Enhancing the Effectiveness of clustering using User Preferences and Voting/Rating

Y. Subba Reddy, Dr. V. Tanuja and Prof. P. Govindarajulu

Department of Computer Science, S.V. University, Tirupati, AP, INDIA

Abstract
Finding the similarity between two objects is the most important fundamental operation in database management as well as in web searching environment. The similarity between two objects is generally computed based on the attribute values of the objects. Traditional similarity measures using only attribute values. In the proposed method, similarity between two objects is the most effective and accurate when the similarity is computed based on the attribute values as well as the voting/rating/preferences of values of attributes. That is, the similarity between objects is not based only on attribute values but instead object similarity is computed based on some of the weighted values of voting/rating/preferences and values of attributes. A linear similarity function is the simplest model for finding weighted similarities between objects. Similarity measure techniques are very much useful in processing database queries such as top-k queries, reverse top-k queries, k-nearest neighbor queries and other different types of queries related to business sales activities. Sometimes there is a need to construct and use multidimensional indexing data structures for efficient search/access of data of very large database sizes and the effective execution of queries. Also sometimes different types of pruning techniques are required for scalability purpose. In general, linear function computations are scalable for similarity finding measurements between objects.

Key words:
Similarity between objects, similarity search computations, top-k queries, reverse top-k queries, and other k-nearest neighbor queries, database queries, database operations, attribute values, customer voting/rating/preferences.

1. Introduction

World Wide Web (WWW) is an information super highway on the Internet and the web is considered as a front-end tool used for searching/accessing data from the different types of databases in the form posed queries. Customers have opportunities to select and purchase desired products through Internet voting/rating of products. Many websites on the Internet have provided means and ways for collecting opinions or ratings or preferences of products through Internet voting/rating. Many E-commerce websites allow their customers to vote or rate business products. Customer voting or preferences or opinions or ratings play an important role in promoting product business sales. Internet voting/rating is the latest trend of many business websites in product sale improvement operations. Many recommender machine learning based software systems are available in the market for product recommendations.

The Internet is a powerful tool for collecting customer based voting/rating/opinions/preferences for products (objects). Recommender software systems are based on new machine learning data analysis techniques used for customers in purchasing products based on the voting/rating/opinions/preferences used in the product ranking order. It is a well-known fact that customers' voting/rating/opinions/preferences directly reflect their mindset, view, or sentiment, knowledge, and intelligence with respect to purchasing trend of objects such as products, services, resources, and any other useful things. Hence, mining through voting/rating/opinions/preferences is the current hot topic in intelligent machine learning based data mining research. Customers provide their views, interests, opinions, preferences, feelings, sentiments and other useful plans and ideas in the form of Internet voting/rating through the available social media websites such as Facebook, Twitter, Google+ and Blogs etc. Similarity search between data objects (products) is the most important and fundamental operation for smooth and effective data management in many modern database applications. Similarity search is used for finding similar conversations likings, and disliking in the social networks and social media sites (e.g., Flickr, YouTube, and Facebook) are a popular distribution outlet for users looking to share their experiences and interests on the Web [2]. Many web applications based on the similarity between objects are rapidly growing. Similarity measurement techniques are useful not only in the data management but also in the many social networks such as Twitter, Facebook, and Google+ etc. similarity search based computations are needed in speech recognition, research areas, weather applications, business applications

Manuscript received September 5, 2017
Manuscript revised September 20, 2017

and many other database applications and so on. Although the importance of user-centric evaluations has become quite clear and vital, the majority of recommender system studies still solely report the traditional, data-centric evaluation results. Recommendation systems are deeply dependent on the user-item rating matrix, which is usually very sparse and the recommendation context, the users are represented as multidimensional vectors, where each item (product) represents a dimension [7]. Users often need to optimize the selection of objects by appropriately weighting the importance of multiple object attributes [8]. Recommender systems provide a user with the content she or he might be interested in, and they have become increasingly popular because of their successful applications in the E-commerce field, such as with Amazon and eBay and traditionally, recommender systems have been evaluated according to accuracy metrics in the Information Retrieval area [4]. Different types of similarity search based measuring techniques have been proposed in the literature for finding similarity between two objects (products). The objects may be database objects, documents, web pages, images, computer graphics objects, comments, data entities, conversations, words, speech, groups of bits, sequences, and strings and so on. The objects may be either logical or physical. The similarity between objects is used to detect abnormal behaviors based on the products customer buy [6]. Finding the similarity between objects is the most important and a fundamental operation ineffective and efficient database management. For example, search engine searches pages or documents that contain similar words all over the World Wide Web (WWW). Similarity search is needed to find customers having abnormal features (outliers) based on the products they buy. Also, similarity search based calculations are required to find similar conversations and comments between the users of the social network environments such as the Facebook and Twitter. The user views and profiles can be analyzed to recommend better products and services to users. Objects are represented by a set of attributes and each attribute will be given a specific value. Traditional similarity metrics are used to find the similarity between two objects in terms of their attribute values. The similarity value is quantified and the similarity will increase as the quantified value approaches to zero. Many similarity metrics have been proposed for evaluating the similarity between two data items, such as the Euclidean distance and the cosine similarity [1]. Euclidean distance and the cosine similarity are two of the most popular and useful similarity metrics that are frequently used for finding similarity measure quantitatively between two different data objects or data records or data sequences or data trajectories and so on. In the literature, many varieties of similarity metrics have been proposed for estimating similarity between two objects. Here objects may be data items or records or tables or relations or objects or files or diagrams or pages or images or documents and so on. Traditional similarity search computations are based solely on their attribute values but not on different opinions of customers on attribute values.

Mathematically business objects or entities or products are represented as points and these points are defined based on the values of the respective attributes of the objects. Newly proposed object similarity-based search method takes care of both similarity metric computation values and customers’ voting/rating/preferences or opinions of users. For example, State Bank of India observes and analyzes operations, preferences, plans, interest rates and other additional details for taking efficient and effective decisions for the smooth functioning of all the operations of SBI. Top management of SBI must establish good rules and policies by creating classification or clustering of tasks, operations, and many other useful operations. Similarly, in the case of product sales, the business manager must execute a query for obtaining product sales details with respect to both customer opinions and characteristics of products. Product characteristics are specified by using a set of attributes. Konstantinos Georgoulas et al [1] introduced a novel framework for user-centric similarity search, which capitalizes on rankings of products based on user preferences to discover similar products. In this paper, authors have introduced a user-centric approach for similarity computation, in which the similarity of two products is defined by taking into account the customer voting/rating/preferences [1]. Similarity search is used for finding pages or documents with similar words over the World Wide Web (WWW) [2]. The similarity between objects is used to detect abnormal behaviors based on the products customer buy [3]. Similarity search is used for finding similar conversations likings and disliking in the social networks [4]. The similarity between two objects is performed using voting/rating/preferences and item-based collaborative filtering techniques may share a similar intuition, but in contrary to our methods, they suggest that customers have a taste of some products and thus rate them [5].

Haydar et al [6] presented a new clustering algorithm based on density and mutual votes. There are many recommendation system technologies which can be divided into four categories content-based (CB), collaborative filtering (CF), network-based (NB), and hybrid recommendation (HR) [7]. With the rapid advances in social networks, services like Facebook, Twitter, and Google+ have provided us revolutionary ways in which of making friends. Recently several recommendation systems have been proposed, that is based on collaborative filtering, content-based filtering, and hybrid recommendation technique. Although the importance of user-centric evaluations has become quite clear and vital, the majority of recommender system studies still solely report the traditional, data-centric evaluation results. Recommender systems provide a user with the content she or he might be interested in and they have become increasingly popular because of their successful applications in the E-commerce field, such as with Amazon and eBay. Reverse nearest neighbor aggregates are of natural interest in decision support systems for applications that compute proximity, based on geographical distance or vector-space similarity, between “servers” and “clients” [9]. Given a set of data points P and a query point q in a multidimensional space, RNN query finds every data point in P with q as its nearest neighbor (NN) [10]. Item-based collaborative filtering techniques may share a similar intuition, but in contrary to our methods, they suggest that customers have a taste of some products and thus rate them [11]. Recently several recommendation systems have been proposed, that is based on collaborative filtering, content-based filtering, and hybrid recommendation technique. Web mining plays very important role for finding the frequent data pattern from Internet, data set, data mart etc and World Wide Web has become a powerful platform to store and retrieve information as well as mine useful knowledge and use that knowledge to predict the interest of people. Web recommendation systems help the website visitors for easy navigation of web pages, quickly reaching their destination and to obtain relevant information [12]. The user’s data files can be constructed by using responses to questions, item ratings, or the user’s navigation information to infer the user’s preferences and interests. Collaborative filtering system collects all information about user’s interest on the website from the web servers/database and calculates the similarity among the user’s interest [12]. Similarity search is used for finding pages or documents with similar words over the World Wide Web (WWW) [13]. Efficient processing of NN queries requires spatial data structures which capitalize on the proximity of the objects to focus the search of potential neighbors only [14]. Recommender systems apply knowledge discovery techniques to the problem of making customized recommendations for information, products or services throughtout a live interaction [15]. Reverse Nearest Neighbor (RNN) queries are of particular interest in a wide range of applications such as decision support systems, problem based marketing, data streaming, document databases, and bioinformatics [16]. The goal of the Reverse the Nearest Neighbor (RNN) problem is to end all points in a given data set whose nearest neighbor is a given query point [16]. The goal of such recommender systems is to assist its users in finding their preferred items from the large set of items [17]. There are many recommendation system technologies which can be divided into four categories content-based (CB), collaborative filtering (CF), network-based (NB), and hybrid recommendation (HR) [18]. Top-k queries retrieve only the k objects that best match the user preference, thus avoiding huge and overwhelming result sets and therefore it is very important for a manufacturer that its products are returned in the highest ranked positions for as many different user preferences as possible [19]. With the rapid advances in social networks, services such as Facebook, Twitter, and Google+ have provided us revolutionary ways of making friends [20]. First, it is necessary to understand customer interests and preferences and then provide suitable products or services at an adequate time [21]. Through customers’ purchasing histories, the product relevance, such as brand, material, size, color, appearance, price, quality, etc., can be studied to understand customers’ preferences toward particular product features [21].

2. Related Work

Achtert Elke, et al. [1] proposed an index structure called RNN-Tree for reverse 1-nearest neighbor search and also proposed the RdNN-Tree that extends the RNN-Tree by combining the two index structures (NN-Tree and RNN-Tree) into one common index. Fagin Ronald [3] gives an overview of some recent algorithms on aggregating information from various sources, in order to obtain the overall top k objects. Haydar Charif et al. [7] proposed a density-based clustering algorithm called mutual vote (MV) that uses a statistical model to adapt itself to each vector’s perception of its neighborhood, and aggregate the perception of neighboring vectors and also they have presented a new clustering algorithm based on density and mutual votes. Konstantinos Georgoulas et al. [5] introduced a novel framework for user-centric similarity search, which capitalizes on rankings of products based on user preferences to discover similar products and paper authors have introduced a user-centric approach for similarity computation, in which the similarity of two products is defined by taking into account the customer voting/rating/preferences [5]. Hristidis Vagelis et al. [8] have implemented the algorithms in a prototype system called PREFER, which operates on top of a commercial database management system. Olfa Nasraoui et al. [11] proposed Fuzzy approximation reasoning method on intelligent web recommendation system. They have extracted the user profile using used web usage. Akrivi Vlachou et al. [19] proposed to reverse top-k queries and study two different versions: monochromatic and bichromatic reverse top-k queries.

3. Problem Definition

Traditional similarity measuring methods used for finding similarity between two objects are completely based on the values of attributes of tuples but not on the voting/rating/preferences of values of attributes of objects. Since a higher percentage of product or service selections and voting for policies or people are taking place on the World Wide Web (WWW) using social networks and E-commerce websites, it is needed to cope with such data trends and usage in terms of understanding such big data and use the analysis from that data for decision-making. Traditional recommendation systems need to be empowered with modern techniques of machine learning. Similarity among products or people or contents is usually measured using some similarity finding techniques such as Euclidean distance, cosine similarity, city block distance and so on. The results produced by these traditional similarity methods may not be accurate and even these results are not applicable or incapable to use in many real-time applications, where the data size is large and the data organization is sparse. To deal with limitations the present work proposes a new product clustering algorithm based on Jaccard coefficient similarity measure that uses not only values of attributes but also voting/rating/preferences of attributes given by the customers

.4. Methodology

A set of products are denoted by the set P and each product is represented by a set of d attributes where d is the dimensionality of the products set. Customer set is denoted by C and customer opinions are also represented as another set. Customer preferences or opinions are represented for all the products and by a single customer in one single table. Hence, ‘n’ number tables are needed to represent all voting/rating/preferences or opinions of all ‘n’ number of customers. Each table stores preferences of all products. In general; similarity of products is computed based on the values of the attributes of the products and customer voting/rating/preferences or opinions. Many of the existing clustering algorithms are based on only attribute values. The present method considers values of attributes as well as user voting/rating/preferences. A linear weighted score function is used to find similarity between products and then products are clustered based on these computed similarity values. Sometimes it may be necessary to cluster customers based on the computed linear mathematical score function values. The linear score function may be either maximization or minimization depending on the criteria of the values of the attributes of the products. Weighted values are computed for each product separately. Many preference values are specified for each product by many customers. Each product has a separate voting/rating/preferences or opinion list of values by a separate customer. The present paper proposes a method for product clustering based on the customer voting/rating/preferences or opinions. The Linear score function is the best function for computing weighted scores of products.

Linear weighted function (LWF) = f(P)

$$f(P) = \sum_{i=1}^{n} atribute\ value\ of\ P_i * opinion\ value\ of\ P_i$$

Where Pi is the ith attribute of the product P. Recommendation systems that try to suggest items (e.g., music, movies, and books) to users have become more and more popular in recent years [8].

5. Queries

Top-k Query:
A top-k query is executed based on the result of a linear function score value. A top-k query is defined by assigning a suitable voting/rating/performance weight for each value of each attribute of the tuple. With respect to one tuple, values of all attributes are assigned with normalized voting/rating/performance weight values with respect to customer voting/rating/preference weight values. A Linear scoring function is used to compute weighted sums of products of values of attributes and their voting/rating/performance weight values. The top-k query produces a result of a ranked list of the k number of products with best scores. The voting/rating/preference weight values of the products directly influence ranking order of top-k query result. In general, large product data set is indexed by a multidimensional indexing tree called R-tree and the top-k query is usually processed by using a state-of-the-art branch-and-bound algorithm.

Reverse Top-k Query:
A reverse top-k query is defined for a given or selected product, p, in such a way that it returns preferred weights for which product p is in the top-k result set. The performance of the reverse top-k query depends mainly on the number of evaluated top-k queries. The threshold pruning based reverse top-k query significantly reduces the number of top-k query executions. Present work uses a reverse top-k query for clustering products.

6. Algorithm

Newly proposed product clustering algorithm uses Jaccard coefficient similarity measure for clustering products

```
1. Read product data set of 'n' tuples into the suitable data structure.
2. Read voting/rating/preferences details of 'm' number of customers for all the products
3. Compute top-k query results for all the products for all the 'm' number of customer voting/rating/preferences
4. Compute reverse top-k queries for all the products obtained in the step-4.
5. s = get Reverse Set Products Count.
6. threshold = get Threshold Value
7.  minimumCount = s * threshold
8. While(s > minimumCount) do
  {
9.  StartCluster = first cluster of the present list of products
10. For cluster i = 2 to last in the current list compute similarity measure, Sim(startCluster, i) and store

11. Combine all the groups whose similarity measure value > than the specified threshold value into one cluster.

12. presentCount = number of groups combined in the
    step12.
13. s = s - presentCount
  }
```

Algorithm Explanation

Example product data set with cardinally value 'n' is stored efficiently in the memory. Each customer specifies voting/rating/performance details of products. Traditional similarity finding methods use only values of attributes for finding similarity measures between products (objects). The present method uses both values of attributes and their respective voting/rating/performance details for computing similarity between two products. A linear scoring function is used for finding similarity measure between two products. This linear function uses both values of attributes and the corresponding voting/rating/performance values. A Top–k query returns a top-k number of the best products based on the linear function score values. Reverse top–k query returns all customers who have included top-k products in their favorite lists. Assume that the value of the variable, s, be the total number of products included in the voting/rating/performance lists of customers. The control structure while loop executes repeatedly and in each iteration, it groups products into clusters. The variable present count represents a total number of products clustered in the current iteration. At the end of a current iteration, a total number of products to be clustered are updated. Note that while loop is repeated until s &gt; the value of minimum count.

7. Example on Sample Data Set

For simplicity usage and easy explanation purpose a synthetic product data set with twelve product details shown in TABLE-1 is considered. Again for simplicity purpose only three attributes are taken. One simple and easy way for getting values of price and maintenance attributes is to generate numeric values using randomization process in the case of synthetic data sets. Also a set of details of voting/rating/performances of ten customers for the selected set products are shown in the TABLE-2. In reality the sizes of products and customers may be very large.

Table 1: Products

Product	Price	Maintenance
P1	60	30
P2	90	70
P3	50	60
P4	35	45
P5	61	29
P6	89	71
P7	62	28
P8	91	69
P9	49	61
P10	60	29
P11	34	46
P12	36	44

Table 2: Customer Preferences

Customer	Price Preference	Maintenance Preference
C1	0.25	0.75
C2	0.60	0.40
C3	0.50	0.50
C4	0.80	0.20
C5	0.40	0.60
C6	0.50	0.50
C7	0.60	0.40
C8	0.40	0.60
C9	0.20	0.80
C10	0.30	0.70

Table 3: Customer-1 Preference list of top-5 Products
= {P1 P5 P7 P10 P12}

Product	Price Preference + Maintenance Preference scores	Total Score
P1	60*0.25 + 30*0.75 = 15.0+22.5	**37.50**
P2	90*0.25 + 70*0.75 = 22.5+52.5	75.00
P3	50*0.25 + 60*0.75 = 12.5+45.0	57.50
P4	35*0.25 + 45*0.75= 8.75+33.75	42.50
P5	61*0.25+29*0.75=15.25+21.75	**37.00**
P6	89*0.25+ 71*0.75=22.25+53.25	75.50
P7	62*0.25 + 28*0.75 = 15.5+21.0	**36.50**
P8	91*0.25+ 69*0.75=22.75+51.75	74.50
P9	49*0.25+ 61*0.75=12.25+45.75	58.00
P10	60*0.25 + 29*0.75=15.0+21.75	**36.75**
P11	34*0.25 + 46*0.75 = 8.5+34.5	43.00
P12	36*0.25 + 44*0.75= 9.0 + 33.0	**42.00**

Table 4: Customer-2 Preference list of top-5 Products
= {P1 P4 P10 P11 P12}

Product	Price Preference + Maintenance Preference scores	Total Score
P1	60*0.60 + 30*0.40 = 36.0+12.0	**48.00**
P2	90*0.60 + 70*0.40 = 54.0+28.0	82.00
P3	50*0.60 + 60*0.40 = 30.0+24.0	54.00
P4	35*0.60 + 45*0.40= 21.0+18.00	**39.00**
P5	61*0.60+29*0.40=36.60+11.60	48.20
P6	89*0.60+ 71*0.40=53.40+28.40	81.80
P7	62*0.60+28*0.40=37.20+11.20	48.40
P8	91*0.60+ 69*0.40=54.60+27.60	82.20
P9	49*0.60+ 61*0.40=29.4+24.4	53.80
P10	60*0.60 + 29*0.40=36.0+11.6	**47.60**
P11	34*0.60 + 46*0.40 = 20.4+18.4	**38.80**
P12	36*0.60 + 44*0.40= 21.6 + 17.6	**39.20**

Table 5: Customer-3 Preference list of top-5 Products
= {P4 P7 P10 P11 P12}

Product	Price Preference + Maintenance Preference scores	Total Score
P1	60*0.50 + 30*0.50 = 30.0+15.0	45.00
P2	90*0.50 + 70*0.50 = 45.0+35.0	80.00
P3	50*0.50 + 60*0.50 = 25.0+30.0	55.00
P4	35*0.50+45*0.50= 17.50+22.50	**40.00**
P5	61*0.50+29*0.50=30.50+14.50	45.00
P6	89*0.50+ 71*0.50=44.50+35.50	80.00
P7	62*0.50 + 28*0.50 = 31.0+14.0	**45.00**
P8	91*0.50+ 69*0.50=45.50+34.50	80.00
P9	49*0.50+ 61*0.50=24.50+30.50	55.00
P10	60*0.50 + 29*0.50=30.0+14.50	**44.50**
P11	34*0.50 + 46*0.50 = 17.0+23.0	**40.00**
P12	36*0.50 + 44*0.50= 18.0 + 22.0	**40.00**

Table6: Customer-4 Preference list of top-5 Products
= {P3 P4 P9 P11 P12}

Product	Price Preference + Maintenance Preference scores	Total Score
P1	60*0.80 + 30*0.20 = 48.0+6.0	54.00
P2	90*0.80 + 70*0.20 = 72.0+14.0	86.00
P3	50*0.80 + 60*0.20 = 40.0+12.0	**52.00**
P4	35*0.80+45*0.20= 28.0+9.00	**37.00**
P5	61*0.80+29*0.20=48.80+5.80	54.60
P6	89*0.80+ 71*0.20=71.20+14.20	85.40
P7	62*0.80 +28*0.20= 49.60+5.60	55.20
P8	91*0.80+ 69*0.20=72.80+13.80	86.60
P9	49*0.80+ 61*0.20=39.20+12.20	**51.40**
P10	60*0.80+ 29*0.20=48.0+5.40	53.40
P11	34*0.80+46*0.20= 27.20+9.20	**36.40**
P12	36*0.80+44*0.20= 28.80 + 8.80	**37.60**

Table 7: Customer-5 Preference list of top-5 Products
= {P4 P7 P10 P11 P12}

Product	Price Preference + Maintenance Preference scores	Total Score
P1	60*0.40 + 30*0.60 = 24.0+18.0	42.00
P2	90*0.40 + 70*0.60 = 36.0+42.0	78.00
P3	50*0.40 + 60*0.60 = 20.0+36.0	56.00
P4	35*0.40+45*0.60= 14.4+27.0	**41.00**
P5	61*0.40+29*0.60=24.4 +17.40	41.80
P6	89*0.40+ 71*0.60=35.60+42.60	78.20
P7	62*0.40 +28*0.60= 24.8+16.8	**41.60**
P8	91*0.40+ 69*0.60=36.4+41.4	77.80
P9	49*0.40+ 61*0.60=19.6+36.6	56.20
P10	60*0.40+ 29*0.60=24.0+17.4	**41.40**
P11	34*0.40+46*0.60= 13.6+27.6	**41.20**
P12	36*0.40+44*0.60= 14.40 + 26.4	**40.80**

Table 8: Customer-6 Preference list of top-5 Products
= {P4 P7 P10 P11 P12}

Product	Price Preference + Maintenance Preference scores	Total Score
P1	60*0.50 + 30*0.50 = 30.0+15.0	45.00
P2	90*0.50 + 70*0.50 = 45.0+35.0	80.00
P3	50*0.50 + 60*0.50 = 25.0+30.0	55.00
P4	35*0.50+45*0.50= 17.50+22.50	**40.00**
P5	61*0.50+29*0.50=30.50+14.50	45.00
P6	89*0.50+ 71*0.50=44.50+35.50	80.00
P7	62*0.50 + 28*0.50 = 31.0+14.0	**45.00**
P8	91*0.50+ 69*0.50=45.50+34.50	80.00
P9	49*0.50+ 61*0.50=24.50+30.50	55.00
P10	60*0.50 + 29*0.50=30.0+14.50	**44.50**
P11	34*0.50 + 46*0.50 = 17.0+23.0	**40.00**
P12	36*0.50 + 44*0.50= 18.0 + 22.0	**40.00**

Table 9: Customer-7 Preference list of top-5 Products
= {P1 P4 P10 P11 P12}

Product	Price Preference + Maintenance Preference scores	Total Score
P1	60*0.60 + 30*0.40 = 36.0+12.0	**48.00**
P2	90*0.60 + 70*0.40 = 54.0+28.0	82.00
P3	50*0.60 + 60*0.40 = 30.0+24.0	54.00
P4	35*0.60 + 45*0.40= 21.0+18.00	**39.00**
P5	61*0.60+29*0.40=36.60+11.60	48.20
P6	89*0.60+ 71*0.40=53.40+28.40	81.80
P7	62*0.60+28*0.40=37.20+11.20	48.40
P8	91*0.60+ 69*0.40=54.60+27.60	82.20
P9	49*0.60+ 61*0.40=29.4+24.4	53.80
P10	60*0.60 + 29*0.40=36.0+11.6	**47.60**
P11	34*0.60 + 46*0.40 = 20.4+18.4	**38.80**
P12	36*0.60 + 44*0.40= 21.6 + 17.6	**39.20**

Table 10: Customer-8 Preference list of top-5 Products
= {P4 P7 P10 P11 P12}

Product	Price Preference + Maintenance Preference scores	Total Score
P1	60*0.40 + 30*0.60 = 24.0+18.0	42.00
P2	90*0.40 + 70*0.60 = 36.0+42.0	78.00
P3	50*0.40 + 60*0.60 = 20.0+36.0	56.00
P4	35*0.40+45*0.60= 14.4+27.0	**41.00**
P5	61*0.40+29*0.60=24.4 +17.40	41.80
P6	89*0.40+ 71*0.60=35.60+42.60	78.20
P7	62*0.40 +28*0.60= 24.8+16.8	**41.60**
P8	91*0.40+ 69*0.60=36.4+41.4	77.80
P9	49*0.40+ 61*0.60=19.6+36.6	56.20
P10	60*0.40+ 29*0.60=24.0+17.4	**41.40**
P11	34*0.40+46*0.60= 13.6+27.6	**41.20**
P12	36*0.40+44*0.60= 14.40 + 26.4	**40.80**

Table 11: Customer-9 Preference list of top-5 Products
= {P1 P5 P7 P10 P12}

Product	Price Preference + Maintenance Preference scores	Total Score
P1	60*0.20 + 30*0.80 = 12.0+24.0	36.00
P2	90*0.20 + 70*0.80 = 18.0+56.0	74.00
P3	50*0.20 + 60*0.80 = 10.0+48.0	58.00
P4	35*0.20+45*0.80= 7.0+36.0	**43.00**

P5	61*0.20+29*0.80=12.20 +23.20	35.40
P6	89*0.20+ 71*0.80=17.80+56.80	74.20
P7	62*0.20+28*0.80= 12.40+22.40	**34.80**
P8	91*0.20+ 69*0.80=18.20+55.20	73.40
P9	49*0.20+ 61*0.80=9.80+48.80	58.60
P10	60*0.20+ 29*0.80=12.0+23.20	**35.20**
P11	34*0.20+46*0.80= 6.8+36.8	**43.60**
P12	36*0.20+44*0.80= 7.2 + 35.2	**42.40**

Table 12: Customer-10 Preference list of top-5 Products = {P1 P5 P7 P10 P12}

Product	Price Preference + Maintenance Preference scores	Total Score
P1	60*0.30 + 30*0.70 = 18.0+21.0	**39.00**
P2	90*0.30 + 70*0.70 = 27.0+49.0	76.00
P3	50*0.30 + 60*0.70 = 15.0+42.0	57.00
P4	35*0.30+45*0.70= 10.50+31.50	42.00
P5	61*0.30+29*0.70=18.30 +20.30	**38.60**
P6	89*0.30+ 71*0.70=26.70+49.70	76.40
P7	62*0.30+28*0.70= 18.60+19.60	**38.20**
P8	91*0.30+ 69*0.70=27.30+48.30	75.60
P9	49*0.30+ 61*0.70=14.70+42.7	57.40
P10	60*0.30+ 29*0.70=18.0+20.30	**38.30**
P11	34*0.30+46*0.70= 10.20+32.20	42.40
P12	36*0.30+44*0.70= 10.80 + 30.8	**41.60**

8. Algorithm Description

Algorithm executes in iterative manner. In the first iterations all the products are compared and then all the products that satisfy threshold value are grouped into one cluster. In the second iteration remaining un-grouped products are compared and all the products that satisfy threshold value are grouped. The process continues until a group of clusters are formed. The algorithm terminates when the remaining un-clustered tuples number falls a specified count.

Table 13: clusters of products using customer opinions

Customer	Top-5 products
C1	P1 P5 P7 P10 P12
C2	P1 P4 P10 P11 P12
C3	P1 P4 P10 P11 P12
C4	P3 P4 P9 P11 P12
C5	P4 P7 P10 P11 P12
C6	P1 P4 P10 P11 P12
C7	P1 P4 P10 P11 P12
C8	P4 P7 P10 P11 P12
C9	P1 P5 P7 P10 P12
C10	P1 P5 P7 P10 P12

Table 14: clusters of customers with respect to products

Product	Favorite list of Customers
P1	C1 C2 C3 C6 C7 C9 C10
P2	------------ NIL ----------
P3	C4
P4	C2 C3 C4 C5 C6 C7 C8
P5	C1 C9 10
P6	------------ NIL ----------
P7	C1 C5 C8 C9 C10
P8	------------ NIL ----------
P9	C4
P10	C1 C2 C3 C5 C6 C7 C8 C9 C10
P11	C2 C3 C4 C5 C6 C7 C8
P12	C1 C2 C3 C4 C5 C6 C7 C8 C9 C10

First top-k best products are determined from the product set with respect to customer voting/rating/preferences and then using reverse top-k method, favorite product lists of customers are determined. After determining favorite lists Jaccard coefficient similarity measure is used for clustering products. Jaccard coefficient computation details are explained by taking hypothetical data set of products.

$$Sim(P_1,P_2) = \frac{P_1 \cap P_2}{P_1 \cup P_2} = 0$$

$$Sim(P_1,P_4) = \frac{P_1 \cap P_4}{P_1 \cup P_4} = \frac{|\{C_2, C_3, C_6, C_7\}|}{|\{C_1, C_2, C_3, C_6, C_7, C_9, C_{10}, C_4, C_5, C_8\}|} = \frac{4}{10} = 0.4$$

$$Sim(P_1,P_5) = \frac{P_1 \cap P_5}{P_1 \cup P_5} = \frac{|\{C_1, C_9, C_{10}\}|}{|\{C_1, C_2, C_3, C_6, C_7, C_9, C_{10}\}|} = \frac{3}{7} = 0.43$$

$$Sim(P_1,P_7) = \frac{P_1 \cap P_7}{P_1 \cup P_7} = \frac{|\{C_1, C_9, C_{10}\}|}{|\{C_1, C_2, C_3, C_6, C_7, C_9, C_{10}, C_5, C_8\}|} = \frac{3}{9} = 0.33$$

$$Sim(P_1,P_9) = \frac{P_1 \cap P_9}{P_1 \cup P_9} = 0$$

$$Sim(P_1,P_{10}) = \frac{P_1 \cap P_{10}}{P_1 \cup P_{10}} = \frac{|\{C_1, C_2, C_3, C_6, C_7, C_9, C_{10}\}|}{|\{C_1, C_2, C_3, C_6, C_7, C_9, C_{10}, C_5, C_8\}|} = \frac{7}{9} = 0.4$$

$$Sim(P_1,P_{11}) = \frac{P_1 \cap P_{11}}{P_1 \cup P_{11}} = \frac{|\{C_2, C_3, C_6, C_7\}|}{|\{C_1, C_2, C_3, C_6, C_7, C_9, C_{10}, C_4, C_5, C_8\}|} = \frac{4}{10} = 0.4$$

$$Sim(P_1,P_{12}) = \frac{P_1 \cap P_{12}}{P_1 \cup P_{12}} = \frac{|\{C_1, C_2, C_3, C_6, C_7, C_9, C_{10}\}|}{|\{C_1, C_2, C_3, C_6, C_7, C_9, C_{10}, C_4, C_5, C_8\}|} = \frac{7}{10} = 0.7$$

Assumed threshold value is 0.5. Products P1, P10, and P12 are clustered in the first iteration. Total number of similarity comparisons = 8.

Table 15: Customer grouping with respect to products

Product	Favorite list of Customers
P3	C4
P4	C2 C3 C4 C5 C6 C7 C8
P5	C1 C9 10
P7	C1 C5 C8 C9 C10
P9	C4
P11	C2 C3 C4 C5 C6 C7 C8

$$Sim(P_3,P_4) = \frac{P_3 \cap P_4}{P_3 \cup P_4} = 0$$

$$Sim(P_3,P_5) = \frac{P_3 \cap P_5}{P_3 \cup P_5} = 0$$

$$Sim(P_3,P_7) = \frac{P_3 \cap P_7}{P_3 \cup P_7} = 0$$

$$Sim(P_3,P_9) = \frac{P_3 \cap P_9}{P_3 \cup P_9} = \frac{|\{C_4\}|}{|\{C_4\}|} = \frac{1}{1} = 1.0$$

$$Sim(P_3,P_{11}) = \frac{P_3 \cap P_{11}}{P_3 \cup P_{11}} = \frac{1}{7} = 0.14$$

Assumed threshold value is 0.5. Products P3, and P9 are clustered in the second iteration. Total number of similarity comparisons = 5

Table 16: Customer grouping with respect to products

Product	Favorite list of Customers
P4	C2 C3 C4 C5 C6 C7 C8
P5	C1 C9 10
P7	C1 C5 C8 C9 C10
P11	C2 C3 C4 C5 C6 C7 C8

$$Sim(P_4,P_5) = \frac{P_4 \cap P_5}{P_4 \cup P_5} = 0$$

$$Sim(P_4,P_7) = \frac{P_1 \cap P_{12}}{P_1 \cup P_{12}} = \frac{|\{C_5, C_8\}|}{|\{C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_1, C_9, C_{10}\}|} = \frac{2}{10} = 0.2$$

$$Sim(P_4,P_{11}) = \frac{P_4 \cap P_{11}}{P_4 \cup P_{11}} = \frac{|\{C_2, C_3, C_4, C_5, C_6, C_7, C_8\}|}{|\{C_2, C_3, C_4, C_5, C_6, C_7, C_8\}|} = \frac{1}{1} = 1.0$$

Assumed threshold value is 0.5. Products P4, and P11 are clustered in the third iteration. Total number of similarity comparisons = 3

Table 17: Customer grouping with respect to products

Product	Favorite list of Customers
P5	C1 C9 10
P7	C1 C5 C8 C9 C10

$$Sim(P_5,P_7) = \frac{P_5 \cap P_7}{P_5 \cup P_7} = \frac{|\{C_1, C_9, C_{10}\}|}{|\{C_1, C_5, C_8, C_9, C_{10}\}|} = \frac{3}{5} = 0.6$$

Assumed threshold value is 0.5. Products P5, and P7 are clustered in the fourth iteration. Total number of similarity comparisons = 1

9. Results

When normal simple Euclidian distance measure is used clustered products are shown in TABLE

Table 18: Final Euclidian Clusters

Cluster No	Clustered Products
1	P1 P5 P7 P10
2	P2 P6 P8
3	P3 P9
4	P4 P11 P12

With preference based proposed similarity measure, all the given products are clustered and final clusters are shown in Table-19

Table 19: Final preference similarity based clusters

Cluster No	Clustered Products
1	P1 P10 P12
2	P3 P9
3	P4 P11
4	P5 P7
*5	P2 P6 P8

In the above table cluster no 5*, is created based on the lowest preferences given or no preferences not yet given by the users. The formation of this cluster needs no more computation. The new approach creates this cluster by eliminating least preference products or new products.

Table20: Comparisons for Euclidian distance based clustering

Iteration No	Number of comparisons
1	11
2	7
3	4
4	2

Table 21: Comparisons for voting based clustering

Iteration No	Number of comparisons
1	8
2	5
3	2
4	1

The comparative Table 22

Iteration No	Euclidian	voting
1	11	8
2	7	5
3	4	2
4	2	1

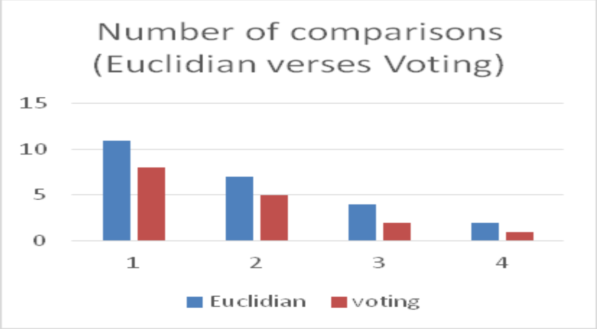

Fig. 1 Comparison between Euclidian and voting methods

From the above tables and figures, it is observed that the number of comparisons in each iteration of clustering is significantly reduced in the proposed method compared to the existing Euclidean distance based clustering. The reason behind this is the inclusion of user preferences in the computations of the proposed clustering procedure. So, it can be concluded that the rating specific computations saves the execution time and provides the richer clusters for product or voting recommendations.

10. Conclusion

Nowadays World Wide Web (WWW) is becoming popular as a front-end system to almost all relational database management systems. This facility creates new opportunities with extended more advanced query capabilities and results. Present study explains a new way of executing queries with customer voting/rating/preferences. The concept known as customer voting/rating/preferences based on the usage of queries is growing rapidly in many real-time applications. Product voting/rating/preferences based technique must be modified and enhanced with other database operations. There is a scope for optimization of voting/rating/preferences based query executions. All these database operations can be extended to fuzzy operations also. Even there is a possibility to define and use probability density function (pdf) for effective management of voting/rating/preferences weighted values of attributes. Products are clustered based on the similarity of their features. The present study proposes a new method for product clustering. The new method computes weighted values in terms of values of attributes of products and their corresponding opinion values specified by customers for each product separately. Sum of weighted values, values of attributes of products multiplied by opinion values, are computed by using a linear weighted function. Based on the weighted sums products are clustered. This new framework is more general and it produces reasonably more accurate results for clustering as well as classification of products. In future, there is a possibility to enhance the linear function by augmenting other features such as error corrections and modifications and so on.

References

[1] Achtert Elke, Christian Bohm, Peer Kroger, Peter Kunath, Alexey Pryakhin, Matthias Renz, “Efficient Reverse kNearest Neighbor Search in Arbitrary Metric Spaces”, SIGMOD 2006 June 2729, 2006, Chicago, Illinois,USA

[2] Becker H., M. Naaman, and L. Gravano, “Learning similarity metrics for event identification in social media”, in Proc. 3rd ACM Int.Conf. Web Search Data Mining, 2010, pp. 291–300

[3] Fagin Ronald, “Combining Fuzzy Information: an Overview”, Appeared in ACM SIGMOD Record 31, 2, June2002,pages109-118

[4] Fazeli Soude, Hendrik Drachsler, Marlies Bitter-Rijpkema, Francis Brouns, Wim van der Vegt, and Peter B. Sloep, “User-centric Evaluation of Recommender Systems in Social Learning Platforms: Accuracy is Just the Tip of the Iceberg”, IEEE Transactions on Learning Technologies, August 26, 2015

[5] Georgoulas Konstantinos, Akrivi Vlachou, Christos Doulkeridis, and Yannis Kotidis, “User-Centric Similarity Search,” IEEE Transactions on Knowledge and Data Engineering, Vol. 29, No. 1, January 2017

[6] Georgoulas K. and Y. Kotidis, “Towards enabling outlier detection in large, high-dimensional data warehouses”, in Proc. Scientific Statistical Database Manage,2012,pp.591–594.

[7] Haydar Charif, Anne Boyer, “A New Statistical Density Clustering Algorithm based on Mutual Vote and Subjective Logic Applied to Recommender Systems”, UMAP 2017 Full Paper UMAP’17, July 9- 12, 2017,Bratislava,Slovakia

[8] HristidisVagelis, Nick Koudas, Yannis Papakonstantinou, “PREFER: A System for the Efficient Execution of Multiparametric Ranked Queries”, ACM SIGMOD ’2001 Santa Barbara, California,USA

[9] Korn Flip, S. Muthukrishnan, Divesh Srivastava, “Reverse Nearest Neighbor Aggregates Over Data Streams”, Proceedings of the 28th VLDB Conference, HongKong,China,2002

[10] Lee Ken C. K., Baihua Zheng, Wang-Chien Lee, “Ranked Reverse Nearest Neighbor Search”, IEEE Transactions on knowledge and Data Engineering. Vol. 20, No.7, July 2008

[11] Nasraoui Olfa and Chris Petenes. 2003. “An Intelligent Web Recommendation Engine Based on Fuzzy Approximate Reasoning”, Proceedings of the IEEE International Conference on Fuzzy Systems Special Track on Fuzzy Logic and the Internet.

[12] Nagarnaik Paritosh, Prof. A.Thomas, “Survey on Recommendation System Methods”, IEEE Sponsored 2nd International Conference on Electronics and Communication System (ICECS 2015)

[13] Rajaraman A., and J. D. Ullman, “Mining of Massive Datasets”, Cambridge, U.K.: Cambridge Univ. Press, 2012.

[14] Roussopoulos Nick, Stephen Kelly, “The Nearest Neighbor Queries”, Proceedings of the 1995 ACM-SIGMOD Intl. Conf on Management of Data, San Jose, CA

[15] Sarwar B. M., G. Karypis, J. A. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms”, in Proc. 10th Int. Conf. World Wide Web, 2001,pp.285–295.

[16] Singh Amit, Hakan Ferhatosmano˘ glue, Ali Aman Tosun, “High Dimensional Reverse Nearest Neighbor Queries”, CIKM’03, November 3–8, 2003, New Orleans, Louisiana, USA.

[17] Song Linqi, Cem Tekin, Mihaela van der Schaar, “Clustering Based Online Learning in Recommender Systems: A Bandit Approach”, the material is based upon work funded by the US Air Force Research Laboratory (AFRL).

[18] Umutoni Nadine, Huiying Cao, Jiangzhou Deng “Competitive Recommendation Algorithm for E-commerce”, 2016 12th International Conference on Natural computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)

[19] Vlachou Akrivi, Charitos Doulkeridis, Yannis Kotidis, Kjetil Nrvag, “Reverse Top-k Queries”, ICDE Conference 2010 978-1-4244-5446-4/10

[20] Wang Zhibo, Jilong Liao, Qing Cao, Hairong Qi, and Zhi Wang, “Friend book: A Semantic-based Friend Recommendation System for Social Networks”, IEEE Transactions on Mobile Computing.

[21] Weng sung-shun, Mei-ju Liu, “Personalized Product Recommendation in E-Commerce”, Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE’04)

Y.Subba Reddy received M.Sc (Computer Science) degree from Bharathidasan University, Tiruchirapalli, TN and M.E degree in Computer Science & Engineering from Sathyabama University, Chennai, TN. He is a research scholar in the Department of Computer Science, Sri Venkateswara University, Tirupati, AP, India. His research focus is on Data Mining in Clustering and Similarity measures.

Dr. V. Tanuja received Master of Computer Applications from Sri Venkateswara University, Tirupathi, AP, and M.Tech in Computer Science & Engineering degree from Acharya Nagarjuna Univerity, Guntur, and Ph.D from S.V. University, Tirupati, AP, and India. She is working as Director, V.R. Institute of PG Studies. Her research focus is on Data Mining in CRM and Data mining in Clustering and Similarity measures

P. GOVINDARAJULU, Professor, Department of Computer Science, Sri Venkateswara University, Tirupathi, AP, India. He received his M. Tech., from IIT Madras (Chennai), Ph. D from IIT Bombay (Mumbai). His area of research is Databases, Data Mining, Image processing, Intelligent Systems and Software Engineering.