Personalized Trust Management in Decision Making: A Dynamic Clustering Approach

Georgios P. Papamichail[1] and Dimitrios P. Papamichail[2]

[1] Department of Management Science & Technology, Athens University of Economics and Business, Greece
[2] Computer Science Department, State University of New York at Stony Brook, New York, USA

This paper presents a personalized approach for distributed trust management by employing the *k*-means range algorithm, a combination of the partitional *k*-means clustering algorithm with orthogonal range search concepts. The aim of this approach is to aid the human or computer agent in organizing information from multiple sources into clusters according to its “trust features”. Thus the agent can perform complicated trust assessments in real-time situations and cooperate with decision-making software to assist in purchasing activities. We conclude by discussing the implications and advantages of this approach in trust management in traditional and mobile e-commerce applications.

Keywords: distributed trust management, *k*-means range clustering, personalized systems, multi-attribute trust modelling.

1. Introduction

As e-commerce and corresponding technologies mature, the issues of security and trust management become increasingly important in everyday transactions. Trust is affecting, in particular, the life of on-line communities where people are rated according to past transactions by other members (e.g. eBay) or by a centralized system (e.g. kazaa). On the one hand, people rely on other people’s recommendations, on brand names and personal knowledge to form an implicit notion of trust. It is apparent that a ’one size fits all’ approach to trust management is not satisfying an individual decision maker who desires to produce an independent assessment of trust based on past experiences, recommendations from trusted parties and personal beliefs. On the other hand, if an individual desires to purchase a particular item through the Internet, the decentralization of information makes it extremely difficult to integrate information from different sources and formats, and produce a categorization of trusted parties that offer this item according to their “trust features”. Therefore, there is a great need for the efficient filtering of information related to trust in a personalized manner.

In parallel, modern organizations develop sophisticated interactive web environments to accommodate the on-line shopping experiences of consumers. The increasing number of alternatives offered through these environments does not allow a thorough evaluation of all available alternatives but, instead, force the decision maker to identify a subset, choosing from a vast range of items and, in a second step, perform relative comparisons among these and make his decisions [10]. In purchasing activities it has been observed [5] that interactive decision aids, that is tools that assist shoppers in their purchase decisions, appear to have strong favourable effects on both the quality and the efficiency of purchase decisions. But, still the basic problem about trust management, which is the dispersion of trust information among numerous peers in the network, remain unchanged. Furthermore, reputation systems, which gather, aggregate and distribute feedback about participants’ behaviour, utilize multiple attributes in order to create an expectation that past transactions are remembered and thus determine an accurate probability for the present behaviour of the corresponding party [12].

At the same time, data clustering algorithms promise to deliver efficient solutions to many of the problems arising from the interactions of decision makers with the increasing volume of information in on-line environments. By the term *clustering* we mean the unsupervised process through which a large number of data items is classified into disjoint and homogenous groups (clusters), based on similarity. While promising in many application areas such as pattern classification, data mining or decision-making, its main drawback lies in the fact that it poses several restrictions to the decision maker when little information is known a priori about the nature of the data [7]. Therefore, the choice of an appropriate method, taking into account these restrictions, is crucial for the effective exploration of interrelationships among the data items required to make an assessment. A simple and most commonly used algorithm producing clusters by optimising a criterion function defined either globally (over all patterns) or locally (on a subset of the patterns) is the *k*-means algorithm [8]. It starts with a random initial partition and keeps reassigning the patterns to clusters based on the similarity between the pattern and the clusters. Its main drawback remains the fact that it is very expensive for the very large data sets of patterns met in real life applications, while it often converges to a local minimum.

In this approach the decision maker models his multi-attribute decision making preferences along with his trust desires in a unified way, describing each of these preferences as a range of values within a user-defined interval. These value intervals form a multi-dimensional "preference vector" with the dimensions defined by the number *d* of the decision parameters. The *d*-dimensional "preference vector" is then mapped to an iso-oriented rectangle (a rectangle whose sides are parallel to the axis). Each of the *d* preference decision dimensions corresponds to each side of the iso-oriented rectangle [9]. At the same time, each of the data items that are stored in an online data repository possesses certain values for each of the corresponding decision dimensions. These values form an *d*-dimensional data point for each data item. Thus the subset of the data items that meet the criteria set by the consumer is comprised of the data points that lie inside the iso-oriented rectangle. In the second step the *k*-means clustering algorithm is employed to classify these data points into disjoint clusters, allowing the decision maker to easily distinguish between alternatives and eliminate at the first level dominated clusters and at the second level dominated data items, using a multi-criteria decision method.

The aforementioned procedure integrates traditional multi-attribute decision approaches with mathematical trust models [1, 15] allowing the definition of a discrete set of verbal tags as input (such as strong trust, weak trust, uncertain trust, weak distrust and strong distrust) by the end-user and using them as decision arguments along with decision making preferences. It does so by combining the multi-dimensional range tree and the *k*-means algorithm to produce an orthogonal range data clustering method, that is an efficient algorithm for personalized trust management in distributed decision making applications. This approach allows the decision maker to model his decision and trust preferences along multiple dimensions, defining in this way multi-dimensional decision vectors. Then the range search reduces significantly the initial data set of patterns that need to be examined for similarity along these multiple decision dimensions, thus narrowing significantly the decision space. Finally the *k*-means algorithm produces data clusters and the decision-maker can focus more effectively on the search for optimal solutions to his needs. This approach allows the development of distributed interactive decision aids in e-commerce applications, since it relies on a set of scalable, user-intuitive, real-time algorithms with affordable time and space complexities. Additionally, it provides the decision maker with a means of comparing items across different dimensions of trust in decentralized environments and choosing, in the end, those belonging to the clusters that satisfy most of his personal trust preferences.

2. Research Approach

In this paper we expand the architecture for trust management described in [2], which relies on all system layers (see Figure 1). These include network and data management, which ensure quality of service in network connections and data replication for increased reliability. Additionally, we divide the trust management layer in two sub-layers namely trust decision-making and trust computational sublayer. The former supports mental processes of the end-user so that he can express his decision along with his trust preferences and obtain the most 'trusted

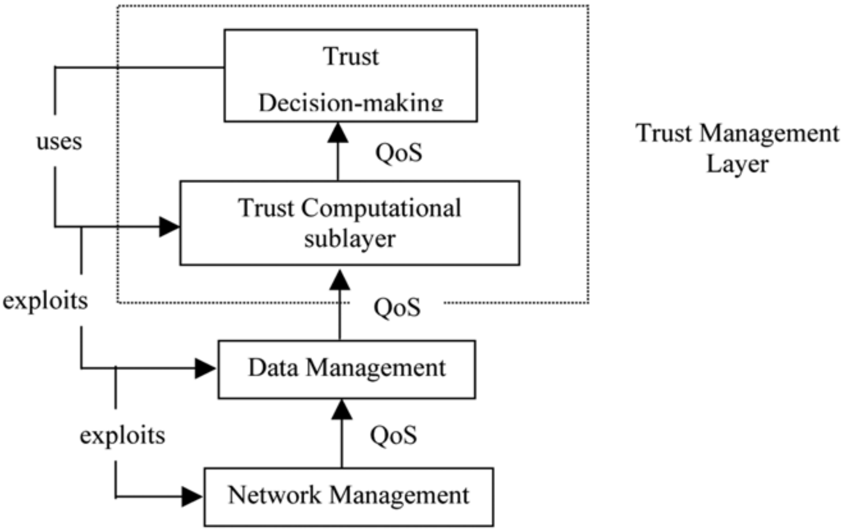

Fig. 1. Trust Management Architecture.

information'. It includes decision-making and negotiation models necessary for the human or computer agents to eliminate dominated options and decide, depending on the decision context and the desires of the end-user. The latter includes all the algorithms and heuristics for the efficient and effective representation and calculation of trust.

To serve the needs of the trust computational sublayer we employed the *k*-means range clustering algorithm, a clustering approach combined with orthogonal range searching characteristics, which represents decision and trust preferences in a unified way, based on the geometrical approach presented below. In this algorithm the decision maker firstly states his preferences defining value intervals for each one of them. For the sake of simplicity and visualisation purposes we assume he states only two. In the first one his preference lies between the values x_1 and x_2 and in the second between the values y_1 and y_2. Thus an iso-oriented rectangle is formed, named *R*, a rectangle with sides parallel to the axis. Additionally, the data items contained in a data repository are depicted as two-dimensional points with values p_i and q_i, and

$$p_i, q_i \in A = \{(p_1, q_1), (p_2, q_2), \ldots, (p_n, q_n) \mid \text{where } (p_i, q_i) \in \Re^2, i, n \in I\}$$

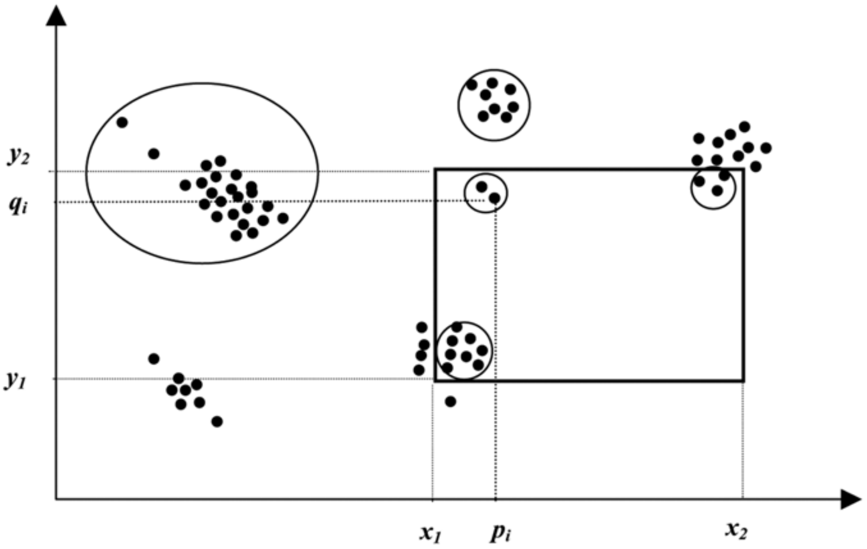

Fig. 2. The decision scene.

This representation forms the decision scene as shown in Figure 2.

The decision maker performs a two-stage process: in the first stage he identifies all the items that lie inside his preference rectangle, that is all data points (p_i, q_i) where $x_1 \leq p_i \leq x_2$ and $y_1 \leq q_i \leq y_2$, and in the second stage he focuses on the clusters formed by these data points. In Figure 2 three clusters lie inside the R rectangle and three others lie outside. The algorithm proposed in this paper serves these two-stages in a straightforward manner. First it employs a range search to determine all the data points lying inside the consumer preference rectangle and then it uses the k-means algorithm family to calculate the corresponding clusters.

What requires particular attention is the fact that for the clusters lying inside the rectangle only the points enclosed in it are reported, reducing significantly both the total number of clusters reported but also the set of data points used, resulting in an enormous reduction in computational time. Furthermore, the decision maker need not define his exact decision making strategy a priori, but can redefine his preferences along with the value intervals after he retrieves and calculates the desired data items and corresponding clusters. In the following sections we present the k-means range algorithm, which operates on both categorical and mixed values, comment on its computational complexity and discuss the implications of its deployment in trust management decision aids.

3. The k-means Range Algorithm

The proposed k-means range algorithm is a two-step process involving a multi-dimensional range search followed by a k-means clustering step in the case of numeric data points [3, 11], or a k-prototypes clustering step in the case of numeric and categorical values [6]. A formal description of the multi-dimensional range search is:

Input: A set S of n data points, so that

$$S = \{(s_1, s_2, \ldots, s_n) \mid \text{where } s_i \in \Re^d, \text{ and } i, d \in I\}.$$

A multi-dimensional rectangle R constituted by a set of two-dimensional points,

$$R = \{(x_1, y_1), (x_2, y_2), \ldots, (x_d, y_d) \mid \text{where } x_i, y_i \in \Re, \text{ and } i, d \in I\},$$

each one representing a rectangle dimension,

Output: All data points m lying inside the rectangle R.

The range tree [14] was introduced to solve the range-searching problem. It uses $O(n \log^{d-1} n)$ space and answers the range-searching problem in $O(\log^d n + m)$ time, where m is the number of data points reported. Furthermore, using a portion on the fractional cascading technique the query time is further reduced to $O(\log^{d-1} n + m)$ [13]. A d-dimensional range tree is defined recursively from the corresponding tree for the

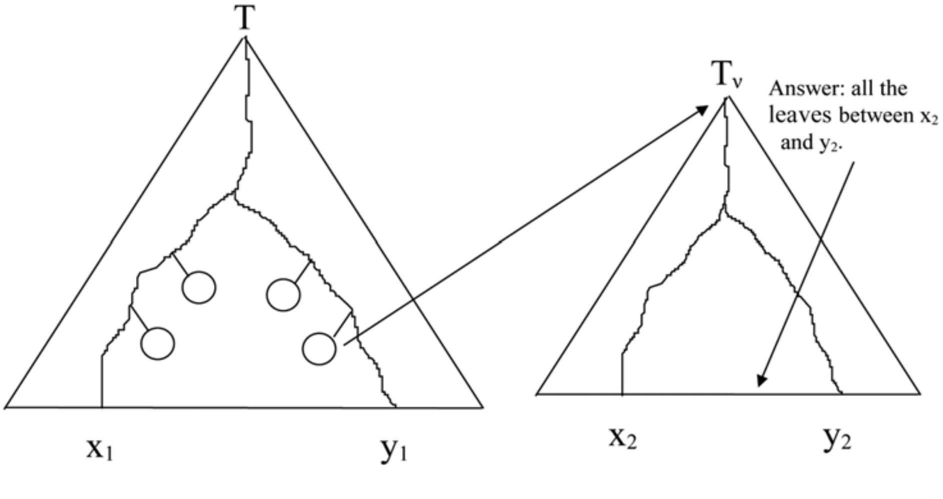

Fig. 3. The 2-dimensional range tree.

$(d-1)$ dimensional case. A 1-dimensional tree can be considered as a leaf-oriented balanced binary search tree. A 2-dimensional range tree is a 1-dimensional one, arranged according to the first coordinate of each point, having additional properties that each leaf corresponds to a point and also that a node corresponds to a 1-dimensional range tree which organizes the points descending from that node, according to their second coordinate (Figure 3). This can be generalized to higher dimensions. For d-dimensional points we maintain a balanced binary search tree which stores in its leaves the points of the given set in increasing order, according to their first coordinate. Any internal node corresponds to a $(d-1)$ dimensional range tree for the $(d-1)$ dimensional points assigned to that node.

In order to perform a range search, we begin by searching with both x_1 and y_1 in the first layer T of our structure in order to find the two leaves so that the first is the nearest after x_1 and the second is the nearest before y_1. Then we find all the nodes that are left and right sons of the nodes on the search path, but not on the search path themselves. Since T is balanced, there are at most $O(\log n)$ such nodes. On a next step, we search for both x_2 and y_2 in each subtree T_v. The union of the answers consists of at most $O(\log^2 n)$ nodes. At the end of the algorithm, we search at most $O(\log^d n)$ 1-dimensional range trees where all the requested data points are stored in the leaves.

Therefore, a description of the proposed k-means range algorithm is as follows:

```
**input** $n$ $d$-dimensional data points, a $d$-dimensional rectangle $R$.
**calculate** all data points, be it $m$, lying inside the rectangle using a $d$-dimensional range tree search.
**input** $k$ (number of the cluster means)
**initialize** $k$ means $y_1, y_2, \ldots, y_k$.
**repeat**
**for each** input data point $x_i$, $1 \leq i \leq m$
    **do**
      **assign** $x_i$ to the $j$-th cluster with nearest mean $y_j$, such that the quantity
      $\left\|x_i^{(j)} - y_j\right\|^2$ or $\gamma\delta(x_i^{(j)} - y_j)$,
      depending on the nature of the attribute, is minimum for all $j$,
      where for $1 \leq j \leq k$
  **for each** cluster $C_j$, where $1 \leq j \leq k$
    **do** recalculate the clustering accuracy
      $r = \frac{1}{|C_j|} \sum_{y_j \in C_j} y_j$
  **compute** the function
    $\sum_{j=1}^{k} \sum_{i=1}^{p} \left\|x_i^{(j)} - y_j\right\|^2 + \gamma \sum_{j=1}^{k} \sum_{i=p}^{m} \delta(x_i^{(j)} - y_j)$
**until** no data point has changed clusters (or the above quality functions become less than a given threshold).
```

In the above algorithm the clustering accuracy r is defined as the number of instances occurring in cluster C_j divided by the number of instances in the data set. The time complexity of the k-means range algorithm can be easily calculated by adding the corresponding complexities of its two parts. First the multi-dimensional range search can be solved in $O(\log^{d-1} n + m)$ time, where m is the size of the answer, as described in the previous section. The k-means part, or, more correctly, the modified k-prototypes clustering algorithm [6], has a time complexity of $O(Tkn)$, where T is the number of iterations, k is the number of clusters and n is the number of data items in the input data set. But, as the range search has produced a significantly smaller set of data points, namely m, which dominates the other two factors T and k, the whole time complexity decreases correspondingly and becomes $O(Tkm)$, depending on a large extent to the nature of the data. Therefore, the time complexity of the k-means range algorithm becomes $O(\log^{d-1} n + Tkm)$ and since $Tk \ll m$, the overall time complexity is $O(\log^{d-1} n + m)$.

4. Implications

The k-means range algorithm supports the personalized trust management processes of the individual decision maker along with his other decision preferences. Firstly, it assists the human agent to form his preferences forming multi-attribute arguments, a step that requires special attention in negotiations and deal making [4].

This takes place in an intuitive manner where the decision maker defines value intervals within which his preferences lie. Secondly, it computes the data items requested by the decision maker, using a multi-dimensional range search, taking into account the explicitly or implicitly defined trust desires of the user. In the final step it produces corresponding clusters assisting, therefore, the decision maker to classify his options. The *k*-means range algorithm is suitable for the development of e-commerce trust management applications since:

- The clustering algorithm used works efficiently both on numeric and categorical data, and also in clustering large data sets. This proves useful especially when handling verbal tags representing categories of trust (categorical data) along with numerical data (i.e. product price, size, production date etc.).
- It can be combined with other decision making and negotiation methods which can be built effectively on top of it, allowing human or computer agents to redefine their criteria and preferences based on the clusters computed. Furthermore, the decision maker need not reveal his strategy, but uses the classification produced to form it.
- It can be integrated in existing database systems as a trust management computational layer. The range search used multi layered leaf-oriented balanced binary search trees, a heavily addressed and analysed data structure, very common to the data structures used for indexing in relational database management systems.
- The decision maker can easily combine data items from various sources using appropriate web services, filter them using the range search and classify them, thus being able to integrate product catalogs from different suppliers in B2B commerce.
- It can be used in distributed asynchronous situations, where agents execute indefinitely, searching for changes or opportunities in combined trust and product patterns.
- In mobile computing models, the computational load imposed by the algorithm described in this paper can be serviced by the trust management parties, and the user may retrieve the filtered information in his mobile device through incremental steps, or apply decision making software to shape his personalized trust criteria and decision preferences.
- The trust management providers need not keep additional information on the human or computer agents apart from their preference rectangles, not even the produced data clusters, thus enhancing ethical factors such as anonymity and transparency in reputation mechanisms, capturing only changes in user status per session.

5. Conclusions

This paper presented a real-time personalized trust management approach, which can be used in decentralized decision making situations. It integrates decision making and trust management, allowing the end-user to represent his preferences using multi-attribute arguments almost intuitively. Operating as a trust computational sublayer, it fits with transparency in wider information systems. Combining two widely used computational methods capable of manipulating efficiently very large data sets, it supports human or computer agents to dynamically adjust their preferences and classify information from various sources to form alternatives and eliminate dominated solutions.

In its last part the paper also briefly discussed the implications of this approach both to the end-user and trust management providers. In the case of the former, it proposed a flexible computational framework that can be integrated in database management systems due to the nature of the trust clustering algorithm. Furthermore, it transposes the computational load to the intermediary or info-broker facilities, enforcing in this way autonomy of the final user. In the case of the latter, it proposed an ethical way to remember personal information of the decision-maker, through his initially stated preference rectangle and his final decisions. Future research involves the development of a prototype and adjustment of the decision making process to accommodate complex situations with multiple preference criteria. In parallel, special attention will be given to the decision and negotiation support structure needed to assist the end-user in adjusting his preferences, without making compromises and forming alternative trust strategies.

References

[1] A. ABDUL-RAHMAN, S. HAILES, A distributed trust model. *Proceedings of the New Security Paradigms Workshop*, (1997), pp. 48–60, ACM Press.

[2] K. ABERER, Z. DESPOTOVIC, Managing Trust in a Peer-2-Peer Information System. *Proceedings of the Tenth International Conference on Information and Knowledge Management*, (2001), pp. 310–317, ACM, New York.

[3] K. ALSABTI, S. RANKA, V. SINGH, An efficient *k*-means clustering algorithm. *Proceedings of the First Workshop on High Performance Data Mining*, (1995), Orlando, Florida.

[4] C. BEAM, A. SEGEV, M. BICHLER, R. KRISHNAN, On Negotiations and Deal Making in Electronic Markets. *Information System Frontiers*, 1 (3) (1999), pp. 241–258.

[5] G. HAÜBL, V. TRIFTS, Consumer Decision Making in Online Shopping Environments: The Effects of Interactive Shopping Aids. *Marketing Science*, 19 (1) (2000), pp. 4–21.

[6] Z. HUANG, Extensions to the *k*-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2 (1998), pp. 283–304.

[7] A. K. JAIN, M. N. MURTY, P. J. FLYNN, Data clustering: A Review. *ACM Computing Surveys*, 31 (3) (1999), pp. 264–323.

[8] J. MCQUEEN, Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[9] G. P. PAPAMICHAIL, D. P. PAPAMICHAIL, Towards using computational methods for real-time negotiations in electronic commerce. *European Journal of Operational Research*, 145 (2) (2003), pp. 3–9.

[10] J. W. PAYNE, J. R. BETTMAN, E. J. JOHNSON, The Adaptive Decision Maker. Cambride University Press, Cambridge, UK, 1993.

[11] V. RAMASUBRAMANIAN, K. PALIWA, Fast *k*-dimensional tree algorithms for nearest neighbor search with application to vector quantization encoding. *IEEE Transactions on Signal Processing*, 40 (3) (1992), pp. 528–531.

[12] P. RESNICK, R. ZACKHAUSER, E. FRIEDMAN, K. KUWABARA, Reputation Systems. *Communications of the ACM*, 43 (12) (2000), pp. 45–48.

[13] Y. K. VAISHNAVI, Computing Point Enclosures. *IEEE Transactions on Computing*, C–31 (1) (1982), pp. 22–29.

[14] D. WILLARD, New data structures for orthogonal range queries. *SIAM Journal on Computing*, 14 (1985), pp. 232–253.

[15] P. R. ZIMMERMAN, The Official PGP User's Guide, MIT Press, 1995.

Received: October, 2003
Accepted: March, 2004

Contact address:
Georgios P. Papamichail
Department of Management Science & Technology
Athens University of Economics and Business
47A Evelpidon and 33 Lefkados str.
11362 Athens
Greece
e-mail: pmichael@aueb.gr

Dimitrios P. Papamichail
Computer Science Department
State University of New York at Stony Brook
Stony Brook, NY 11794-4400
New York, USA
e-mail: dimitris@cs.sunysb.edu

GEORGIOS P. PAPAMICHAIL is an Adjunct Assistant Professor at the Department of Management Science and Technology at the Athens University of Economics and Business. His areas of interest include decision making and negotiations in e-commerce, enterprise information integration and strategic information systems for public administration.

DIMITRIOS P. PAPAMICHAIL is a Ph.D candidate at the Computer Science Department at the State University of New York at Stony Brook. His areas of interest include computational biology and bioinformatics.