1-2010

Applying soft cluster analysis techniques to customer interaction information

Randall E. DURAN
Singapore Management University, rduran@smu.edu.sg

Li ZHANG

Tom HAYHURST

Citation

DURAN, Randall E.; ZHANG, Li; and HAYHURST, Tom. Applying soft cluster analysis techniques to customer interaction information. (2010). *Marketing intelligent systems using soft computing*. 49-78.
Available at: https://ink.library.smu.edu.sg/sis_research/6451

Applying Soft Cluster Analysis Techniques to Customer Interaction Information

Randall E. Duran, Li Zhang, and Tom Hayhurst

Singapore Management University and Catena Technologies Pte Ltd
Catena Technologies Pte Ltd, 30 Robinson Road, Robinson Towers #11-04,
Singapore 048546
e-mail: randallduran@smu.edu.sg

Abstract. The number of channels available for companies and customers to communicate with one another has increased dramatically over the past several decades. Although some market segmentation efforts utilize high-level customer interaction statistics, in-depth information regarding customers' use of different communication channels is often ignored. Detailed customer interaction information can help companies improve the way that they market to customers by taking into consideration customers' behaviour patterns and preferences. However, a key challenge of interpreting customer contact information is that many channels have only been in existence for a relatively short period of time, and thus, there is limited understanding and historical data to support analysis and classification. Cluster analysis techniques are well suited to this problem because they group data objects without requiring advance knowledge of the data's structure. This chapter explores the use of various cluster analysis techniques to identify common characteristics and segment customers based on interaction information obtained from multiple channels. A complex synthetic data set is used to assess the effectiveness of k-means, fuzzy c-means, genetic k-means, and neural gas algorithms, and identify practical concerns with their application.

1 Introduction

The number of ways that companies and customers communicate has increased dramatically over the past few decades. For example, retail banking customer interactions have gone beyond branch, mail, and person-to-person phone communications to include interactions through ATMs, bank web sites, email, mobile messaging, internet chat, social networking, and virtual reality environments. Although market segmentation efforts have utilized high-level customer interaction statistics – such as the frequency of interactions with a customer – in-depth information available regarding customers' use of different communication channels is often ignored. Making use of detailed customer interaction information can improve the way that organizations characterize customers' behaviour and preferences. Consequently, this

J. Casillas & F.J. Martínez-López (Eds.): Marketing Intelligence Systems, STUDFUZZ 258, pp. 49–78.
springerlink.com

knowledge, either alone or combined with other demographic information, can provide marketing efforts with a competitive advantage.

Unlike most traditional sources of data used for customer segmentation, there is limited historical context for interpreting customer interaction information; many of the channels have only been in existence for a relatively short period of time, and new ones are continuing to evolve. Unsupervised classification techniques, such as cluster analysis, are well suited to help address this challenge because they group data based only on descriptions of the data and their relationships, which are extracted directly from the raw information without requiring advance knowledge of its structure. Furthermore, within the domain of cluster analysis methods, techniques that make use of fuzzy logic and artificial intelligence – such as genetic and neural algorithms – have the potential to provide unique insights into customers' behaviour patterns and achieve superior computational efficiency.

This chapter explores the use of various cluster analysis techniques to identify common characteristics and segment customers based on interaction information, such as the frequency, time, duration, and purpose of each interaction across multiple channels. The effectiveness of k-means, fuzzy c-means, genetic k-means, and neural gas algorithms is assessed to provide an understanding of the techniques' effectiveness and identify practical concerns with their application. Specifically, the goal of this research is to answer four questions: Can customer segments be identified only using customer interaction data? How accurately are the segments drawn? How well do clusters match known customer profiles? How well do soft computing approaches to cluster analysis perform, as compared with traditional methods?

In order to illustrate its relevance, the analysis is presented in the context of supporting the marketing activities of a retail bank. The effectiveness of the clustering is assessed using synthesized data sets that include interaction patterns that represent different retail banking customer groups. Starting with a synthetic data set that has a known composition enables the effectiveness of the cluster analysis to be evaluated independently from variations and uncertainties in the real data to which it is applied. Trying to validate these techniques using data derived from real-world customer interactions would be very difficult. In this case, there might be multiple meaningful customer groupings and the cluster analysis could identify ones that do not correspond to groupings derived using other approaches, making the comparison and validation of different approaches problematic. Furthermore, lack of underlying information could make it more difficult to correlate and verify the groupings, thus raising doubts regarding the validity of the clustering results. For example, distinct clusters might be identified for part-time workers who are also students and part-time workers who are not, but it would be difficult to confirm this distinction if the bank's customer records did not have recent information about customers' school enrolment status. Using synthetic data to support the evaluation of the clustering methods avoids this concern.

The structure of this chapter is as follows. The first section provides a literature review of cluster analysis, discussing its use within the financial services industry and for customer relationship management (CRM). The second section outlines a business context and discusses how the synthetic interaction data were constructed. The third section describes the research approach. The fourth section presents the results. The conclusion identifies practical applications of this research and identifies further areas of investigation.

2 Literature Review

Cluster analysis as a statistical tool has been actively studied in several fields such as statistics, numerical analysis, and machine learning. From a practical perspective, it has played an important role in various data mining applications in the domain of marketing, CRM, and computational biology. The following section provides a brief introduction to basic cluster analysis concepts and lists a few of its applications in the financial services industry.

2.1 Background of Cluster Analysis

Cluster analysis is a collection of techniques for dividing a set of objects into meaningful groups based on features that describe the objects and their relationships (Tan 2005). The desired result is that objects within a group should tend to be similar to one another, and objects in different groups tend to be less similar. This similarity is typically measured as the “distance” between each pair of objects, according to a metric appropriate to the type of data being measured. To help illustrate this concept, Fig. 1 shows a simple example of how a bank could use cluster analysis to segment its customers. Each customer is described by the number of credit cards they possess and how often they have made online bill payments over the past two years. Three clusters can be obtained, as shown in Fig. 1. Cluster 1 is comprised of customers that have a large number of credit cards and make many online payments; cluster 2 contains customers that have an intermediate number of credit cards and make relatively few online payments, and cluster 3 contains all customers with few credit cards who rarely make online payments. This simple example only uses two variables, whereas a cluster analysis will typically involve many more.

Cluster analysis can be regarded as an unsupervised classification method, because it classifies data based on their underlying structure or characteristics. In contrast, supervised classification techniques assign a class label to new objects according to an existing model that is based on objects with known labels. For example, supervised classification methods could be used to label credit card applications as either ‘approved’ or ‘rejected’ according to a model derived from a pre-existing set of applications with well-understood characteristics. Conversely, cluster analysis methods would divide a set of credit card applications into multiple groups, whose underlying characteristics could then be used as the basis for deciding whether to approve the applications, group by group. In market research studies, both supervised classification methods and cluster analysis methods are used to divide data into different segments. Supervised classification methods find segments characterized by predicted customer behaviours and are mostly used for targeted labelling, whereas cluster analysis methods are more explorative and are often used to discover unknown groups, without any *a priori* information that is used as a training set.

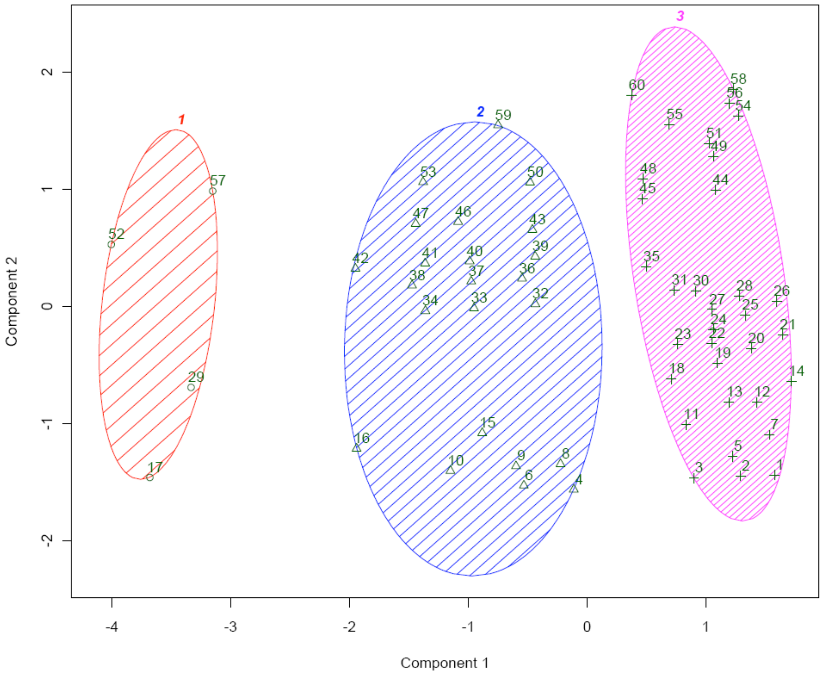

Fig. 1 A sample clustering of bank customers

Clustering methods can be broadly categorized as partitional, hierarchical, or overlapping (Hruschka 2009). Partitional clustering methods divide a set of objects into a number of non-overlapping clusters, the number of which is usually predefined, such that each object is in exactly one cluster. K-means clustering (MacQueen 1967) is a widely used partitional clustering algorithm. The k-means algorithm first allocates a number of randomly selected points to be the initial centre of each cluster, and then assigns each object in the data set to the nearest centre to form clusters. The cluster centres for the next iteration are then assigned to be the centres of the clusters from the previous iteration, and the objects are reassigned to the new centres. This process is repeated until the centres are stable and do not change with subsequent iterations.

Hierarchical clustering is an alternative to partitional methods. These methods distribute the data into a set of nested clusters organized as a tree structure. Agglomerative methods start with as many clusters as objects in the dataset and repeatedly merge the two closest clusters until a single cluster remains. Divisive methods start with a single cluster containing all data and then repeatedly split clusters until a stopping criterion is met (Han 2001). Both partitional and hierarchical methods can be considered to be exclusive, or crisp, clustering methods because each object is placed in exactly one cluster.

Alternatively, overlapping clustering methods can assign objects to more than one cluster. Fuzzy clustering, for example, allows each object to belong to multiple clusters; each object will be associated with a cluster based on a weighting between zero and one. Each object's total weight across all associated clusters is equal to one, or, in other words, each cluster shares a portion of the object. The fuzzy c-means (FCM) algorithm is a popular fuzzy clustering algorithm. It is broadly similar to k-means, but instead of assigning points to their closest cluster centre, a membership degree is defined to describe the proximity of the object to each cluster centre (Nikhil 1996).

Many of these clustering methods depend on randomly generated initial clusters. It is possible that, depending on the starting configuration, some clusters never have points assigned to them and, as a result, the clustering process can terminate with a sub-optimal solution. One way to address this problem is to run the clustering algorithm repeatedly with different, randomly generated, initial values, and then select the solution that minimizes the objective function, which defines the evaluation criterion of the solution. However, this approach can be very time consuming, and there is no guarantee that an optimal solution will be achieved within a given number of iterations.

Another approach to finding an optimal solution is to use evolutionary algorithms. These produce clusters by iteratively sampling clustering solutions from the search space, evaluating them against the objective function, and applying a mutation, crossover, or selection operator to generate new solutions. While evolutionary algorithms do not guarantee that an optimal solution will be found, they tend to generate more promising solutions during the exploration of the search space. Evolutionary algorithms, therefore, have a higher probability of reaching an optimal solution with fewer random initializations than repeatedly running k-means, although each run might take a longer time (Hruschka 2009). Genetic algorithms are a common type of evolutionary algorithm.

Like evolutionary algorithms, competitive learning algorithms can also be used to determine optimal clustering solutions. Some competitive learning methods can also be used to automatically find the optimal number of clusters (Fritzke 1997). Competitive learning algorithms iteratively adapt the locations of cluster centres based on the input data and gradually move towards the optimal solution using neural network methods. There are two categories of competitive learning algorithms: hard competitive learning and soft competitive learning. Hard competitive learning methods, such as k-means, use a "winner-takes-all" approach during the adaptation of the winning cluster centre for each input data point. Soft competitive learning methods address k-means' sensitivity to the initial values' positions by using a "winner-takes-most" approach during the adaptation of cluster centres, so not only the winning cluster centre is adapted, but also some or all the other centres. For example, the neural gas algorithm (Martinetz 1993), is a competitive learning algorithm that ranks the cluster centres according to their distance to each given data point and then adapts them in the ranked order to move towards the optimal solution. Another, similar, technique is the Self-Organizing Map (SOM), originated by Kohonen (2001). The difference between neural gas and SOM is that neural gas does not have a topology imposed on the network, while SOM has a fixed network dimensionality which makes it possible to map the usually large n-dimensional input space to a reduced k-dimensional structure for easy data visualization (Fritzke 1997). Nevertheless, studies have shown that the traditional k-means clustering method has produced higher classification accuracy than neural networks using Kohonen learning (Balakrishnan 1994).

2.2 Applications in the Financial Services Industry and Customer Relationship Management

Cluster analysis has been used as a multivariate statistical modelling technique in the financial services industry for a variety of purposes. Credit risk managers have combined supervised and unsupervised classification techniques to evaluate credit risks. Zakrzewska has investigated the combination of cluster analysis and decision tree models by first segmenting customers into different clusters characterized by similar features and then building decision trees that define classification rules for each group separately (Zakrzewska 2007). Each credit applicant was assigned to the most similar group from the training dataset, and their credit risk was evaluated based on rules defined for the group. Results of the cluster analysis on credit risk datasets demonstrated greater precision than decision tree models.

Cluster analysis has also been used in credit card portfolio management to identify potentially bankrupt accounts, fraudulent transactions, and distressed credit card debt (Allred 2002; Peng 2005). Clusters of accounts can be used to predict credit card holders' behaviours, allowing appropriate policies to be developed for each individual cluster. Likewise, Edelman has applied an agglomerative hierarchical clustering method to group monthly credit card payment transactions so that the groupings can be used to assist in the scheduling of resources allocated to address delinquent accounts (Edelman 1992).

Another growing application area for cluster analysis is customer relationship management (CRM), which utilizes data from various sources, including demographic information, transaction history, and call centre activities. CRM evaluates customer behaviour, such as spending habits, to help optimize and fine-tune marketing and pricing strategies. For example, as part of a CRM survey, a large sample of respondents could be divided into different market segments according to a number of variables related to consumer behaviour. Appropriate services and products can then be tailored to suit each particular market segment and therefore achieve the highest efficiency and profitability. Balakrishnan et al. (1996) have applied both competitive learning and k-means algorithms to generate clusters of coffee brand choice data that supported strategic marketing decisions. A combination of both methods was found to provide useful segmentation schemes.

Most of these applications have made use of traditional clustering methods, and have relied primarily on demographic and transactional data. However there is doubt as to how useful this information is for practical business purposes such as predicting customer profitability (Campbell and Frei 2004). In contrast to previous efforts, the remaining sections of this chapter will examine how fuzzy and artificial intelligence-based clustering methods can be applied to customer interaction information.

3 Business Context and Data Used for Analysis

This section introduces the business context of this study emphasizing on the pervasive multi-channel customer interaction data available and the benefits it can bring to marketing practices with the help of cluster analysis. The synthetic data used for analysis is also described including its structure and characteristics, which are carefully designed to simulate the real customer interaction pattern.

3.1 Business Context

The business context for this analysis is that of a bank attempting to obtain meaningful marketing insights from its interactions with retail customers. Retail customers typically use many different channels, including the call centre, voice recognition units (VRU), text messaging, Internet web sites, dedicated mobile web sites, physical branches, and automated teller machines (ATMs). Use of a combination of these channels by banking customers is common in developed nations. Because of banks' rapid adoption of new channels, this business domain provides good potential for utilizing cluster analysis techniques to analyse and segment customer information. Multi-channel customer interaction data analysis could be performed independently, or in support of existing data mining and segmentation practices.

Electronic channels, particularly text messaging and the Internet, have the potential to supply rich information about customers' behaviour; however, because they are new, their usage is not well understood. Accordingly, Sinisalo et al. recommend that when supporting "next generation" channels, firms should go beyond demographic and psychographic data, and use behavioural data to profile and categorize customers (Sinisalo et al. 2007). Customers that have similar behaviour patterns can then be grouped together for analysis and servicing. Multi-channel interaction information provides a fertile source of data for achieving this objective. Furthermore, while the demographic data that is commonly used for customer marketing analysis is widely available, customer interaction data is a new and relatively untapped resource.

One application of segmentation using interaction information would be to correlate identified customer groups with marketing considerations such as sensitivity to fees, product type preferences, loyalty, and default risk. Such information could be used to help determine the best products and services to offer to those customer groups. Customers who interact primarily via branches and call centres and consistently have lengthy interactions might be classified as "chatters" who highly value human interaction as part of the banking experience. Based on this interaction-based insight, a dedicated relationship manager could be included in a bundled service package that the bank could offer specifically to chatters. Beyond revenue generation purposes, effective classification of customer behaviour patterns can also be useful for risk control purposes. For example, certain customer segments may not be offered credit products if, according to their interaction characteristics, as a group they have a higher propensity to default. Moreover, previous research has shown that banking customers' use of a specific channel can be correlated to their economic value to the enterprise, even when controlling for demographic differences (Hitt and Frei 2002).

3.2 Synthetic Data Structure and Design

A key objective of this study is to evaluate the effectiveness of cluster analysis by running it on a data set where the factors driving interaction behaviour are completely understood. Using real-world customer data for this purpose would be impractical, since it would be extremely difficult, if not impossible, to obtain information about all the pertinent factors that influence each customer's behaviour. Understanding these underlying factors is necessary to effectively evaluate the results of the cluster analysis. Synthetic data sets were, therefore, designed independently from the cluster analysis implementation. To avoid biasing the research approach based on knowledge of the input data, the data set design parameters were only shared with the analysis team towards the end of the analysis.

The data sets were generated algorithmically and represented different types of retail banking customers. The goal was to produce realistic, complex sets of data that characterized different user groups and subgroups, which can then be used to determine how accurately the cluster analysis could identify the underlying customer groups as segments based on their interactions. Specifically, the data generation was driven by the following factors:

- Who – their age range and lifecycle stage
- Why – the purpose of their interaction
- When – time of day and day of week of the interaction
- Where and how – which channel was used for the interaction

Who was the primary driver for determining the interaction pattern. Customers were broken up into three primary groups and eight subgroups, which produced eleven subgroup-category combinations. The timing of customer interactions was generated by sub-group specific functions that took into consideration biases of that group towards times of the day and days of the week when they would contact the bank. Channel access rules were also taken into account, whereby branch access was limited to weekday business hours and from 9am to 1pm on Saturday. The interaction frequency – defined as the average number of interactions per month – was also varied by subgroup.

Detailed interaction profiles were defined for each of the customer subgroups. These profiles describe the purpose of the interaction, in what proportion different channels are used for each interaction type, and the duration of each interaction, according to channel. Table 1 shows the profile summary for one subgroup, Working High School Students.

While the synthetic data were designed to be realistic, some relevant factors were knowingly omitted due to overall project scope. Specifically, the synthetic data had the following limitations: 1) the data only included customer-initiated interactions; 2) only a subset of the available channels and transaction purposes were represented; and 3) interactions were distributed evenly throughout the days of the month. However, it is not expected that expanding the scope and complexity of the data set to address these concerns would significantly affect the results of the cluster analysis.

Table 1 Sub-group data construction parameters for Working High School Students

Interaction Purpose	*Proportion of all sub-group interactions*	*Interaction Channel*	*Proportion of purpose*	*Duration (norm. dist.)*	
				mean	*stdev*
Deposit	20%	ATM	70%	60	30
		Branch	30%	180	60
Withdrawal	40%	ATM	90%	80	30
		Branch	10%	200	60
Acct Application	5%	Internet	50%	800	300
		Branch	50%	300	100
Account inquiry	20%	Internet	40%	200	80
		Branch	10%	300	100
		Call Centre	20%	300	100
		Mobile-SMS	30%	30	2
Marketing inquiry	10%	Internet	30%	200	80
		Branch	30%	300	100
		Call Centre	40%	300	100
Funds transfer	5%	Internet	40%	150	80
		Branch	5%	300	100
		Call Centre	25%	300	100
		Mobile-SMS	30%	30	2

3.3 Synthetic Data Group Characteristics

The synthetic data design was documented in a tabular form that spanned eight pages. A simplified, more qualitative presentation of the customer subgroup characteristics is provided as follows. Abbreviations for each of the subgroups, or customer types, are provided in parentheses for later reference.

- *High school students* – have relatively few interaction purposes and transact exclusively after school and on weekends. As a group, they interact relatively infrequently and favour automated channels. Working high school students (SHW) have proportionately more withdrawals than non-working students (SHN).

- *University students* – have a wider range of interaction purposes, including interactions related to credit cards. They also perform more electronic fund transfers, i.e. bill payments, than high school students and favour automated channels. Working university students (SUW) interact evenly across the 8am to

12pm time period, at a medium frequency. Non-working students (SUN) favour the evening and weekends and interact at a low frequency.

- *Workers* – have the widest range of interaction purposes, including interactions related to loans, such as mortgages. They have a balanced use of different channels, not favouring automated channels over any other, and interact at a medium frequency. Full time workers (WAF) interact mostly before work, during lunch breaks, after work, and on weekends. Part time workers (WAP) interact evenly between 6am and 10pm.
- *Unemployed* – are similar to Workers but have fewer fund transfers and more account inquiries. Unemployed customers (WAU) tend to favour the branch and call centre channels over the Internet. They interact relatively infrequently and do so evenly between 8am and 12pm.
- *Domestic* – are similar to Workers but do relatively more funds transfers and transact evenly between 6am and 12pm. Domestic customers (WAD) interact at a high frequency.
- *Retired-age workers* – favour the branch and phone channels over automated channels, interact at a medium frequency, and have longer interactions than other groups. Otherwise, retired age customers who work full time (RAF) have similar interaction characteristics to those of Workers, except that they have fewer application-related interactions and more withdrawal-related interactions. Retired-age customers who work part time (RAP) are similar except that they do proportionately more deposit interactions and favour the daytime during weekdays to interact.
- *Retirees* – have interaction behaviour that is very similar to retired-age workers, but interact at a low frequency. Like retired-age part-time workers, retirees (RAN) prefer to interact during the daytime on weekdays.

The structure of the synthetic data was designed to simulate interaction patterns of actual subgroups of the general population. While each of the groups had its own unique interaction characteristics, there was also significant overlap between their behaviour patterns. Customer age, the high level partition between the groups, was not included the data sets provided for analysis, since a main objective was to determine whether the clustering techniques could identify meaningful segments without the support of demographic information.

A clean data set, where all the customers consistently followed their prescribed behaviour patterns, was produced to serve as a baseline. However, it is unlikely that in a real-world environment that such consistency would be found. Therefore, data sets with different levels of random noise were also produced. Noise was quantified as the percentage of customers' interactions that would follow a random pattern rather than the prescribed behaviour patterns. Additionally, a data set was generated that included a group of hybrid, or "transitional", customers, who exhibited one group behaviour during the first half of the time period and another group of behaviour during the second half. Specifically, the transitional group's interactions over the time period alternated between unemployed and full time employed behaviour patterns.

4 Research Approach

The research approach applied multiple cluster analysis techniques to simulated multi-channel customer interaction data, as discussed in Section 3. The primary objective was to assess the effectiveness of different algorithms and determine their usefulness in different situations. Whereas cluster analysis can involve a number of different steps (Nargundkar 2000) – such as variable selection, data validation, data standardization, addressing of outliers, algorithm selection, determining the number of clusters, and validation of results – they may not all be relevant depending on the context of the analysis. The general process and how it was applied to the interaction-based customer data are discussed in the following subsections. Additional details are presented in the experimental results section.

4.1 Variable Selection

Variable selection is the first step of the cluster analysis process. It determines the dimensions used in the cluster analysis. The number of suitable variables often depends on the data being analyzed and the granularity of the clusters desired. The selection process can be done either through judgmental selection, which is to choose the variables manually, or by factor analysis, which is to define the selected variables as a set of factors, usually extracted as a linear combination of an initial set of variables (Goldberg 1997). For the purposes of this study, judgmental selection was chosen over factor analysis because the features of the clusters could be easily derived and analyzed from the variables directly, rather than requiring to be extracted through factor analysis.

Judgmental selection of variables requires a good understanding of the data being analyzed and how well the variables reflect the characteristics of the data. When judgmental selection is used, it is beneficial to select more variables than necessary at the beginning and then eliminate redundant ones after performing several iterations of cluster analysis. Assessing the spread of cluster means across all dimensions can be used to determine which of the variables are useful and which ones should be dropped (Nargundkar 2000).

Interpreting customer interaction data derived from multiple channels is a challenging task given the large number of user behaviour variables associated with each channel. For example, when a customer communicates with the bank via a call centre, the bank can record when the communication starts and ends, who initiated it, and its purpose. Customer communications initiated through the bank's web site will produce similar information, which can be obtained from web server logs and user session data (Rho 2004). The customer interaction record was initially defined as the set of characteristics common to all of the communication channels.

Table 2 shows the set of variables selected for the customer interaction data and their defined values. The raw data were then transformed into customer description data, where data points describe a customer's interactions with the bank over a period of time. Details of this transformation are presented in the experimental results section.

Table 2 Selected variables for customer interaction data

Variable Name	*Defined Values*	*Value Type*
Customer ID	e.g. 998831	Nominal
Channel	Branch / ATM / Call-Centre / SMS / Web	Nominal
Purpose	Account Inquiry / CCA Inquiry / Marketing Inquiry / CCA Application / Account Application / Loan Application / Withdrawal / Deposit / Funds Transfer	Nominal
Initiator	Customer / Bank	Nominal
Date and time	e.g. 2009-03-03 12:41:37 PM	Interval
Duration	0 ~ 3600 sec	Ratio

4.2 Data Validation

When preparing data for cluster analysis, it is generally necessary to validate the data. Invalid values should be removed if they cannot be fixed or replaced. However, because the data set analysed was synthesized and flaws were not included by design in the data set, this step was not relevant to the cluster analysis in this particular case. While it could have been possible to include flaws in the synthesized data, doing so would not have yielded any significant benefit, since this effort was mainly focused on assessing the effectiveness of clustering algorithms on the data.

4.3 Data Standardization

It is necessary to map the variables being analyzed to an equivalent scale so that the clustering algorithms can effectively compare different variables, regardless of how they were originally measured. How variables are standardized will depend on their value type. For example, nominal variables may be standardized by creating multiple binary variables for each of the nominal states and grouping them in order to avoid the influence of increased number of predictors. Interval and ratio variables can be standardized by normalizing the values to have a mean of 0 and standard deviation of 1.

When analysing the customer interaction information, standardization was only performed on the aggregated customer description data since these are the data used for cluster analysis instead of the customer interaction data. To illustrate how the standardization was put into practice, consider the following case. The total number of interactions per customer was measured over a given period of time. Sample values ranged from 9 to 116, with a mean of 52, and a standard deviation of 19. Likewise, the proportion of interactions via branch was measured as a ratio ranging from 0 to 1, with a mean of 0.25, and a standard deviation of 0.15. By normalizing both variables to have a mean of 0 and standard deviation of 1, both variables will make an equal contribution to the similarity measurement in cluster analysis.

4.4 Addressing Outliers

Observations that deviate significantly from the rest of the data, referred to as outliers, are common in data sets. It is often the case that outliers represent unusual behaviours or erroneous data. Hence, including outliers can bias cluster analysis results. Once the data have been standardized, outliers can be identified, based on how many standard deviations the points are away from the mean in each dimension. If a data point is too far from the mean, it often indicates an outlier. As was the case with data validation, because the data set used for analysis was synthesized, it was assumed that no erroneous data were present. Furthermore, it was of interest to see how the cluster analysis algorithm would organize the entire data set. Based on this rationale, no outliers were removed from the data set.

4.5 Algorithm Selection

As discussed in Section 2.1, the most appropriate clustering method to use will normally depend on the characteristics of the data being analysed. However, because a key objective of this research was to compare the effectiveness of different types of algorithms, multiple techniques were applied. In order to ensure that the same customer segments can be consistently identified from the same set of data, the chosen clustering algorithm should be as stable as possible. This makes evolutionary algorithms and soft competitive learning algorithms potentially good choices for evaluation, for the reasons discussed in Section 2.1.

Four clustering techniques were applied to customer interaction data, to compare their effectiveness. The first technique was the traditional k-means algorithm. The second technique was the genetic *k*-means (GKM) algorithm, which uses the same objective function as *k*-means but with an evolutionary approach to searching the solution space. The third technique was the neural gas algorithm (NG), which is based on soft competitive learning. Subsequently, the most efficient and effective algorithm of these three was then compared with a fuzzy clustering algorithm. In summary, K-means, genetic k-means (GKM) and neural gas algorithm (NG) were selected as crisp techniques; the fuzzy c-means algorithm was selected as fuzzy algorithm.

4.6 Decide Number of Clusters

When applying the above-mentioned clustering algorithms, the number of clusters must be chosen in advance. Determining a suitable number of clusters is important, since using too few clusters is likely to result in very broadly characterized clusters that do not show the complete structure of the data set, while using too many clusters may mistake random noise in the data for actual information. The idea is, therefore, to pick a number that produces a clustering solution that is both statistically good and contains meaningful clusters with respect to the data being analysed.

Compactness and separation are two commonly used criteria to evaluate clustering results. High compactness means that the data points within each cluster are close to each other. High separation means that the clusters themselves are widely spaced. Ideally, a good clustering should have both high compactness and separation. While there is no perfect way to determine the optimal number of clusters, they are commonly chosen by visual inspection or computation of statistical measures.

To visually determine a good cluster number, the selected clustering algorithm was repeatedly run using different numbers of clusters. For each clustering, the clusters were plotted in the dimensions of the two principal components that explain the largest variances of the data. These cluster data plots were then used to assess the compactness and separation of the clustering results. In addition, the normalized cluster means for each variable dimension were computed and each cluster's normalized cluster means sorted according to their absolute values. The highest-ranked means for each cluster showed the dominant features for each cluster. The feature representation of each cluster helped to interpret the "meaning" of each clustering solution. The two methods can be combined to choose a number of clusters that gives a cluster plot with compact and well-separated clusters, and where each cluster has meaningful characteristics.

Statistical methods can also be used to estimate the optimal number of clusters. For crisp methods, the simplest measure of compactness is the within-cluster sum of squares (WSS) metric and the simplest measure of separation is the between-cluster sum of squares (BSS) metric (Tan 2005). The Calinski and Harabasz index (CHI) (Calinski 1974) and the Hartigan index (HI) (Hartign 1975) are both based on WSS and BSS measures. These techniques can be viewed as line charts that compare the number of clusters on the x-axis to the index values on the y-axis as well as the successive differences of the index values. Where the chart of the successive differences is convex, the knee point in the curve is the place where the transition occurs from substantive clusters to erroneous clusters. This provides a good indication of the optimal number of clusters. The Dunn index (DI) can also be used to measure both compactness and separation in terms of intra-cluster and inter-cluster distances (Dunn 1974). The maximum Dunn index value defines the optimal number of clusters. The Silhouette Coefficient (SC) also provides a measure of compactness and separation (Kaufman 1990). The maximum of the average silhouette coefficient of all points determines the optimal number of clusters. In addition, the Hubert gamma statistic evaluates the separation of the clusters, which is maximized at the optimal number of clusters (Halkidi 2001).

In the field of fuzzy clustering analysis, two frequently used cluster validity indexes are partition coefficient (PC) and partition entropy (PE) (Bezdek 1974). Both indexes measure the fuzziness of a partition based on the membership values. A higher partition coefficient value and a lower partition entropy value signify a less fuzzy partition, and, hence, denser clustering. Xie and Beni introduced an XB index that measures the ratio of total variation of the data points with respect to the cluster centres to the minimum total separation between the cluster centres. The smaller the XB index, the better the clustering solution (Xie and Beni 1991).

Both visual inspection and statistical measures were used to analyze the results of using the different clustering algorithms, for ranges of three to sixteen clusters. For visual inspection, cluster plots and cluster feature representations were produced. For statistical analysis, five index values – DI, SC, Hubert gamma, CHI and HI – were computed for the crisp clustering algorithms. Three index values – PC, PE and XB – were computed for the fuzzy c-means algorithm. A subjective combination of the visual and statistical assessment was then used to determine the optimal clustering.

4.7 Validate Clustering Results

Once an optimal set of clusters has been generated, it is important to evaluate how well the clustering algorithm has partitioned the data set. One simple way is to verify visually whether the clusters are well separated. However, this can be rather difficult, especially for high dimensional data sets. Therefore, procedures have been proposed to evaluate the results of a clustering algorithm. There are three ways of evaluating cluster validity (Halkidi 2002). The first approach is based on external criteria, by comparing the result with a pre-defined clustering structure that reflects the a priori characteristics of the data. The second type is based on internal criteria, by evaluating the result against some statistics derived from the data itself such as a proximity matrix. The statistics discussed in Section 4.6 can be used as internal evaluation criteria to determine how good a clustering solution is without comparing with other clustering solutions. The third type is based on relative criteria, which are mainly used to compare clustering solutions resulting from the same algorithm but with different parameters.

To compare different crisp clustering techniques applied to the same data set, the corrected Rand index (Gordon 1999) can be computed, to measure the level of agreement of the class labels. A high value for this measure indicates a high level of agreement. A similar measurement is the contingency table, which computes the number of data points that fall into the same clusters between two clustering solutions. In this study, both measurements were used as relative criteria to evaluate different solutions derived from each crisp algorithm – traditional k-means, genetic k-means, and neural gas – comparing different runs of the same algorithm, and also to compare solutions produced by different algorithms from the same data.

Since the structure of test data set was known, the clustering results could be compared with a known baseline. In the study, comparison against the baseline was used to assess the effectiveness of both crisp and fuzzy clustering algorithms on clean and noisy data. To evaluate how closely the fuzzy clustering results matched the known class labels, the Fuzzy Rand index was calculated which is based on the membership correlation between the data points (Campello 2007). This is a common validation method based on external criteria.

In order to evaluate whether a clustering algorithm was consistently producing the same clustering, it was necessary to compare two clustering solutions derived from different data sets with the same underlying customer interaction behaviour characteristics. This comparison was difficult and there have not been any active studies in this area. A feature matching approach that computes the percentage of matching features in each cluster for the two clustering solutions was used for this purpose. A high matching score indicates a high similarity between the two clustering solutions. Detailed validation procedures are illustrated in the experimental results section.

5 Results

Once the synthetic customer interaction data had been generated, the first step was to generate the customer description data from the interaction events. The customer description data was then input into the clustering algorithms to partition the set of customers into different clusters with unique characteristics. All the data processing and cluster analysis procedures were implemented in the R language and computation environment. R was chosen because it provided off-the-shelf clustering algorithms and was well suited for data manipulation and graphing. All the experiments were run on a Core 2 Duo 2.4 GHz machine with 2GB RAM.

The following subsections discuss the results produced at each stage of the cluster analysis when applied to two similar but distinct synthetic data sets. The final subsection provides a qualitative discussion of the results and their implications.

5.1 Variable Selection

The primary synthetic data set that was analyzed contained 100,442 interactions performed by 1,933 unique customers. Initially 24 variables that describe customers based on their interaction history were identified. Of the 24 variables, five describe the percentage of interactions a customer makes via each of the five different channels; nine describe the percentage of interactions a customer makes for each of the nine different purposes; another eight variables describe the percentage of interactions a customer makes during different time frames of the day, different days of the week and different periods of the month; and the last two variables capture the number of interactions a customer makes and the average interaction duration across all channels. The average interaction duration was defined as the normalized average duration across all channels. For each customer, the mean duration of use of each channel was first normalized to the interval of 0 to 1 with respect to the minimum and maximum duration of all interactions using that channel. The duration variable was then computed as the mean of all the channel mean durations. To help represent usage patterns more accurately, variables were defined as the percentage of total interactions rather than as the absolute number of interactions.

Based on this initial set of variables, clustering results were generated for different numbers of clusters. Cluster means were then calculated across all dimensions for each cluster to help determine which variables were superfluous and should be eliminated. In particular, variables with very small min-max cluster mean spread and small cluster means were considered as non-discriminative, which were eliminated from the cluster analysis.

5.2 Standardize Data

Since all the values were obtained from the synthetic data set, it was assumed that there were no invalid or missing values. Next, to put all variables on an equivalent scale, the values were standardized to have a mean of 0 and a standard deviation of 1 across all customers. Table 3 shows the original and standardized values of

Table 3 Original and standardized values for a subset of variables for one customer

Variable Name	*Original Value*	*Mean*	*Standard Deviation*	*Standardized Value*
Interactions	30	51.96	18.81	-1.167
Branch	0.100	0.2501	0.1509	-0.9973
AcctInquiry	0.1667	0.1329	0.0702	0.4812
Duration	0.1470	0.1940	0.0395	-1.1899

several variables. The original values were specific to a particular customer, whereas the mean and standard deviation were calculated from the number interactions for all the customers.

5.3 Decide Number of Clusters

As discussed in Section 4, because the synthetic data set did not contain any outliers the candidate clustering algorithms were applied direct to the derived customer description data, and the optimal number of clusters was then assessed using both visual inspection and computed statistical measures. The following paragraphs illustrate how different metrics can be applied to the clustering results generated by k-means and the fuzzy c-means algorithms to determine the optimal number of clusters. These two algorithms serve as examples of crisp and fuzzy clustering techniques, respectively.

Since, due to the random initialization problem, both k-means and fuzzy c-means algorithms do not always produce a stable and optimal solution, repeating the clustering process many times helps to increase the likelihood of obtaining a stable solution. Hence, they were both run 3,000 times, and the clustering solution with the minimum value of the objective function was chosen. It was determined that 3,000 repetitions was a sufficiently large number to guarantee a relatively stable, close-to-optimal solution for the data sets examined.

In order to visualise the clustering results, they were displayed using the CLUSPLOT algorithm (Pison 1999), which shows how the points are distributed according to the two principal components, and represents clusters as ellipses of various sizes and shapes. These plots were helpful for seeing where single clusters appeared to be composed of multiple, distinct sub-clusters. When sub-clusters are visually identified in this way, it can be potentially beneficial to increase the number of clusters. Fig. 2 shows an example of the diagram generated for a 6-cluster solution on the clean data set using the k-means algorithm. Note that the appearance of overlapping clusters in the figure is due to the projection of the multidimensional data set onto a two dimensional view.

To gain further insights into the features that each cluster represented, a "feature plot" was generated for each cluster, in the form of a bar chart that shows the cluster's primary characteristics. These characteristics are determined by the highest-ranked cluster means as described in section 4.6. Fig. 3 gives an example of

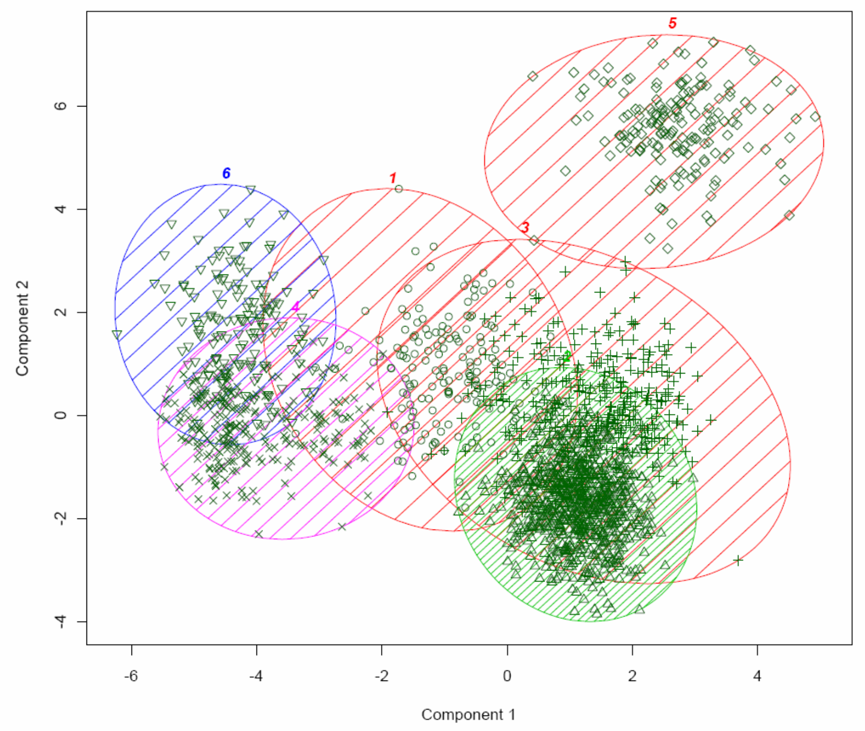

Fig. 2 Diagram of a clustering solution with 6 clusters generated using k-means

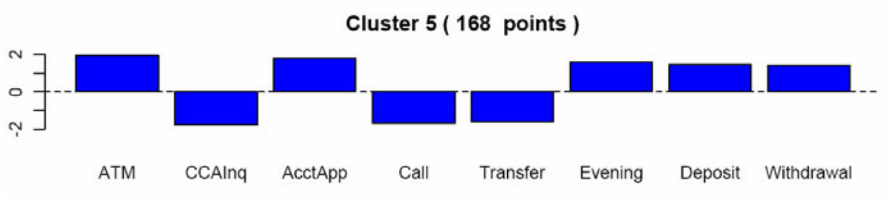

Fig. 3 Feature plot of a sample cluster selected from a 6-cluster solution generated using k-means

the feature plot for one cluster in a six-cluster solution generated using k-means on the clean data set. It shows the top eight features for this cluster are: heavy ATM usage, few CCA inquiries, many account applications, few call interactions, few fund transfers, heavy evening usage, and many deposit and withdrawal transactions. Similarly, Table 4 shows the top six features sorted in decreasing order of their absolute significance for all six clusters. The clusters' nicknames summarize their main characteristics.

Table 4 Top six features in decreasing order of absolute significance for a 6-cluster solution generated using k-means

ID	*Cluster Nickname*	*# of Cust.*	*Top six features in decreasing order of absolute significance*					
			1	*2*	*3*	*4*	*5*	*6*
1	Heavy SMS and Web users for various purposes	854	+LoanApp	+Transfer	+Interaction	+MarketInq	+SMS	+Web
2	Web and SMS users for CCA application	322	+CCAApp	+Web	+SMS	-AcctInq	-LoanApp	-Branch
3	Heavy branch users	252	+Duration	+Branch	+Interaction	-Deposit	-Web	-SMS
4	Infrequent marketing inquiry users via call-centre	174	+MarketInq	-Night	+Call	-Withdrawal	-Interaction	-AcctInq
5	Evening ATM users for account application	168	+ATM	-CCAInq	+AcctApp	-Call	-Transfer	+Evening
6	Weekday night users that prefer branch	163	-Weekends	+Night	-Evening	+Branch	-Web	+Duration

Table 4 shows that, as measured by the number of customers, the first cluster is extremely large compared to the other clusters. Also, interactions for all the purposes of loan applications, funds transfers, and marketing inquiries are the dominant features of this cluster. Based on these two observations, the number of clusters was increased to eight in an effort to partition the first cluster. However, the 8-cluster solution did not subdivide this cluster by interaction purpose, as expected. In fact, it divided the large cluster into two smaller clusters, one representing users who make many loan applications and the other representing users who make many funds transfers. In addition, it also extracted a group of users from the second largest cluster that represents heavy web and SMS users applying for credit card accounts, only at night. This implies that simply analyzing the cluster features is not a satisfactory method for determining the optimal number of clusters. However, the feature plots were useful for interpreting the meanings of the clusters and providing qualitative insights that supported the quantitative assessment methods.

The Hubert gamma, Dunn Index, Silhouette Coefficient (SC), Calinski-Harabasz Index (CHI), and Hartigan Index (HI) statistical measures were also computed for different numbers of clusters to help determine the optimal number of clusters for the crisp clustering algorithms. Fig. 4(a) shows the values for three of these indexes across different numbers of clusters. The plot shows that the Hubert gamma and SC index reach their maximum values with three clusters. However, the indexes also have large values with 4, 5, or 6 clusters. Inspecting the within-cluster sum of squares plot, the optimal number of clusters should be found at the knee point of the curve. The plot shown in Fig. 4(b) indicates that 3 or 4

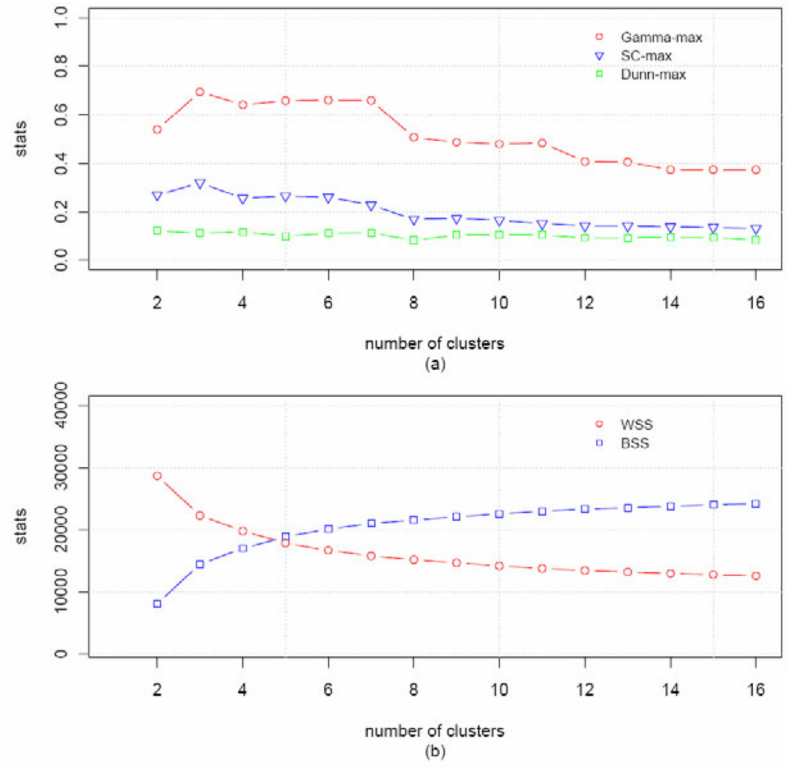

Fig. 4 Index values for solutions generated using k-means: (a) Common indexes; (b) Within-cluster and between-cluster sums of squares

clusters appear to be good knee points. Both the CHI and HI indexes imply that 6 or 7 clusters would be optimal, since the successive differences of both indexes are minimized there. The 3-cluster solution would produce relatively large and broad segments, which may not be useful within the business context. To obtain a finer-grained clustering result, the 6-cluster solution appears to be the next best choice.

For the fuzzy c-means algorithm, the partition coefficient (PC), partition entropy (PE), and Xie & Beni (XB) indexes were computed to help decide the optimal cluster number on the same data set. Fig. 5(a) shows the values of the PC and PE indexes on solutions with different number of clusters generated using the fuzzy c-means algorithm. The plot shows that 3 clusters appears to be the optimal solution, since the PC index is maximized and the PE index is minimized at that number. However, as with the crisp clustering, the 3-cluster solution would produce a very coarse-grained result. The 6-cluster solution appears to be the next best choice, as shown in Fig. 5(b).

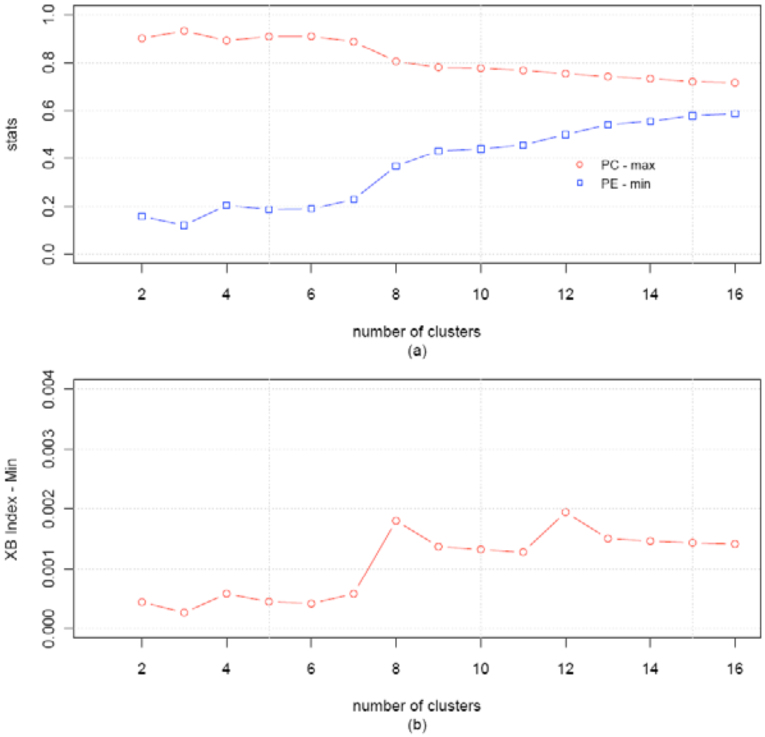

Fig. 5 Index values for solutions generated using fuzzy c-means: (a) Partition coefficient and partition; (b) Xie & Beni index

5.4 Validate Clustering Results

Various experiments were conducted to test the validity of the k-means, neural gas, genetic k-means, and fuzzy c-means clustering algorithms. First, the three crisp clustering algorithms and the fuzzy c-means algorithm were applied to both clean and noisy data sets to determine which algorithm was able to most efficiently generate stable and meaningful solutions. Second, the similarity of the clustering solutions was compared using the results from the crisp algorithms applied to multiple synthetic data sets that had same underlying structure and parameters. Ideally, since the synthetic data sets were constructed the same way, the features of the optimal clustering solutions generated for each of the different data sets should match one another. Third, the fuzzy c-means algorithm was applied to both clean and noisy data sets, and the stability of the solutions was analyzed. A detailed comparison of the k-means and the fuzzy c-means algorithm on both clean and noisy data sets is provided in Section 5.5.

To evaluate the stability of the three crisp algorithms, two trails for each algorithm were performed on the same data set and the results were compared using the Rand index metric discussed in Section 0. To avoid the problem of obtaining suboptimal clustering as a result of random initialization, the *k*-means algorithm was repeatedly run 3,000 times in each trial. In contrast, both neural gas and genetic k-means were run only once, but with a large number of iterations. The neural gas algorithm was run with the following learning rate parameters: 1,000 iterations and λ_i=10, λ_f=0.01, ε_i=0.5, ε_f=0.005. Increasing the number of iterations did not improve the stability of the results. For the genetic k-means algorithm, each clustering solution was represented by a vector of the cluster centres' coordinates, which has length 144 for a 6-cluster solution with 24 dimensions. The population size was defined as 100 and the mutation chance was chosen as 0.25%.

Table 5 shows the Rand index values for two 6-cluster solutions generated using each algorithm on several clean and noisy data sets. Repeated k-means produced the most stable performance across data sets with different levels of noise, followed by repeated fuzzy c-means. All the algorithms tended to become more volatile when the noise levels increased, especially the neural gas and genetic k-means algorithms. It was also observed that these algorithms' stability could be improved with repeated runs. However, the total execution time was much longer for neural gas and genetic k-means than k-means, even though they took fewer runs to reach a similar stability. For example, to generate a 6-cluster solution with 99% stability on the 15% noise data set, repeated k-means took less than 3 minutes to execute the Hartigan and Wong algorithm (Hartigan 1979) 3,000 times, whereas the neural gas algorithm took about 20 minutes and the genetic k-means algorithm took about 30 minutes to complete a single run. Given its overall stability under various conditions, repeated k-means was selected as the crisp clustering algorithm to be used as a baseline for compassion with the fuzzy c-means in the subsequent experiments.

Table 5 Rand index values on two runs of each algorithm for various data sets

Data Sets	*Clean*	*5% Noise*	*10% Noise*	*15% Noise*	*20% Noise*	*30% Noise*
Repeated k-means	1.000	0.998	0.982	0.991	0.969	0.994
Neural gas	0.997	0.930	0.922	0.821	0.707	0.735
Genetic k-means	0.883	0.845	0.828	0.769	0.698	0.717
Fuzzy c-means	0.998	0.995	0.983	0.986	0.966	0.991

Another observation was that there was significant variance in the clustering solutions obtained from the three algorithms, even at the 99% stability level. Table 6 shows a contingency table comparing two clustering solutions generated using k-means and neural gas algorithms on the clean data set. The table shows the number of points that fall in the same clusters between two clustering solutions. It was noticed that cluster 2 of neural gas was split into clusters 2 and 3 of k-means, while cluster 4 of k-means was split into cluster 3 and 5 of neural gas. This result was probably due to the fact that the algorithms can generate different suboptimal solutions that correspond to different local minima.

Table 6 Contingency table comparing two clustering solutions generated using k-means and neural gas algorithm

Cluster	*NG-1*	*NG-2*	*NG-3*	*NG-4*	*NG-5*	*NG-6*
k-means-1	170	0	3	1	1	0
k-means-2	0	248	0	0	0	0
k-means-4	0	0	331	0	523	0
k-means-6	0	0	0	310	10	0
k-means-3	0	162	0	0	0	1
k-means-5	0	0	0	0	0	167

To test whether the clustering procedure consistently identified clusters with the same features on different data sets, two different data sets were constructed with the same set of parameters were tested with the same process. To evaluate how close the two solutions were, a matching score, defined as the average percentage of matching features among the first six dominant features across all clusters, was computed. Table 7 shows the matching scores between the clustering solutions with different number of clusters on two different data sets. The matching scores between the clustering solutions using the selected set of variables show that more than 90% of features matched across all clusters for both k-means and neural gas. The reason for the low score with genetic k-means is largely due to the non-convergence of the algorithm in this case.

Table 7 Feature matching score between clustering solutions on different data sets

Algorithms	*Matching scores for different number of clusters*						
	3	*4*	*5*	*6*	*7*	*8*	*9*
Repeated k-means	1.000	0.958	1.000	1.000	0.929	1.000	0.833
Neural gas	1.000	0.958	0.967	1.000	0.912	1.000	0.907
Genetic k-means	1.000	0.958	0.733	0.750	0.690	0.792	0.704

To evaluate the stability of the fuzzy c-means algorithm, two runs of the algorithm were performed on both clean and noisy data sets. A contingency table was then produced that compared the number of points that fall into the same clusters by “hardening” the fuzzy clusters. Hardening the fuzzy clusters was achieved by uniquely assigning individual points to their closest cluster center. The Rand indexes for those solutions were calculated as 1.0 and 0.935, respectively.

5.5 Discussion and Analysis

As discussed in section 3.2, the synthetic data were generated from a population of eleven customer types. This allows a comparison matrix to be calculated from the derived clustering and the original data. Table 8 compares the result of running the k-means algorithm with six clusters on the clean data set with the original customer types. As can be seen, the original customer types, as defined in Table 8, largely fall cleanly into the generated clusters. The horizontal cluster IDs show how a six-cluster solution can be generated from the eleven customer types, and the vertical cluster IDs are those generated by the k-means algorithm. As there is no inherent meaning in a cluster ID, the rows have been reordered to show the correspondence between the two clustering more clearly.

Table 8. Contingency table comparing crisp 6-way clustering with customer types on clean data set

Customer Type	*RAN*	*RAF*	*RAP*	*SHN*	*SHW*	*SUN*	*SUW*	*WAD*	*WAF*	*WAP*	*WAU*
Cluster ID	*1*	*1*	*2*	*3*	*3*	*4*	*4*	*5*	*5*	*5*	*6*
2	167	82	5	0	0	0	0	0	0	0	0
3	0	0	162	0	0	0	0	0	0	0	1
5	0	0	0	67	100	0	0	0	0	0	0
6	0	0	0	0	0	130	80	0	5	5	0
4	0	0	0	0	0	1	3	112	578	160	0
1	0	2	0	0	0	3	1	0	1	2	166

Performing the same analysis while varying the number of clusters shows that all the statistically “good” clustering solutions, from three to seven clusters, correspond well to the original customer types. The three-cluster k-means solution distinguishes between high-school students, retired-age customers and approximately half of the unemployed adults, and the remaining population (university students and the other working-age adults). Moving to four clusters splits out the university students, and with five clusters the unemployed form a cluster on their own. The six-cluster solution shown above separates the retired-age customers who work part-time from the other retired-age customers, and with seven clusters, the university students are cleanly separated according to whether or not they have a job.

As the number of clusters increases, this clean separation would ideally continue until all of the eleven customer types form their own cluster. However, with eight clusters, the working population (employed or domestic) splits into two clusters, but not according to their customer type. As the number of clusters is increased further, the correspondence between customer types and clusters becomes still less clear. This demonstrates that there is a point at which the clustering algorithms become unable to distinguish between meaningful patterns in the data and random variations in customer behaviour. Although there are structural differences in the customers’ interaction patterns, the statistical overlap between the different customer behaviour patterns is too great to allow the clustering to differentiate the underlying groups, using the interaction behaviour metrics chosen for analysis.

Had the variables used for clustering been finer grained, further cluster segmentation may have been possible. For example, the variables used to measure the time when interactions occurred for cluster analysis were defined as four, six-hour periods. This coarse categorization was surprisingly effective, given the more subtle timing differences that characterised the underlying data different customer groups. Had the analysis used hourly variables instead, better segmentation results for larger numbers of clusters may have been achievable. Because analysis effort was designed independently from the synthetic data construction, the benefit of using finer-grained time periods was not obvious *a priori*.

Table 9 Rand index values for different numbers of k-means clusters compared to customer types

Artificial Clustering	*Number of clusters in k-means solution*							
	3	*4*	*5*	*6*	*7*	*8*	*10*	*12*
Same as k-means	0.862	0.872	0.964	0.966	0.945	0.669	0.590	0.525
11-cluster	0.645	0.799	0.852	0.874	0.884	0.881	0.900	0.900

Table 9 shows Rand index values comparing k-means clusterings to artificial clusterings based on the known customer type. Two types of artificial clusterings were generated: one by taking the same number of clusters as the k-means solution and assigning customer types to each cluster according to their weighting in the k-means solutions, and an 11-cluster solution with one cluster corresponding to each customer type. As can be seen, the Rand index values comparing the solutions to the same-sized artificial clustering drop sharply at eight and more clusters, meaning that the k-means clusterings do not closely correspond to the artificial clusterings. This agrees with the results of the visual inspection and statistical analysis obtained earlier to determine the optimal number of clusters. When comparing the k-means clusterings to the eleven predefined customer types, the higher numbers of clusters perform best, although significant agreement is reached from six clusters onwards.

Table 10 Comparison of fuzzy and crisp clusterings on clean data

First clustering	*Second clustering*	*Fuzzy Rand Index*
k-means	Known customer types	0.874
Fuzzy c-means	Known customer types	0.832
k-means	Fuzzy c-means	0.857

In order to compare the crisp and fuzzy clustering solutions, a fuzzy membership matrix was generated from the crisp clusterings, and the Fuzzy Rand Index was computed to compare the crisp, fuzzy, and customer type solutions on the clean data set.

Table 10 shows that the fuzzy c-means algorithm performs well on the clean data set, although not as well as the k-means algorithm. This is not surprising as the customer types constitute a crisp clustering, and the Fuzzy Rand index tends to give higher values when comparing crisp clusterings.

The effect of adding noise, as discussed in Section 3.3, to the data sets can be seen in Table 11. Both k-means and fuzzy c-means clusterings were computed, with six clusters, and then the clustering was compared to the 11-way known customer type clustering, by calculating the Fuzzy Rand index. Note that, while the Fuzzy Rand index gives the same result as the Rand index when comparing crisp clusterings, it always produces smaller values when applied to fuzzy clusterings. For example, whereas two identical crisp solutions always have a Fuzzy Rand Index of 1.0, using the Fuzzy Rand index to compare two fuzzy clusterings will always produce a value less than 1, even when they are identical.

Table 11 Fuzzy Rand index values for 6-cluster solutions compared with known customer types

Data set	*k-means*	*Fuzzy c-means*
Clean	0.874	0.832
5% noise	0.833	0.776
20% noise	0.783	0.761

Both crisp and fuzzy algorithms show some degradation in the presence of noise at the 5% level, and this effect is significantly more pronounced at the 20% level.

For the data set that included transitional customers, the crisp clustering represented the transitional customers as being split between the clusters corresponding to working and unemployed customers. In the fuzzy clustering, however, they had a fractional membership in each cluster. The fuzzy representation is more meaningful in this case and corresponds better to the actual business scenario. If a number of customers spend half of the time employed and half unemployed, it is more accurate to describe them all as having partial membership of both groups, rather than to arbitrarily categorise some of them as employed and others as unemployed. In order to compare clustering of data including transitional customers, a fuzzy 11-cluster solution was generated from the known customer types, allocating the transitional customers 50% weight in each of the WAF and WAU groups. Moreover, fuzzy clustering would also be expected to quickly detect customer transitions; small shifts in cluster weightings would become apparent soon after the transition occurred and then continue to increase over time.

Table 12 Comparison of fuzzy and crisp clusterings on transitional customers

First clustering	*Second clustering*	*Fuzzy Rand Index*
k-means	Known customer types	0.842
Fuzzy c-means	Known customer types	0.823
k-means	Fuzzy c-means	0. 917

Table 12 shows that both crisp and fuzzy clusterings produce very similar solutions. While the k-means has a slightly better Fuzzy Rand index score, this metric does not reflect the benefit of the fuzzy solution to describe partial membership across multiple clusters.

6 Conclusion

In summary, both crisp and fuzzy clustering algorithms appear to be useful for extracting meaningful groupings from customer interaction information. For the given data sets, the customers were accurately segmented according to their underlying types using common characteristics related to the time, channel, and transaction type of their interactions. These results suggest the techniques could be applied to customer channel interaction data with unknown characteristics with the expectation of drawing substantive conclusions about customer groupings based on their interaction behaviour patterns. For marketing purposes, these groupings – related to communication and lifestyle preferences – could be used instead of existing demographic and transactional-based customer segmentation models. Alternatively, they could be used in conjunction with existing segmentation models to provide new criteria for subdividing existing segments.

Fuzzy clustering was found to be superior for accurately describing customers whose underlying group membership was in flux. Hence, this technique may be better suited for applications where the customers migrate between groups. Using soft competitive learning and genetic algorithms to generate crisp clusters did not, in this particular case, appear to improve the efficiency of the clustering process. However, the soft competitive learning algorithm produced meaningful clustering results that were fairly close to the k-means results.

There are a number of potential business applications that could benefit from these techniques. The first, and most obvious, is to supplement existing customer marketing segmentation models with information derived from customer interaction information. Segments derived from demographic and transaction-based models could be compared with interaction-based clusters to gain additional insights into customer groupings. For example, whereas traditional segmentation tools might group all college students into one segment, clustering based on interaction behaviour could be used to further divide the group into sub-segments of “scholars”, “socialites” and “video gamers”. Specific products and services could then be marketed more effectively to each of these sub-groups.

Another potential application is to market products and services to customers solely based on the customers’ interaction behaviour groupings, ignoring any demographic disparities. For example, if a cluster analysis shows that there is a subset of retirees who behave in a way that is similar to university students, it could be beneficial to treat them as university students from a marketing perspective, even though they do not share the same demographic profile. The key point here is that that the demographic data may be insufficient to fully understand the desires and needs of some customers. Analysis of customer interaction information can provide an alternative angle from which to view customers and their behaviour.

Since fresh customer interaction data are generated continuously, it may be beneficial to perform cluster analysis on a regular basis. In particular, migration of customers between clusters over time could provide companies with a data-driven way of detecting new marketing opportunities. For example, if a customer's interaction pattern shifts from an "employed" cluster to a "retired" cluster, different products or services could be marketed towards them, accordingly. If they migrated to an "unemployed" cluster, collection of payments due could be pursued more aggressively. Alternatively, if cluster analysis showed that certain customers were moving to a cluster that was highly correlated with near-term customer attrition, retention efforts could be increased for those customers. Overall, fuzzy clustering techniques show the greatest potential for being able to identify the movement of customers between clusters.

While the results of this research are quite promising, further validation using real-world data sets is required. Real-world data are expected to differ from the synthetic data in two important ways. First, because companies may not be able to capture or easily aggregate multi-channel customer interaction data, fewer variables may be available for analysis. This consideration would have the most impact on the variable selection and data validation parts of the analysis process, and may affect the accuracy of the results. Second, the data variability is expected to be larger than was simulated in the synthetic data set. Higher variability could potentially lead to greater difficulty in identifying the optimal number of clusters and increased overlap between clusters where fuzzy clustering methods. Hence, interpreting the clustering results could become more challenging with real-world data.

Besides applying these techniques to live customer data, further investigation is also warranted in several areas. First, it would be beneficial to understand the sensitivity of the different clustering techniques to data set size, in order to understand the minimum amount of data required to achieve reasonably accurate results. Second, determining the effects of increasing the channel metrics' granularity – particularly time increments – would be of interest. Finally, determining the advantage provided by taking into account additional interaction details, such as the phone number of inbound calls or the Internet protocol (IP) address of the customer for web sessions, could also increase the strength of this approach.

References

Allred, C., Hite, K., Fonzone, S., Greenspan, J., Larew, J.: Modeling and data analysis in the credit card industry: bankruptcy, fraud, and collections. In: IEEE Systems and Information Design Symposium (2002)

Balakrishnan, P.V., Cooper, M.C., Jacob, V.S., Lewis, P.A.: A study of the classification capabilities of neural networks using unsupervised learning - A comparison with K-means clustering. Psychometrika 59(4), 509–525 (1994)

Balakrishnan, P.V., Cooper, M.C., Jacob, V.S., Lewis, P.A.: Comparative performance of the FSCL neural net and K-means algorithm for market segmentation. European Journal of Operational Research 93(2), 346–357 (1996)

Bezdek, J.C.: Cluster validity with fuzzy set. Journal of Cybernet 3, 58–72 (1974)

Calinski, T., Harabasz, J.: A dendrite method for cluster analysis. Communication in statistics 3, 1–27 (1974)

Campbell, D., Frei, F.: The persistence of customer profitability: empirical evidence and implications from a financial services firm. Journal of Service Research 7(2), 107–123 (2004)

Campello, R.J.G.B.: A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment. Pattern Recognition Letters 7(28), 833–841 (2007)

Dunn, J.C.: Well separated clusters and optimal fuzzy partitions. Journal of Cybernetica 4, 95–104 (1974)

Edelman, D.B.: An application of cluster analysis in credit control. IMA Journal of Mathematics Applied in Business and Industry 4, 81–87 (1992)

Fritzke, B. (1997), Some competitive learning methods, `http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/` (accessed July 24, 2009)

Goldberg, R.: Proc. Factor: How to interpret the output of a real-world example. In: SESUG 1997 (1997)

Gordon, A.D.: Classification, 2nd edn. Chapman and Hall, Boca Raton (1999)

Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. Journal of Intelligent Information Systems 17, 107–145 (2001)

Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Cluster validity methods. SIGMOD 31, 40–45 (2002)

Han, J.W., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann Publishers, San Francisco (2001)

Hartigan, J.A.: Clustering algorithms. Wiley, New York (1975)

Hartigan, J.A., Wong, M.A.: A K-means clustering algorithm. Applied Statistics 28, 100–108 (1979)

Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A., De Carvalho, A.C.P.L.F.: A survey of evolutionary algorithms for clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 39(2), 133–155 (2009)

Hitt, L., Frei, F.: Do better customers utilize electronic distribution channels? The case of PC banking. Management Science 48(6), 732–748 (2002)

Kaufman, L., Rousseeuw, P.J.: Finding groups in data. In: An Introduction to cluster analysis. Wiley, New York (1990)

Kohonen, T.: Self-organizing maps. Springer, New York (2001)

MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proc. of the Fifth Berkeley Symposium on Math., pp. 281–297 (1967)

Martinetz, T., Berkovich, S., Schulten, K.: ‘Neural-Gas’ network for vector quantization and its application to time-series prediction. IEEE Transactions on Neural Networks 4(4), 558–569 (1993)

Nargundkar, S., Olzer, T.J.: An application of cluster analysis in the financial services industry. Case Study (2000)

Nikhil, R.P., James, C.B., Richard, J.H.: Sequential competitive learning and the fuzzy c-means clustering algorithms. Neural Networks 9(5), 787–796 (1996)

Peng, Y., Kou, G., Shi, Y., Chen, Z.: Improving clustering analysis for credit card accounts classification. In: International Conference on Computational Science, pp. 548–553 (2005)

Pison, G., Struyf, A., Rousseeuw, P.J.: Displaying a clustering with CLUSPLOT. Computational Statistics & Data Analysis 30(4), 381–392 (1999)

Rho, J.J., Moon, B.J., Kim, Y.J., Yang, D.H.: Internet customer segmentation using web log data. Journal of Business & Economics Research 2(11), 59–74 (2004)

Sinisalo, J., Salo, J., Karjaluoto, H., Leppaniemi, M.: Mobile customer relationship management: underlying issues and challenges. Business Process Management Journal 13(6), 771–787 (2007)

Tan, P.N., Steinbach, M., Kumar, V.: Introduction to data mining. Addison Wesley, Reading (2005)

Xie, X.L., Beni, G.: A validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell. 13(8), 841–847 (1991)

Zakrzewska, D.: On integrating unsupervised and supervised classification for credit risk evaluation. Information Technology and Control 36(1A), 98–102 (2007)