straightforward. We can follow either of the two approaches –

1. Convert the context data to binary attributes that take only 2 values – 0 or 1. Then use binary similarity measures to compute context similarity

2. Do not convert the context data to binary. Leave it as categorical, and use nominal similarity measures.

Let q1, q2 … qv represent the context categories. Total values that need to be binarized are:

$$\sum_{i=1}^{v} Si * qi \quad \ldots (3)$$

Where Si is the number of possible values that the context category qi can take.

If this value is small, then the first approach can be taken. An example wherein converting to binary is not practical is for the context category of "location". This is an attribute that can take literally thousands of values, especially if the application domain is geographical in nature. Thus, the decision to use either approach depends on the underlying context data.

The following steps are required to be performed to calculate the score for the objective function of context similarity –

1. 1. For every cluster Ki in Chromosome X find most frequently occurring context Fc in that cluster

2. Compare Fc with Ci

3. Get final score for X

To accomplish the first step, the context is represented as a v-dimensional matrix. Each dimension q1,q2...qv represents context attributes and |qv| = no. of values qv can take. So the matrix can be represented as:

ConMatrix [|q1||q2|...|qv|]

Where,

ConMatrix [q1A] [q2B]... [qvZ] = frequency of occurrence of context combination [A, B…Z] ... (5)

Two Main operations to be performed over it are populating the matrix and iterating over it to find the index of the largest value in it. The result of the second step gives Fc, i.e., the most frequently occurring context in Cluster Ki. The complexity of both these operations is

|q1|*|q2|* … * |qv| … (4)

Both these operations are not time efficient as the values of **v** and **|qv|** increase.

Another approach is used which improves the complexity of the 2 operations to be performed on it. In this approach, ConMatrix is represented as a Directory of Keys. This data structure is nothing but a list of unique keys pointing to some values. Each key points to a value. This structure is adapted to be used in this research. The indices of ConMatrix represent a context combination in the database. e.g.

ConMatrix [2, 3, 5] = 4

Indicates that

- there are 3 context attributes
- their current values under consideration are 2,3,5
- the value 4 indicates how many times the context combination [2,3,5] has occurred in the database

It is represented as key value pairs as **{(2, 3, 5), (4)}**.
Populating the matrix can is now carried out as follows:

- for(1 to |Ki|)
- Add context combination to directory as key
- If Key already exists
- Increment corresponding Value by 1
- end for

The worst case complexity of this operation is $\mathbf{|Ki|^2}$.

The second operation to be performed is iterating over the list of keys to find which combination has occurred maximum times. The worst case complexity of this operation is **|Ki|**. Thus the complexity using this approach remains unaffected by the values of **v** and **|qv|** and is only affected by the size of the cluster |Ki|.

The second step involves comparing Fc with Ci (The context that was randomly assigned to Ki during population generation). The higher this score, the more preferred is the solution. Depending on whether that context data is binary or categorical, there are two similarity measures that can be used.

a. Binary similarity measure – Jaccard Distance

The Jaccard Index is used to compute the similarity between assigned cluster context and available object context if the contexts are expressed as binary vectors.

Let Ci = [1,0,0,1,1 …] be the randomly assigned context and Fc = [1,1,0,0,1 …] be frequently occurring both of size v.

Let,
a = number of variables on which values in both vectors are 1
b = number of variables where value in Ci is 1 and Fc is 0
c = number of variables where value in Ci is 0 and Fc is 1

d = number of variables where values in both vectors are 0

a + b + c + d = p, the number of variables.

Jaccard similarity = a / (a + b + c) ... (5)

b. Categorical similarity measure
One of the most commonly used measure is the Overlap Measure. It can be calculated as follows [6].

Sim (Fc, Ci) = 1 if Xk = Yk

= 0 otherwise

Where, Xk and Yk are the values taken by attribute Ak for Ci and Fc respectively.

However, this measure is too simplistic. Alternative measures are available, that use other information available in the data set, like frequency of occurrence of a value to compute similarity. A measure Goodall3 introduced in Boriah et al. (2008) is defined below.

Sim (Fc, Ci) = 1 – $p^2k(Xk)$ if Xk = Yk

= 0 otherwise … (6)

Where $p^2k(Xk) = f(Xk) *(f(Xk) - 1)) / N (N-1)$

And,

f(Xk) is the number of times the attribute Ak takes value Xk in the data set.

The last step involves summing up all individual scores to obtain the final score for Chromosome X.

$$Score(X) = \sum_{i=1}^{m} contextScore(i) \quad \dots (7)$$

4.3.2 *Cluster Validity*

The sum of the squared error (SSE) measures the quality of a clustering, which is also known as scatter. We prefer the one with the smallest SSE since this means that the centers of this clustering are a better representation of the points in their cluster. The SSE is formally defined as follows [7]:

$$SSE = \sum_{i=1}^{m} \sum_{x \in Ki} Dist(Ki, x)^2 \quad \dots (8)$$

Where, **m** is number of clusters,

Ki is the centroid of i^{th} cluster in chromosome X,

x is a data point in Ki,

Dist is Euclidean distance between x and Ki.

4.3.3 *Multi Objective Optimization*

We have 2 objectives that need to be optimized.

The context similarity score must be maximum, and SSE must be minimum for a given chromosome. The concept of non-dominated solutions can be used to find such solutions that satisfy both these constraints in the best possible manner. As mentioned before, every chromosome is considered as a possible solution and each chromosome is evaluated for both objective functions then each solution will have 2 values associated with it –

a) Context Similarity (larger the score the better)

b) SSE

Hence we can plot a graph of the solutions with respect to these values. A Non-Dominated solution is a solution wherein it is not possible to alter it so as to increase the advantage of one objective, without deteriorating, or decreasing the current advantage of another objective. It is a balanced solution.[5]

There is no one solution that will satisfy this condition. It is generally a set of solutions. The aim is to find such a set. We use NSGA-2 algorithm for the same. The core function of such algorithms is the **dominates(p,q)** function.

The objective of context matching has to be maximized and the objective of cluster validity has to be minimized. The job of the dominates(p,q) function is to find whether solution p outperforms q it in both respects. If yes, then p is said to dominate q. If no solution dominates p, p is known as a non-dominated solution. The pseudo-code for the same is given below.

- if Q.Objv1 > P.Objv1 and Q.Obj2 <= P.Obj2
- then P does not dominate Q
- else if Q.Objv1 >= P.Objv1 and Q.Obj2 < P.Obj2
- then P does not dominate Q
- else
- P dominates Q

5. PROPOSED ALGORITHM

Using the modeling described in section above, the following algorithm is proposed as a solution. Each iteration is a generation. The algorithm is assumed to converge after a pre decided number of generations. This value is obtained empirically.

The algorithm is as follows:

Input: Randomly generated initial population P chromosomes

Output: Set of evolved chromosomes containing cluster configurations and context information

```
Step 1: Generate a population of P chromosomes randomly
Step 2: generationNumber = 1\)
Step 3: until generationNumber ! = lastGeneration do
Step 4: Mutate (P, mutationProbability)
Step 5: Crossover (P, crossoverProbability)
Step 6: for each chromosome C
Step 7: compute cluster objects using k-means
Step 8: scoreContext = evaluateContext(C)
Step 9: scoreCluster = evaluateClusters(C)
Step 10: end for
Step 11: FastNonDominatedSort(p)
Step 12: Select N best solutions for next generation
Step 13: Go to 3
```

6. IMPLEMENTATION

The implementation of the above algorithm is done using the Trip Advisor data set [8]. It is a dataset of 4668 rows with the following attributes – User ID, User State, Hotel ID, Hotel State, Trip type, Rating. The first four attributes are considered as object attributes (representing K) and the remaining two are considered as context attributes.

The algorithm is implemented using Java and H2 database engine on Intel(R) Core(TM) i3 CPU.

The values for number of iterations, chromosome size (number of clusters) population size and mutation probability have to be decided empirically. The results presented below use a population size of 100, with each chromosome having 10 clusters each and mutation probability of 20%. The algorithm is run for 30 generations.

7. RESULTS AND DISCUSSIONS

The general trend of the graph is that is the numerical score of context matching score (X-axis) increases, so does the cluster validity score (Y-axis). However, as discussed in previous sections, the larger the score of context matching, the better and lower the score of SSE, the better the cluster quality.

Hence, the results interpreted from the graph indicate that an improvement in the score of context matching results in a deterioration of cluster quality. Also, the deterioration is more than subtle. This conforms to the definition of non -dominated solutions. Thus we can say that the results obtained are a set of non-dominated solutions.

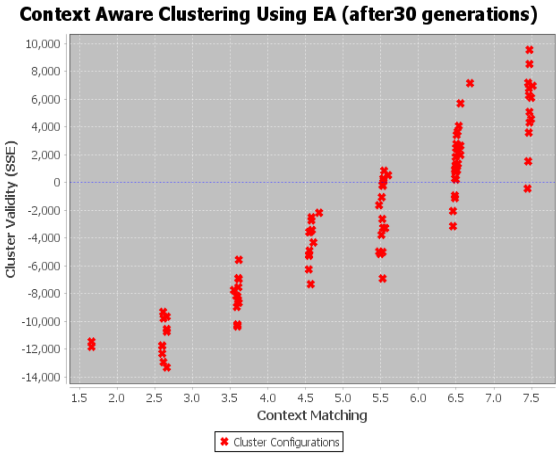

Fig. 2: Results after 30 generations

8. CONCLUSION

Evolutionary approach is a heuristic approach and takes considerable time to finish execution. However it has some notable advantages that make its use worthwhile. Adopting an evolutionary approach to create clusters provides an extremely rich solution space. It finds all possible cluster and context combinations. Pre-computing such combinations of context-cluster configurations will benefit information retrieval in the system that it is applied to. Lastly, The output of this approach also provides a very compact representation of the data set.

9. FUTURE SCOPE

Possible areas for future work could include application of this algorithm to a recommendation system. There have been attempts to build recommendation systems using evolutionary approach [9]. However it would be interesting to observe the results of applying this algorithm to such systems. The output of this algorithm is a set of cluster configurations with context embedded in them. They can be used as an input to a recommendation system to quantify exactly how much pre-computed context aware clusters improve recommendation results.

Another direction is parallelization. Evolutionary algorithms are quite time consuming as the number of generations required for them to converge may be high. However they are inherently parallel. Adapting this algorithm to run in parallel is another direction for future work.

Also, the implementation can be extended to other context aware datasets so that the results can be compared to further analyze the effectiveness and efficiency of the algorithm

10. REFERENCES

[1] Gediminas Adomavicius, Bamshad Mobasher, Francesco Ricci, Alex Tuzhilin, Context Aware Recommender Systems, AI MAGAZINE Fall 2011. ISSN 0738-4602.

[2] A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing, Chapter 2, Springer, Natural Computing Series 1st edition, 2003, ISBN: 3-540-40184-9 Corr. 2nd printing, 2007, ISBN: 978-3-540-40184-1

[3] Anirban Mukhopadhyay, Ujjwal Maulik, Sanghamitra Bandyopadhyay, Carlos A. Coello Coello. Survey of Multiobjective Evolutionary Algorithms for Data Mining: Part II. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 18, NO. 1, FEBRUARY 2014

[4] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan , A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 6, NO. 2, APRIL 2002

[5] Abdullah Konak, David W. Coit, Alice E. Smith. Multi-Objective Optimization Using Genetic Algorithms: A Tutorial.

[6] Boriah, Shyam, Varun Chandola, and Vipin Kumar. "Similarity measures for categorical data: A comparative evaluation." red 30.2 (2008): 3.

[7] Ming-Hseng Tseng, Chang-Yun Chiang, Ping-Hung Tang, Hui-Ching Wu, A STUDY ON CLUSTER VALIDITY USING INTELLIGENT EVOLUTIONARY K-MEANS APPROACH, Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, Qingdao, 11-14 July 2010.

[8] Marcheggiani, D., Täckström, O., Esuli, A, Sebastiani, F.: Hierarchical Multi-Label Conditional Random Fields for Aspect-Oriented Opinion Mining. In: Proceedings of the 36th European Conference on Information Retrieval (ECIR 2014).

[9] Yi Zuo, Maoguo Gong, Jiulin Zeng, Lijia Ma, and Licheng Jiao. Personalized Recommendation Based on Evolutionary Multi-Objective Optimization. IEEE Computational intelligence magazine | February 2015