A fuzzy prediction model for calling communities

Keivan Kianmehr* and Reda Alhajj

Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
and
Department of Computer Science
Global University
Beirut, Lebanon
Fax: +403–284–4707
E-mail: mkkian@ucalgary.ca
E-mail: alhajj@ucalgary.ca
*Corresponding author

Abstract: The analysis of logs related to social communities has recently received considerable attention for its importance in shedding light on social concerns by identifying different groups, and hence helps in resolving issues like predicting terrorist groups. In general, identifying calling communities can be used to determine a particular customer's value according to the general pattern of behaviour of the community that the customer belongs to; this helps in creating an effective targeted marketing design, which is significantly important for increasing profitability. In the telecommunications industry, machine-learning techniques have been applied to the Call Detail Record (CDR) for predicting customer behaviour such as churn prediction. In this paper, we pursue the identification of the calling communities and demonstrate how cluster analysis can be used to effectively identify communities using information derived from the CDR data. We use the information extracted from the cluster analysis to identify customer calling patterns. Customer calling patterns are then input to a classification algorithm to generate a classifier model for predicting the calling communities of a customer. We apply two different classification methods: the Support Vector Machine (SVM) algorithm and the fuzzy genetic classifier. The latter method is used for possibly assigning a customer to different classes with different degrees of membership. The reported test results demonstrate the applicability and effectiveness of the proposed approach.

Keywords: social communities; classification; clustering; customer behaviour; support vector machine; SVM; fuzzy genetic classifier.

Reference to this paper should be made as follows: Kianmehr, K. and Alhajj, R. (2011) 'A fuzzy prediction model for calling communities', *Int. J. Networking and Virtual Organisations*, Vol. 8, Nos. 1/2, pp.75–97.

Biographical notes: Keivan Kianmehr received his BSc degree in Software Engineering from IUST, Tehran, Iran. After spending over five years in the computer industry as a Software Developer and Project Leader, Kianmehr joined the graduate programme in the Department of Computer Science at the University of Calgary, where he completed his MSc degree in 2006. Currently, he is a PhD candidate under the supervision of Professor Reda Alhajj. He has published over 30 papers in prestigious refereed venues.

Copyright © 2011 Inderscience Enterprises Ltd.

He received teaching excellence awards at the University of Calgary and several prestigious scholarships including NSERC CSD and iCore. He is a successful mentor for undergraduate and graduate students. His research focuses on machine learning techniques, social networks, XML, data mining methodologies and bioinformatics.

Reda Alhajj is a Professor in the Department of Computer Engineering at the University of Calgary, Alberta, Canada. He has published over 280 papers. He served on the programme committee of several international conferences including IEEE ICDE, IEEE ICDM, IEEE IAT, SIAM DM, *etc*. He also served as guest editor of several special issues and is currently the Programme Chair of IEEE IRI 2009, CaSoN 2009, ASONAM, ISCIS, MS and OSINT-WM. He is on the editorial board of several journals and Associate Editor of *IEEE SMC – Part C*. Dr. Alhajj has a research group of ten PhD and eight MSc students working primarily in the areas of biocomputing and biodata analysis, data mining, multiagent systems, schema integration and reengineering, social networks and XML. He received the Outstanding Achievements in Supervision Award from the Faculty of Graduate Studies at the University of Calgary.

1 Introduction

Identifying social communities is an emerging research area that has already attracted the attention of several research groups. The main theme is to analyse logs that reflect social communication between different parties. The analysis leads to valuable discoveries that may have essential social and economical impact. From social perspective, the discoveries may highlight terrorist groups, family relationships, friendship or it may reveal more insight into the communities such as the development of trust by people participating in Virtual Communities (Daneshgar and Ho, 2008; Striukova and Rayna, 2008; Okkonen, 2004). From economical perspective, the analysis may lead to target certain customer groups. We concentrate on the latter perspective in this paper. In particular, we investigate customer relationships by analysing Call Detail Records (CDRs) obtained from a telecommunication company. Identifying terrorist groups have been previously studied by researchers, *e.g.*, Nasrullah and Larsen (2006).

Communities are defined as the collection of nodes (individuals) within which the links are dense but among which the links are sparse (Girvan and Newman, 2002). In the literature, many algorithms have been developed to discover network communities. They can generally be divided into three main categories:

1 graph theoretic methods such as random walk methods and physics-based methods (Pons and Latapy, 2005; Palla *et al*., 2005; Kim and Jeong, 2005), the spectral methods (Fiedler, 1973; Pothen *et al*., 1990; Fiedler, 1975; Shi and Malik, 2000), and the Kernighan-Lin algorithm (Kernighan and Lin, 1970)

2 hierarchical methods such as agglomerative methods based on the structural similarity metrics (Scott, 1991; Burt, 1976; Wasserman and Faust, 1994) and divisive methods based on betweenness metrics, like the Tyler algorithm (Tyler *et al*., 2003), and the Radicchi algorithm (Radicchi *et al*., 2004)

3 methods for detecting hyperlink-based web communities such as the Maximum Flow Communities (MFC) algorithm (Flake *et al*., 2002), the Hyperlink Induced Topic Search (HITS) algorithm (Kleinberg, 1999), the Spreading Activation Energy (SAE) algorithm (Pirolli, 1996), and others (Kumar *et al*., 1999; Chakrabarti *et al*., 1999).

FEC (Yang *et al*., 2007) is a new algorithm, recently proposed, to mine signed social networks where both positive within-group relations and negative between-group relations are dense. FEC considers both the sign and the density of relations as the clustering attributes, making it effective for not only signed networks but also conventional social networks including only positive relations. Also, FEC adopts an agent-based heuristic that makes the algorithm efficient and capable of giving nearly optimal solutions. FEC depends on only one parameter whose value can easily be set and requires no prior knowledge on hidden community structures.

More recent works (Berger-Wolf and Saia, 2006; Tantipathananandh *et al*., 2007) have concentrated on community identification in dynamic social networks considering the fact that social networks tend to change over time. The framework proposed in Berger-Wolf and Saia (2006) for identifying communities in dynamic social networks makes explicit use of temporal changes. Most communities tend to evolve gradually over time, as opposed to assembling or disbanding spontaneously. Whenever information about events in the social network is available, this temporal information is desirable to use in order to identify both communities with high intra-community similarity and also observe their persistence and development over time.

In Tantipathananandh *et al*. (2007) the assumption is that time is discrete, and in each time step, social interactions are observed in the form of several complete subgraphs of individuals (not every individual needs to be observed in each time step). Based on these observed groupings, true underlying communities and their developments over time are identified so that most of the observed interactions can be explained by the inferred community structure. The problem is posed as a combinatorial optimisation problem, based on the observation that individuals tend to first not change their 'home community' too frequently, and second tend to interact with the home community most of the time. After formulating the problem as an optimisation problem, authors prove that finding the most explanatory community structure is NP-hard and APX-hard, and propose algorithms based on dynamic programming, exhaustive search, maximum matching, and greedy heuristics.

The problem that has been tackled in our work is different from the beforehand mentioned works in the direction that they consider full degree of membership for the members of a community. However, we argue that some members of a community may partially belong to a community while they might also belong to another community.

1.1 Our contribution

In telecommunication, a CDR is a record containing information about recent system usage, such as the identities of sources (points of origin), the identities of destinations (endpoints), the duration of each call, the amount billed for each call, the total usage time in the billing period, the total free time remaining in the billing period, and the running total charged during the billing period. The format of the CDR varies among telecom providers and call-logging software.

In recent years, we witness dramatic increase in the competition among telecommunication companies in order to detain their current customers and acquire new ones. For this reason, the ability to dynamically classify and predict customers' behaviours according to their calling patterns obtained from CDR data has attracted considerable attention in the research community; it is beneficial for Yan *et al*. (2005):

1 Churn prediction: The goal is to understand when and why company's customers are likely to leave so that appropriate action can be planned. Customers become 'churners' when they discontinue their subscription and move their business to a competitor company. This has been developed in the telecommunication industry using data mining techniques. Data mining is applied in this area to perform two major tasks:

   - predict whether a particular customer will churn and when this will happen
   - understand why particular customers churn.

   By predicting which customers are likely to churn, the telecommunication company can reduce the rate of churn by offering the customers new incentives to stay.

2 Identifying calling communities: this can be used for determining a particular customer's value according to the general pattern behaviour of the community that the particular customer belongs to. This helps the effective targeted marketing design which is significantly important for increasing profitability in the telecommunication industry.

In Kianmehr and Alhajj (2008), we have proposed a framework in which an unsupervised machine learning technique, namely clustering is used to classify customers of a mobile service provider into appropriate calling communities according to the statistics extracted from the CDR data. Clustering technique is one of the most prominent approaches for identifying unknown classes amongst a group of objects, and has been used as a tool in many fields such as biology, image analysis, finance, *etc*. The classification algorithms evaluated in this paper use an unsupervised learning mechanism, wherein unlabelled training data is grouped based on similarity. Once an acceptable clustering has been found using the similarities and dissimilarities in the training data set, the clustering is transformed into a classifier by using a classification technique. In the proposed framework in Kianmehr and Alhajj (2008), the clusters are labelled, and a new object is classified with the label of the cluster which it is most similar to. For building the classifier model, a supervised Machine Learning (ML) techniques is employed to automatically classify and to identify customers' communities based on characteristics of customers extracted from clusters. We have employed different machine learning techniques to build classifier models and compared them in terms of classification accuracy and computational performance. However, the type of the classifier that is to be used to identify calling communities and customers' value is very crucial. For instance, in a marketing campaign, selecting inappropriate customers is very costly for verifying the impact of the marketing campaign when it is intended to modify the strategy toward the target group. For this reason, in this study we have decided to study the possible application of a fuzzy genetic algorithm in order to approach the customer classification problem in a fuzzy manner.

The fuzzy classification offers more convenience for selecting customer subgroups and for measuring the efficiency and validity of the communities regarding the classification goal Werro *et al*. (2006); for example, the marketing campaign design. The flexibility of fuzzy approach featured by the application of membership functions provides the ability to increase or decrease the homogeneity between the targeted customers depending on whether the proposed products are very specific or intended for a large community.

In this study, the agglomerative hierarchical clustering approach has been applied for the clustering task. It can produce an ordering of the objects (cluster tree), which may be more informative for the nature of the data being used here. For classification task, two different techniques have been selected. The Support Vector Machine (SVM) as a statistical-based learning approach has been used to build the classifier model. The main reason that we have employed SVM is that it has shown to be very effective in terms of performance and running time among other investigated machine learning techniques when applied for customer community prediction, in our earlier work in Kianmehr and Alhajj (2009). A fuzzy genetic algorithm has been also applied for the classification task. The SVM algorithm has been employed to identify crisp user communities and fuzzy genetic algorithm has been adapted for assigning a particular customer to possibly more than one community with different degrees of membership. This way, we differentiate between strict membership and gradual membership. Fuzziness is attractive because it facilitates the possibility of having partial membership in a given group. Each fuzzy set has a corresponding membership function which is used to decide for each customer its degree of membership in the group.

The rest of this paper is organised as follows. Section 2 defines CDR data and introduces calling neighbours. Section 3 presents details of the proposed method for automated identification of calling communities. Section 4 reports experimental results. We discuss the possible fuzzy classification of customers' communities using our proposed method in Section 5. Finally, we conclude this paper with a summary of the proposed framework in Section 6.

2 CDR data and calling neighbours

Because of the confidential nature of the customers' information and to preserve privacy, the information provided by telecommunication companies is very limited for modelling except the CDR data. From the CDR data, it is possible to extract customer's calling destination numbers, duration and frequency for each destination number for a particular period of time. Customer's calling neighbours can be also identified by using the links of who calls whom Yan *et al*. (2005). There are two types of calling neighbours:

1 Direct calling neighbour: A person who calls the customer or whom the customer calls. The majority of these neighbours of a customer may be outside of the service provider's own network, and no information about them is available. An example of direct calling neighbour can be described as follows: members of a research community call each other heavily.

2 Indirect calling neighbour: A person who calls the same number(s) as another customer does. For example, employees of a large organisation such as a bank who work in different local branches may call their headquarter frequently, and each

employee is possibly a calling neighbour of other employees. Even though employees of different branches may not know each other, they can be classified into the community of colleagues.

To measure the closeness of the first type of calling neighbours, the calling frequency or duration has been used. For the second type of neighbours, a distance measure, which will be presented in more detail in Section 3.2, has been applied to quantify the closeness of a neighbour. Discovering calling neighbours will result in forming calling communities. However, there are two major challenges in using the CDR data for finding the calling communities:

1 The majority of the destination phone numbers are outside of the service provider's network. Therefore, information about these customers such as their calling records and links between them is not available.

2 The other type of phone numbers are those in the service provider's network, and each of them corresponds to a customer of the service provider. The connectivity between these customers is very sparse because customers more likely call numbers outside their home network.

After identifying the customers communities, the information derived from the calling communities can be used for building a classifier model. This classifier model is able to assign a new customer to one community or possibly more than one existing communities according to his/her general calling pattern extracted from the CDR data as well as closeness of his/her direct and indirect calling neighbours. In terms of market campaign design, the characteristics of communities can be used for increasing the profitability of the service provider company by targeting appropriate communities and customers. Intuitively, people in the same community have more likely similar behaviours because of their influences on the community. For instance, if a customer more frequently calls member of a community in which members generally accept service promotion offers, that particular customer can be considered for targeted service promotions.

3 Model details

In this section, the classifier model that has been developed for the course of this study, will be described in more detail. The system consists of three major phases: (1) data preprocessing, (2) clustering, and (3) classification.

3.1 Data preprocessing

The CDR data used in this work was given by a telecommunication company providing wireless services. Because of the confidential nature of the data and business rules, the CDR data for a short period of time is given. However, the results show that the classifier model built based on the information extracted from calling links within the given CDR data perform classification at a reasonable accuracy.

During the first step of the data preprocessing, all the phone numbers (customers) within the service provider's network have been identified. Since the CDR data includes information about the customers, all the source phone numbers have been considered as the subscribers of the service provider. The given data set consists of 55 000 calling records of 2000 subscribers (distinct phone numbers within the service provider's network). Calls with very low duration (less than 5 sec) are assumed to have no effect on identifying the subscriber's neighbours and are ignored.

In the second step, for each subscriber, his/her own phone number, called destination numbers (within and outside of the home network), and each destination number's calling duration and frequency in that particular period of time are extracted from the CDR data. During pre-processing of the CDR data, inactive customers have been excluded from the data since these numbers greatly skewed the distance distribution. Inactive customer refers to a customer who barely makes a phone call within a particular time period.

3.2 Clustering of customers

Clustering refers to the process of partitioning a set of data points into a set of meaningful sub-classes, called clusters. It helps users to understand the structure in a data set. Clustering is very useful in explanatory pattern-analysis, grouping, decision making, and machine-learning situations, including document retrieval, image segmentation, pattern classification, *etc.* Clustering is unsupervised learning, which means that we do not know how many classes there are and what each class properties are; and it does not depend on any training examples. Rather, clusters can be created statistically or by using neural and symbolic unsupervised induction methods. The common clustering techniques are hierarchical algorithms, partitional clustering, and density-based clustering, among others. The main application of clustering task in this work is to automatically identify the calling communities. Every clustering algorithm requires a similarity (distance) measure to perform the cluster analysis.

3.2.1 Distance measures

In order to identify the closeness of a particular customer to his/her direct calling neighbours, a similarity measure weighted by call duration or frequency between two phone numbers has been used. The weighted similarity measure is a first order distance Yan *et al*. (2005) defined as follows:

$$D_1(i,j) = \frac{1}{w_{i,j}} \tag{1}$$

where $w_{i,j}$ is the call duration or frequency between customers (or phone numbers) i and j.

As described earlier, the direct calling neighbours of customers within the service provider's network are very sparse because most of the phone numbers are outside the service provider's network. Therefore, it is needed to include another distance measure, namely the second order distance (Yan *et al*., 2005), which reveals relationships between customers according to their indirect calling patterns where two customers have common direct calling neighbours. In order to define the second order distance between a pair of customers i and j, the set of phone numbers customers i and j called, respectively, in the specified time period, are weighted by calling duration or frequency:

$$N_w(i) = \{w_i(1), w_i(2), \ldots, w_i(n_i)\} \tag{2}$$

where n_i is the total number of distinct phone numbers customer i called during the specified time period. Here, we normalise the value of calling duration or frequency, $w_i(k)$, such that $\sum_{k=1}^{n_i} w_i(k) = 1.$ Finally, the second order distance between two customers i and j is defined as follows:

$$D_2(i,j) = 1 - \frac{1}{2} \sum_{x \in N(i) \cap N(j)} w_i(k_i(x)) + w_j(k_j(x)) \tag{3}$$

where x is a common phone number which both customers i and j called during the specified time period, and $k_i(x)$ and $k_j(x)$ are the corresponding weights of common called phone number x in $N_w(i)$ and $N_w(j)$, respectively. According to Equation (3), two customers are considered very close when they call some common numbers frequently (or heavily), regardless of how many other numbers they regularly call. The distance between a pair of customers is 1 (maximum possible distance) if they do not call any common phone number.

3.2.2 Clustering technique

Using the first and second order distance measures, a hierarchical clustering approach which works based on links between customers, can be applied to identify communities. In order to incorporate both the first and second order distance measures into the similarity measure of the clustering algorithm, the following distance measure has been defined:

$$D(i,j) = (1-\alpha)D_1(i,j) + \alpha D_2(i,j). \tag{4}$$

The term α controls the degree of relevance of the first and second order distance measures and typically depends on the network characteristics of the mobile service provider. As $\alpha \to 0$, the similarity measure approaches the first order distance measure. Intuitively, α captures the customers indirect calling neighbours.

Equation (4) makes it possible for the clustering algorithm to merge clusters with the most number of links, which are defined as the common neighbours of the customers based on both direct and indirect calling patterns. That is, the clustering algorithm identifies communities the members of which have short first order distance, but large second order distance between each other, as well as communities whose members have large first order distance but short second order distance between each other.

For discovering calling communities in this study, the agglomerative hierarchical clustering algorithm has been used. It is a bottom up clustering approach that investigates grouping in the given data by creating a cluster tree according to a particular distance measure.[1] The algorithm starts with the disjoint clustering, which places each the n objects in an individual cluster. The clustering algorithm being employed imposes the way that the proximity matrix should be interpreted to merge two or more of these trivial clusters, thus nesting the trivial clustering into a second partition. The process is repeated to form a sequence of nested clustering in which the number of clusters decreases as the sequence progresses until a single cluster containing all n objects remains. The output is a tree that represents a multi-level hierarchy, where clusters at one level are grouped together to form clusters at the next higher level. This allows the user of the system to decide what level or scale of clustering is most appropriate for customers of a mobile service provider. The MATLAB Statistics Toolbox has been used for conducting the hierarchical clustering. The basic procedure to perform hierarchical clustering in our model is as follows:

1 Find the similarity and dissimilarity (including both first and second order distances) between every pair of customers in the CDR data: In this step, we calculate the distance between pair of objects using our similarity function defined in Equation (4).

2 Create a cluster tree which groups the customers into a binary hierarchical tree: In this step, pairs of customers that are in close proximity are linked together using a linkage function. The proximity of customers to each other is measured using the distance information generated in Step 1. A hierarchical tree is then formed as newly formed binary clusters are recursively grouped into larger clusters.

3 Decide on the level to divide the hierarchical tree into clusters: In this step, we divide the customers in the hierarchical tree into clusters using a cluster function. Our cluster function can create clusters by detecting natural groupings in the hierarchical tree by considering the accuracy of a classifier which, is built based on the obtained clusters.

3.3 Building the classifier

Classification is a technique used for prediction, which is one of the most attractive aspects of data mining. It is simply the process of building a classifier model based on some known objects and predefined classes. The task involves two major steps. First, exploring through data objects (in the training set) to find a set of classification rules which determine the class of each object according to its attributes. Second, building a classifier based on the extracted rules to predict the class or missing attribute value of unseen objects. The application of classification task in our study can be elaborated as follows:

After identifying user communities, it is time to transform the clusters into a classifier model which is able to predict a community where a new customer belongs to according to his/her calling patterns. We use the community of a customer as his/her class label and then we derive several features using the information extracted from calling links and clusters. Finally, we create a training set in which every column represents a distance feature and every row represents a particular customer's values for the features.

3.3.1 Distance features

The following input features have been constructed for each customer from his/her calling neighbours based on the first order and second order distances. The features are defined as follows:

- Total number of customer's direct calling neighbours: this feature describes if a particular customer is socially connected to many other customers through direct links.

- Percentage of a customer's calls made to his/her closest direct calling neighbour: this feature shows if a customer is a socialise member of a network by not making most of the phone calls to his/her closets direct neighbour.
- Percentage of a customer's direct calling neighbours which are within the service provider's network: this feature describes how likely a user is to call customers inside his/her home network, in other words, it shows if a particular customer has many social conductivities within the home service network.
- Percentage of a customer's direct calling neighbours which are outside of the service provider's network: this feature describes how likely a user is to call customers outside his/her home network, in other words, it shows if a particular customer has many social conductivities outside the home service network.
- The shortest distances of a customer to all existing classes: this feature represents the closeness of a particular customer to all existing classes.
- Percentage of direct calls to neighbours (within the network) belonging to all existing classes: this feature measures the social connections of a particular customer to different classes (communities) in terms of direct calling neighbours.
- Percentage of indirect calls to neighbours (within the network) belonging to all existing classes: this feature quantifies the social connections of a particular customer to different classes (communities) in regards to indirect calling neighbours.

For every customer within the service provider's network, a feature vector based on the above feature definitions is built. This feature vector represents relationships between this particular customer and all other customers within the existing communities. Then, a set of feature vectors, each of which corresponds to a specific customer, is used as the training set for the classification algorithm.

3.3.2 Classification technique

For building a classifier model that satisfies the purpose of this paper, two different approaches have been used. The first approach is SVM from the family of statistical-based learning algorithms.

SVM are known as a powerful technique for classification problems (Vapnik, 1998; Cristianini and Taylor, 2000). The goal of SVM is to construct a separating hyperplane that is maximally distant from different classes of the training data (Optimal Separating Hyperplane). Basically, SVM can be used to solve binary classification problems. However, in order to use SVM for real world classification tasks, the idea has been extended for multi-class problems as well. The extension can be done either during the learning process or during the decision process. One-vs-rest and adaptive code algorithm are two of the most well-known extensions of SVM to multi-class problems (Allwein *et al*., 2000; Ratsch *et al*., 2002). The formal definition of linear SVM can be described as follows (Vapnik, 1998; Cristianini and Taylor, 2000). Let the training data of two separable classes with n samples be represented by $(\vec{x}_1, \vec{y}_1), (\vec{x}_2, \vec{y}_2), \ldots, (\vec{x}_n, \vec{y}_n)$, $i = 1, 2, \ldots, n$ where $\vec{x} \in R^N$ is an N dimensional space (feature vector) and $y_i \in \{-1, +1\}$ is the class label, maximise:

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\vec{x}_i^T \cdot \vec{x}_j), \tag{5}$$

under the constraints $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $\alpha_i \geq 0, i = 1, 2, \ldots, n.$ The optimal separating hyperplane can be found from the solutions α_is to this maximisation problem. A soft-margin algorithm as an extension of the basic algorithm is available, when the instances are not linearly separable. For many real world classification problems, SVM has been found to be more effective and faster than other machine learning methods such as neural networks. However, SVM suffers from understandability and interpretability.

The second approach that has been applied for classification task is a fuzzy genetic rule-based classification technique (Ishibuchi *et al.*, 1992). In contrast to SVM, this rule-based approach is more understandable by humans, but suffers from efficiency issues. In this work, the characteristics of the fuzzy-rule based approach have been used for possibly doing fuzzy classification.

For conducting the classification based on SVM, a MATLAB interface of LIB-SVM (Chang and Lin, 2008) has been used. LIBSVM is a free library for SVM classification and regression. The fuzzy genetic rule-based algorithm has been implemented in MATLAB. The MATLAB genetic algorithm built-in functions have been integrated into the implementation. In the rest of this section, we first present some basics of the fuzzy set theory and genetic algorithm required to understand the rest of this paper, then we describe the fuzzy genetic rule-based.

Overview of fuzzy set theory

A fuzzy set has a continuum of grades of membership (Zadeh, 1965). There is a gradual transition from 'belonging to a set' to 'not belonging to a set' (Jang and Sun, 1995). Fuzziness does not come from the randomness of members of the set, but from the uncertain and/or imprecise nature of abstract concepts (Jang and Sun, 1995).

The construction of a fuzzy set F relies first on identifying a domain of values that could belong to the set, generally referred to as the *Universe of Discourse* (Jang and Sun, 1995). The set is characterised by a membership function, denoted $\mu_F(x)$ that associates each value in the Universe of Discourse to a real number on the unit interval [0,1]. The specification of the membership function is completely subjective to the person who defines it. For example, a temperature that one person considers to be hot, may be considered cold by another individual. In general, a fuzzy set F with universe of discourse X is defined as a set of ordered pairs:

$$F = \{(x, \mu_F(x)) \mid x \in X\}.$$

Fuzzy sets may be discrete or continuous. Let X be the set of possible grades a student may receive on a paper: $X = \{A, B, C, D, F\}$. The discrete fuzzy set 'High grades' (H) could be represented as the following (note the alternative notation for fuzzy sets): $H = \{A/1.0, B/0.7, C/0.2, D/0, F/0\}$. Alternately, let X be the set of possible ages for a human being. Then the continuous fuzzy set 'about 50 years old' (G) could be represented as (Jang and Sun, 1995):

$$G = \{(x, \mu_G(x)) \mid x \in X\}, \text{ where, } \mu_G(x) = \frac{1}{(1 + ((x-50)/5)^4)}.$$

The fuzzy set can be specialised to the classical set simply by creating a membership function that only yields the values 0 and 1 (Zadeh, 1965). This illustrates that any value $x \in X$ either does or does not belong to A.

Basics of the genetic algorithms

John Holland (1975) pioneered genetic algorithms as initiated with cellular automata. Genetic algorithms use an evolutionary approach (Whitley, 1994). A genetic algorithm uses a population as a set of potential solutions and each solution has a fitness value and the population evolves at every generation by using a Darwinian approach.

Genetic algorithms have been successfully used for the search and optimisation problems. They have advantages such as finding optimal solutions. A genetic algorithm cannot always find the global optima, but generally can find near optimum solutions in complex search spaces. Koza (1995) expresses that genetic algorithm is fast in effectively searching complex, highly nonlinear, multidimensional search spaces practically. The more interesting point is that it is not necessary to know the problem domain or how the fitness function works.

At every generation of the genetic algorithm, the population undergoes a natural analogy to reproduction, *sexual crossover* and *mutation* operators alongside and the survival of fittest test. These are inspired by the evolution of a population (Koza, 1995). At every generation, the population contains a better set of solutions in parallel to the *fitness criteria.* Each individual in the population represents a solution in the search space. Fitness hints how the individual is situated according to the solution. All the operators that are inspired from biological evolution are targeted to find better individuals. Having modelled natural life, offsprings are generated by using *recombination (crossover)*, so offsprings have the probability of being mutated.

According to the given problem, the search space is covered with possible solutions. Two main parts are considered specific to the problem under investigation by the genetic algorithm; these are encoding and the evaluation function, *i.e.*, the fitness function. Every individual is called a chromosome and is represented by a fixed length string containing partial solutions called genes. Genes are described with alphabet. Genes are traditionally encoded in bits, and a chromosome is a string of (0,1) bits but other encoding schemes are also possible (Whitley, 1994). Algorithm 1 provides pseudo code of genetic algorithm.

Simple Genetic Algorithm.

```
Initialize the population
Evaluate fitness of the individuals in the population
While (The termination Criteria is not satisfied)
 {
   Select parents from the population for reproduction
   Apply recombination(sexual crossover)
   Apply mutation
   Evaluate fitness of individuals in the population
 }
```

In Algorithm 1, the population is initialised and individuals of the first iteration (generation) are evaluated. Two primary parameters in genetic algorithm are population size and termination criteria. Some examples of termination criteria are maximum number of iterations, and a change in fitness value does not exceed a given threshold or the best so far (worst so far) fitness in the population does not improve. The genetic process continues until the termination criteria is satisfied. While it is not satisfied, population members are selected for reproduction, and individuals mate for reproduction to generate offspring with sexual crossover and mutation. The main steps in Algorithm 1 are:

Step 1 *Initialise population*: At the initial step, the population consists of randomly generated individuals. A possible encoding of a chromosome can be in binary representation (see Figure 1).

Step 2 *Evaluate fitness of individuals in population*: Each individual in the population, and each of the children that are reproduced from the parents, are evaluated and assigned a fitness value to indicate how it is located in the solution set. Regarding the problem type, the smaller (greater) the estimated fittest value, the more survival chance is given to the individual to remain in the population.

Step 3 *Select parents from the population for reproduction*: At every generation, individuals from the population mate. Individual evaluation, fitness value, plays a significant role in the selection of mates for breeding. Despite a lower fitness value, an individual has the probability of being selected to be put in the population for the latter generation or mating. However, probabilistically it has a lower chance. This is good to preserve diversity in the search space; it avoids premature convergence. Some very well known selection methods are roulette wheel selection and tournament selection.

In the roulette wheel selection method, a probability value is assigned for each individual depending on its fitness. Parents are picked by taking the probability value into account. Tournament selection picks a number of individuals to select the best among them to be a spouse for mating. The other spouse is also picked in the same way and they mate for reproduction.

Step 4 *Apply recombination (sexual crossover)*: Children are formed using parents. This is achieved by having parents doing crossover with a crossover probability. If they do not undergo crossover, children will be exactly the same as the parents. Otherwise, they swap random portions of their chromosomes. There are variations of crossover operators such as: single point crossover, multipoint crossover, and uniform crossover. Crossover leads to effective search of solutions by combining optimal sub-solutions. Single point crossover is a commonly used crossover technique that swaps the contents of two chromosomes after randomly selecting a split point (see Figure 2).

Step 5 *Apply mutation*: After crossover, generated children are most likely to have mutation. Genes in the chromosome may change stochastically. This helps moving locally or globally in the search space. All genes in an individual can stochastically be changed. As shown in Figure 3, after applying mutation on a gene, the gene content changes from 0 to 1.

Figure 1 Binary representation of a chromosome

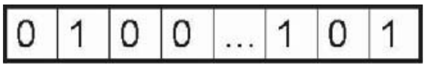


Figure 2 A single point crossover example (see online version for colours)

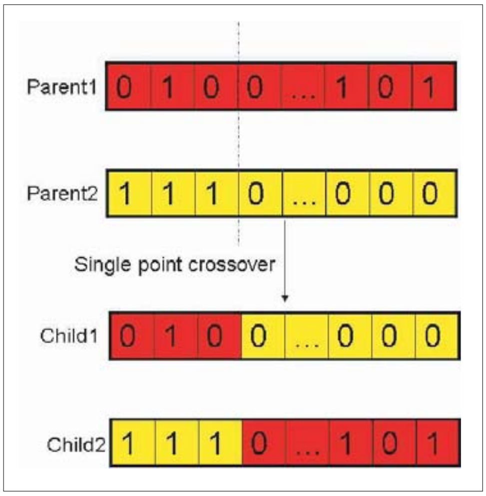
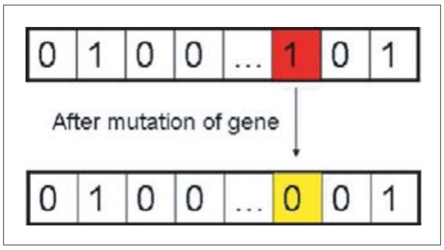


Figure 3 A mutation example (see online version for colours)



Fuzzy genetic rule-based systems

In order to build a fuzzy rule-based system, the major task is to find an appropriate fuzzy rule set which represents the problem (Ishibuchi *et al*., 1999). Genetic algorithms have shown to be a powerful tool for performing:

- generation and optimisation of fuzzy rule-base
- generation and tuning of membership functions (Cordon *et al*., 2001).

In a fuzzy rule-based system, fuzzy *if-then* rules for an *n*-dimensional pattern classification problem are defined as follows (Ishibuchi *et al*., 1999):

$$\text{Rule} \quad R_j : \text{If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{jn} \text{ then Class } C_j \text{ with } CF_j, \tag{6}$$

where:

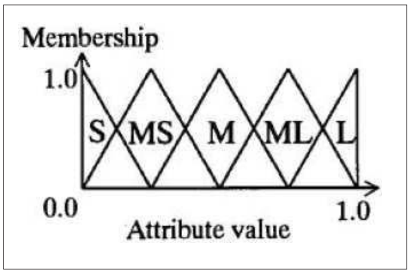
R_j = the label of the j-th fuzzy *if-then* rule
j = indexes the number of rules
$x = (x_1, x_2, \ldots, x_n)$ = an n-dimensional pattern vector
A_{ij} = an antecedent fuzzy set with linguistic label (*i.e.*, a linguistic value such as *small* or *large)* on the i-th axis
C_j = a consequent class
CF_j = a certainty grade.

As the antecedent fuzzy sets A_{ij}'s, five linguistic values shown in Figure 4 and 'don't care' have been used. Therefore, the number of combinations of the antecedent fuzzy sets is 6^n, which is very large in the case of high-dimensional problems.

As shown in Figure 4, the meaning of each linguistic value is specified by a triangular membership function on the unit interval [0,1]. 'don't care' has been handled by a special linguistic value with the following membership function:

$$\mu_{don't\,care}(x) = \begin{cases} 1 0 \le x \le 1, \\ 0 \text{ otherwise.} \end{cases} \tag{7}$$

Figure 4 Membership functions of five linguistic values



Notes: S: Small, MS: Medium Small, M: Medium, ML: Medium Large, and L: Large.

In this study, a small number of fuzzy *if-then* rules have been randomly generated. When antecedent fuzzy sets of a fuzzy *if-then* rule are specified, its consequent class and certainty grade are determined by applying a heuristic (Nakashima *et al*., 2004) as follows:

Step 1 Calculate $\beta_{Class\ h}(R_j)$ for Class h as:

$$\beta_{Class\ h}(R_j) = \sum_{x_p \in Class\ h} \mu_{j1}(x_{p1}) \times \ldots \times \mu_{jn}(x_{pn}), h = 1, 2, \ldots, C, \tag{8}$$

where $\mu_{ji}()$ is the membership function of the fuzzy set A_{ji}.

Step 2 Find Class $\hat{h}$ that has the maximum value of $\beta_{Class\ h}(R_j)$:

$$\beta_{Class\ \hat{h}}(R_j) = \max\{\beta_{Class\ 1}(R_j), \ldots, \beta_{Class\ C}(R_j)\}. \tag{9}$$

If two or more classes take the maximum value, the consequent class C_j of the rule R_j cannot be determined uniquely. In this case, specify C_j as $C_j = 0$. If a single class takes the maximum value, let C_j be Class h. The grade of certainty CF_j is determined as:

$$CF_j = \frac{\beta_{Class\ \hat{h}}(R_j) - \bar{\beta}}{\sum \beta_{Class\ h}(R_j)}, \tag{10}$$

where:

$$\bar{\beta} = \frac{\sum_{h \neq \hat{h}} \beta_{Class\ h}(R_j)}{c-1}. \tag{11}$$

After generating a small number of initial fuzzy *if-then* rules, the genetic algorithm has been applied to optimise the initial rule set so that it will be able to classify the test set with a reasonable classification accuracy. This work does not involve the adjustment of membership functions or certainty grade.

The fuzzy genetic rule-based classification systems are basically categorised into Pittsburgh Smith (1980), Michigan Goldberg (1989) and Booker *et al.* (1989) approaches. In the Pittsburgh approach, a set of fuzzy rules is coded as a string to be used in the genetic algorithm, and the performance of the rule set is evaluated for classification. In contrast, the Michigan approach codes every individual fuzzy rule as a string, and the performance of the rules is evaluated explicitly. Here, we have followed the implementation of Pittsburgh approach as it is faster compared to Michigan.

Assume that the fuzzy *if-then* rule R_j in Equation (6) is denoted by its n antecedent fuzzy sets as $R_j = A_{j1} \dots A_{jn}$. That is, R_j is coded as a string (chromosome) of length n. Let S be a set of N fuzzy *if-then* rules (*i.e.*, $S = \{R_1, \dots, R_N\}$). S is denoted by a concatenated string of the length $n \times N$, where each substring of length n corresponds to a single fuzzy *if-then* rule. In other words, the rule set S is formulated as:

$$R_j = A_{11} \dots A_{1n} A_{21} \dots A_{2n} \dots A_{N1} \dots A_{Nn}. \tag{12}$$

Figure 5 (adapted from Castro *et al.*, 2004) shows a chromosome representing a fuzzy rule-based set containing k rules related to a problem with four features, *X1*, *X2*, X3, and *X4* (Castro *et al.*, 2004). Each rule is represented by four genes indicating the index of the fuzzy set associated to the four features, respectively. The 'don't care' value is represented by 0 within the chromosome.

Figure 5 Chromosome representing a fuzzy rule base with k rules

2	3	0	1	.....	0	2	1	3
R_1				.....	R_k			

The fitness of the rule set S is measured as:

$$fitness(S) = NCP(S) \tag{13}$$

where $NCP(S)$ is the number of correctly classified training patterns by S.

The genetic algorithm has been set to use the uniform crossover, where each substring is handled as a block. That is, some rules are exchanged between the two parents by the crossover. A mutation operation has been set to randomly replace an antecedent fuzzy set of a rule with another one.

4 Experimental analysis

This section is dedicated to describe the evaluation criteria and the conducted experiments. We summarise the experimental results and highlight the performance and applicability of the system.

4.1 Evaluation criteria

After linking the customers of the service provider network into a cluster tree, it has to be decided to divide the tree at a particular level to generate the clusters. For the community identification problem, the validity of the clusters produced by a clustering algorithm is an important consideration. The reason may be articulated as follows: once the clustering is complete, each of the clusters must be labelled and then used in the classification task. Since the number of generated clusters is subjective to the accuracy of the classifier, the approach that has been used to determine the validity of the cluster divisions is to compare the accuracy of the classifiers built based on different cluster divisions. That is, we divide the cluster tree at a level that generates two clusters at the beginning. Then we labelled the clusters and build a classifier using SVM algorithm. At the next iteration, we increase the number of clusters by 2 and we divide the cluster tree such that it generates that particular number of clusters. The stopping criteria is when an acceptable classification accuracy is obtained using the generated clusters. To evaluate the accuracy of the classifier model, the cross validation method (Edelstein, 1999) has been used. Basically, the data is randomly divided into 5 disjoint groups. The first group is set aside for testing and the other four are put together for model building. The model built on the 80% group is then used to predict the group that was set aside. This process is repeated a total of five times as each group in turn is set aside. Finally, a model is built using all the data. The mean of the five independent error rate predictions is used as the error rate for the final model.

4.2 Experimental results

Using the CDR data, a cluster tree has been built. In all the experiments, α has been set to 0.75 in the distance measure formula. By choosing such a large value, we give more weight to the second order distance than the first order distance. We believe that customer's indirect calling patterns will provide more useful information compared with direct calling patterns since a customer more likely calls numbers which are not inside his/her home service provider.

The overall effectiveness of the clustering algorithm is calculated using overall accuracy of the classifier model. This overall accuracy measurement determines how well the clustering algorithm is able to create communities that contain customers with similar behaviours. The number of correctly classified customers in a cluster is referred to as the True Positives (TP). Customers that are not correctly classified are considered False Positives (FP). The overall accuracy is thus calculated as follows:

$$\textit{overall accuracy} = \frac{\sum TP \text{ for all classes}}{\text{total number of customers}}. \tag{14}$$

Figure 6 and Table 1 show how the clustering algorithm was evaluated with *K* being the number of clusters created from the cluster tree. The minimum, maximum, and average results for the clustering algorithm are shown as well. The classification algorithm used for evaluating clusters is SVM. The reason that fuzzy genetic classifier has not been used is because of its running time.

Figure 6 Accuracy using hierarchical clustering with SVM (see online version for colours)

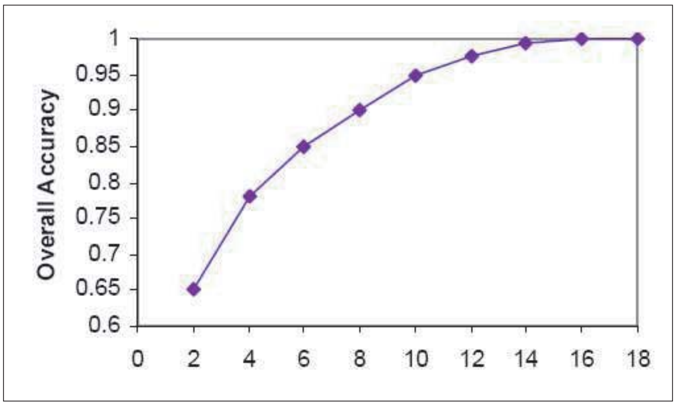


Table 1 Accuracy using SVM

	Average	*Minimum*	*Maximum*
Accuracy	89.97%	65%	100%
Number of cluster	9	2	18

Considering the above analysis, it has been decided to set the number of clusters to ten for building the classification model. That is, every cluster has been labelled as a community, and customers within all the ten communities have been used as the training set for the classification algorithm.

The examination of the overall accuracy between the SVM classifier and the fuzzy genetic approach using five-fold cross validation can be seen in Table 2. The LIBSVM default parameter settings have been used for running the SVM algorithm as follows: the kernel function is radial basis function with γ is set to $1/k$, where k is the number of attributes in the dataset. Based on some initial test runs, the following settings have been applied for running the fuzzy genetic classifier:

1 The number of fuzzy rules: 40, 60 or 80

2 The number of rule sets: 20

3 Crossover probabilities: 0.9

4 Mutation probabilities: 0.1

5 Stopping condition: 500 generations.

Table 2 Overall accuracy of each algorithm

Algorithm	*Average (%)*	*Minimum(%)*	*Maximum(%)*
SVM	99.75	97.95	100
Fuzzy genetic	86.4	75.25	94

For the given CDR data, SVM has an average overall accuracy of 98.5%, whereas in comparison, the fuzzy genetic classifier has an overall accuracy of 82.5%. Thus, we find that SVM outperforms the fuzzy genetic classifier by almost 13%. This shows that using genetic algorithm for tuning the fuzzy rules of the fuzzy genetic classifier results in a reasonable accuracy but not as high as the accuracy obtained by SVM. However, the rules in the fuzzy genetic classifier are easily understandable and interpretable.

The runtime of both approaches is an important consideration because the model building phase is computationally time consuming. For the analysis, all operations are performed on a Dell Optiplex 745 with an Intel Core2 Duo 6600 @ 2.4 GHz processor and 3 GB of RAM. The number of data objects in the training set is 2000. In general, the runtime for the SVM classifier was significantly less than the fuzzy genetic algorithm when building the classification models. For example, with 2000 objects running five-fold classification took less than a second, whereas fuzzy genetic took much longer to build the classification model. The running time of the fuzzy genetic approach while varying the number of fuzzy rules is shown in Table 3. Although the SVM classifier was faster, the size of the training set is ultimately limited by the amount of memory because both approaches must load the entire training set into memory before building the model.

Table 3 Running time of fuzzy genetic classification algorithm

Data set	*Number of rules*	*Accuracy (%)*	*CPU time (min)*
CDR data	40	86.4	448
	60	80.80	365
	80	73.15	378

According to our experiments SVM has two significant benefits when applied to the community classification problem. First, it is easy to use in a way such that it does not rely on extensive tuning of parameters by users like the fuzzy genetic algorithm, and it requires minimum tuning. This feature makes it easier to use than other learning models, which need more user involvement in the model tuning process. Second, having a very strong mathematical and theoretical foundation, SVM is computationally efficient and outstandingly well-performing on classification problems.

On the other hand, genetic operations in Pittsburgh approach are not directly based on the performance of fuzzy if-then rules. Thus even good fuzzy if-then rules (especially included in poor rule sets) can easily disappear by generation update. Of course, good rule sets are inherited to the next population as elite individuals. Further, since a population consists of a number of rule sets, computation and time and memory storage are main concerns in fuzzy genetic algorithm based on the Pittsburgh approach.

5 Discussion

Recall that, fuzzy classification is one possible approach that offers more convenience for selecting customer subgroups and for measuring the efficiency and validity of the communities regarding the classification goal, like the marketing campaign design. The classes identified by SVM are crisp and the classifier model does not provide the ability for assigning a particular customer to more than one class. However, the characteristics of the fuzzy genetic classifier model which contains a set of fuzzy *if-then* rules can be used for possibly doing fuzzy classification. The procedure applied for assigning a customer to more than one class with different degrees of membership (*i.e.*, fuzzy classification) is performed as follows.

The fuzzy *if-then* rules in the final classifier model are divided into n subsets, where n is the number of distinct available classes (communities). For instance if the clustering has identified five communities, then there exist five subsets. For a particular customer, every subset is examined to see whether there is any compatible rule with the feature vector representing the customer calling pattern. If exists, such a rule is able to assign the particular customer to the class of the subset under test. The certainty factor of the compatible rule will then represent the degree of membership of this customer to its identified class (community). If the algorithm finds more than one compatible rule with the customer feature vector from the same subset, then the average certainty factor of all compatible rules is considered to be the customer's degree of membership to this class. The same procedure will be repeated for a particular customer using all the subsets, and a user may be assigned to different classes with different degrees of membership. The class in which the customer has the highest degree of membership will be identified as the main class. Meanwhile, this particular customer may belong to other existing classes with lower degrees of membership. Here it is worth noting that a customer does not belong to a particular class if its degree of membership to that class is zero.

Table 4 displays a sample final fuzzy rule set generated by the fuzzy genetic classifier. Suppose that pattern P is compatible with rules {1, 7, 10}, with corresponding certainty factors of {0.9501, 0.4565, 0.4447}. Then, P is assigned to class 1 with degree of membership equals to $AVG(0.9511, 0.4565) = 0.7038$ since it is compatible with two fuzzy rules {1, 7} belonging to class 1. Pattern P is also compatible with rule {10} from class 1. Therefore, this rule can assign P to class 2 with the degree of membership equals to 0.4447.

Table 4 Example of a final fuzzy rule set generated by fuzzy genetic classifier

Rule	*Class*	*CF*
1	1	0.9501
2	2	0.2311
3	2	0.6068
4	2	0.4860
5	1	0.8913
6	2	0.7621
7	1	0.4565
8	2	0.0185
9	1	0.8214
10	2	0.4447

6 Summary and conclusions

In this paper, we demonstrated how cluster analysis can be used to effectively identify calling communities by using information derived from the CDR data. We used the information extracted from the cluster analysis to identify customer calling patterns. Customers calling patterns are then given to a classification algorithm to generate a classifier model for predicting the calling communities of a customer. This work is especially important for targeted marketing campaigns in the telecommunication industry since the CDR data is often the only primary data source available for the customers. Based on the assumption that customers in the same calling community might behave similarly, targeted efforts can be focused on certain communities. Further, the fuzzy classification proposed in this project provides more convenience for selecting customer communities and for measuring the efficiency and validity of the communities regarding the marketing campaign design. The flexibility of the fuzzy approach featured by the application of membership functions provides the ability to increase or decrease the homogeneity between the targeted customers depending on whether the proposed products are very specific or intended for a large community.

References

Allwein, E., Schapire, R. and Singer, Y. (2000) 'Reducing multiclass to binary: a unifying approach for margin classifiers', *The Journal of Machine Learning Research*, Vol. 1, pp.113–141.

Berger-Wolf, T.Y. and Saia, J. (2006) 'A framework for analysis of dynamic social networks', *Proc. KDD 06*, pp.523–528.

Booker, L.B., Goldberg, D.E. and Holland, J.H. (1989) 'Classifier systems and genetic algorithms', *Artificial Intelligence*, September, Vol. 40, Nos. 1–3, pp.235–282.

Burt, R.S. (1976) 'Positions in networks', *Social Forces*, Vol. 55, pp.93–122.

Castro, P.A.D., Santoro, D., Camargo, H.A. and Nicoletti, M.C. (2004) 'Improving a Pittsburgh learnt fuzzy rule base using feature subset selection', *Proceedings of the International Conference on Hybrid Intelligent Systems*, Kitakyushu, pp.180–185.

Chakrabarti, S., van der Berg, M. and Dom, B. (1999) 'Focused crawling: a new approach to topic-specific web resource discovery', *Proc. Eighth Intl Conf. World Wide Web (WWW 99)*, pp.1623–1640.

Chang, C.C. and Lin, C.J. (2001) 'LIBSVM: a library for support vector machines', http://www.csie.ntu.edu.tw/~cjlin/libsvm (last accessed 30 September 2008).

Cordon, O., Herrera, F., Gomide, F., Hoffmann, F. and Magdalena, L. (2001) 'Ten years of genetic fuzzy systems: current framework and new trends', *Proceedings of the Joint IFSA World Congress and NAFIPS International Conference*, Vancouver, Canada, pp.1241–1246.

Cristianini, N. and Taylor, J.S. (2000) *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press.

Daneshgar, F. and Ho, S. (2008) 'Sociological factors affecting trust development in virtual communities', *Int. J. Networking and Virtual Organisations*, Vol. 5, No. 1, pp.51–63.

Edelstein, H.A. (1999) *Introduction to Data Mining and Knowledge Discovery*, 3rd ed., Potomac, MD: Two Crows Corp.

Fiedler, M. (1973) 'Algebraic connectivity of graphs', *Czechoslovakian Math. J.*, Vol. 23, No. 98, pp.298–305.

Fiedler, M. (1975) 'A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory', *Czechoslovakian Math. J.*, Vol. 25, No. 100, pp.619–637.

Flake, G.W., Lawrence, S., Giles, C.L. and Coetzee, F.M. (2002) 'Self-organization and identification of web communities', *Computer*, Vol. 35, No. 3, pp.66–71.

Girvan, M. and Newman, M.E.J. (2002) 'Community structure in social and biological networks', *Proc. Natl Academy of Science (PNAS)*, Vol. 99, No. 12, pp.7821–7826.

Goldberg, J.D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA: Addison-Wesley.

Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press.

Ishibuchi, H., Nakashima, T. and Kuroda, T. (1999) 'A hybrid fuzzy genetics-based machine learning algorithm: hybridization of Michigan approach and Pittsburgh approach', *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Tokyo, Japan, Vol. 1, 12–15 October, pp.296–301.

Ishibuchi, H., Nozaki, K. and Tanaka, H. (1992) 'Distributed representation of fuzzy rules and its application to pattern classification', *Fuzzy Sets Systems*, November, Vol. 52, No. 1, pp.21–32.

Jang, J.S.R. and Sun, C.T. (1995) 'Neuro-fuzzy modeling and control', *Proceedings of the IEEE*, Vol. 83, No. 3, pp.378–406.

Kernighan, B.W. and Lin, S. (1970) 'An efficient heuristic procedure for partitioning graphs', *Bell System Technical*, Vol. 49, pp.291–307.

Kianmehr, K. and Alhajj, R. (2008) 'Effectiveness of machine learning techniques for automated identification of calling communities', *Proceedings of the International Conference on Information Visualisation (Symposium on Open Source Intelligence and Web Mining)*, London, UK, July.

Kianmehr, K. and Alhajj, R. (2009) 'Calling communities analysis and identification using machine learning techniques', *Journal of Expert Systems with Applications*, Part 2, April, Vol. 36, No. 3, pp.6218–6226.

Kim, D-H. and Jeong, H. (2005) 'Systematic analysis of group identification in stock markets', *Physical Rev. E*, Vol. 72, No. 4, id. 046133.

Kleinberg, J.M. (1999) 'Authoritative sources in a hyperlinked environment', *J. ACM*, Vol. 46, No. 5, pp.604–632.

Koza, J.R. (1995) 'Survey of genetic algorithms and genetic programming', *Proceedings of WESCON Conference*, pp.589–594.

Kumar, R., Raghavan, P., Rajagopalan, S. and Tomkins, A. (1999) 'Trawling the web for emerging cyber-communities', *Proc. Eighth Intl Conf. World Wide Web (WWW 99)*, pp.1481–1493.

Nakashima, T., Ishibuchi, H. and Bargiela, A. (2004) 'Constructing fuzzy classification systems from weighted training patterns', *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Vol. 3, pp.2386–2391.

Nasrullah, M. and Larsen, H.L. (2006) 'Structural analysis and mathematical methods for destabilizing terrorist networks', *Proceedings of the International Conference on Advanced Data Mining Applications*, Lecture Notes in Artificial Intelligence (LNAI 4093), Springer-Verlag, pp.1037–1048.

Okkonen, J. (2004) 'How virtuality affects knowledge work: points on performance and knowledge management', *Int. J. Networking and Virtual Organisations*, Vol. 2, No. 2, pp.153–160.

Palla, G., Derenyi, I., Farkas, I. and Vicsek, T. (2005) 'Uncovering the overlapping community structure of complex networks in nature and society', *Nature*, 9 June, Vol. 435, No. 7043, pp.814–818.

Pirolli, P., Pitkow, J. and Rao, R. (1996) 'Silk from a sows ear: extracting usable structures from the web', *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pp.118–125.

Pons, P. and Latapy, M. (2005) 'Computing communities in large networks using random walks', *Proc. 20th Intl Symp. Computer and Information Sciences (ISCIS 05)*, pp.284–293.

Pothen, A., Simon, H. and Liou, K.P. (1990) 'Partitioning sparse matrices with eigenvectors of graphs', *SIAM J. Matrix Analysis and Application*, Vol. 11, No. 3, pp.430–452.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V. and Parisi, D. (2004) 'Defining and identifying communities in networks', *Proc. Natl Academy of Science (PNAS)*, Vol. 101, No. 9, pp.2658–2663.

Ratsch, G., Smola, A.J. and Mika, S. (2002) 'Adapting codes and embeddings for poly-chotomies', *RSISE, CSL*, Machine Learning Group, Canberra, Australia and Frauenhofer FIRST, Berlin, Germany.

Scott, J. (1991) *Social Network Analysis: A Handbook*, Newbury Park, CA: Sage Publications.

Shi, J. and Malik, J. (2000) 'Normalized cuts and image segmentation', *IEEE Trans. Pattern Analysis and Machine Intelligent*, August, Vol. 22, No. 8, pp.888–905.

Smith, S.F. (1980) 'A learning system based on genetic algorithms', PhD dissertation, University of Pittsburgh, Pittsburgh, Pennsylvania.

Striukova, L. and Rayna, T. (2008) 'The role of social capital in virtual teams and organisations: corporate value creation', *Int. J. Networking and Virtual Organisations*, Vol. 5, No. 1, pp.103–119.

Tantipathananandh, C., Berger-Wolf, T. and Kempe, D. (2007) 'A framework for community identification in dynamic social networks', *Proceedings of the 13th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining (KDD '07)*, ACM, New York, New York, pp.717–726.

Tyler, J.R., Wilkinson, D.M. and Huberman, B.A. (2003) 'Email as spectroscopy: automated discovery of community structure within organizations', *Proc. First Intl Conf. Communities and Technologies (C&T 03)*, pp.81–96.

Vapnik, V.N. (1998) *Statistical Learning Theory*, New York: John Wiley, p.732.

Wasserman, S. and Faust, K. (1994) *Social Network Analysis*, Cambridge University Press.

Werro, N., Stormer, H. and Meier, A. (2006) 'A hierarchical fuzzy classification of online customers', *Proceedings of the IEEE International Conference on E-Business Engineering (ICEBE '06)*, Shanghai, China, October, pp.256–263.

Whitley, D. (1994) 'A genetic algorithm tutorial', *Statistics and Computing*, Vol. 4, No. 2, pp.65–85.

Yan, L., Fassino, M. and Baldasare, P. (2005) 'Predicting customer behavior via calling links', *Proceedings of IEEE International Joint Conference on Neural Networks*, Vol. 4, pp.2555–2560.

Yang, B., Cheung, W. and Liu, J. (2007) 'Community mining from signed social networks', *IEEE Trans. on Knowl. and Data Eng.*, Vol. 19, No. 10, pp.1333–1348.

Zadeh, L. (1965) 'Fuzzy sets', *Information and Control*, Vol. 8, No. 3, pp.338–353.

Note

1 http://www.mathworks.com/ (last accessed 30 September 2008).