Personalised Online Sales Using Web Usage Data Mining

Zhang, X., Edwards, J.M., and Harding, J.A.

Abstract

Practically every major company with a retail operation has its own web site and online sales facilities. This paper describes a toolset that exploits web usage data mining techniques to identify customer internet browsing patterns. These patterns are then used to underpin a personalised product recommendation system for on-line sales. Within the architecture, a Kohonen neural network or self organising map (SOM) has been trained for use both off-line, to discover user group profiles, and in real time to examine active user click stream data, make a match to a specific user group, and recommend a unique set of product browsing options appropriate to an individual user. Our work demonstrates that this approach can overcome the scalability problem that is common among these types of system. Our results also show that a personalised recommender system powered by the SOM predictive model is able to produce consistent recommendations.

Keywords: data mining, neural network, SOM, internet sales

1.0 Introduction

Manufacturing industry is increasingly seeking to engage with their customer base through personalised sales. This may be through assisting potential customers to identify an appropriate product, or alternatively by customising products to meet particular customer requirements. In either case, if potential customers can fulfil their requirements quickly and easily, they are more likely to make a purchase. Hence, practically every major company with a retail operation has its own web site and online sales facilities, where web personalisation has become an essential element, improving the customer experience and encouraging the customer's loyalty.

The general problem of many online sales sites is "too much choice". Finding the right product becomes time consuming and the benefit of the online service is reduced. Web personalisation aims to "provide users with the information they want, without expecting them to ask for it explicitly" [1]. In [2], a discussion of personalised recommendation applications is presented where typical examples of contemporary systems include online vendors Amazon.com, and ebay.com. Their approach is based on observing peoples' web site navigation behaviour and looking for recurring themes. They can then make product recommendations which comply with a particular customer's preference on return visits to the site. However, this advice may not be helpful as in many cases, when a user returns to the site, they are searching for a different type of product.

In our work, web personalisation means providing tailored information to the individual in order to make navigation of a site easier, enabling products that could meet the customer's requirements to be easily located. This personal support is based on their current (rather than previous) navigation behaviour, which is discovered in real time

during their current visit. Data mining has the potential to be successful in identifying customer navigation patterns. However, it is also computer intensive and cannot provide satisfactory (real-time) response times with today's processing capability. To overcome this problem our toolset uses an architecture that combines: 1) the benefits of an off line pattern discovery module for long term learning about multiple users, with; 2) a real-time product recommendation module to personalise the web site visit for any currently active visitor. The offline module uses data mining techniques to identify slowly changing customer-browsing patterns from multiple anonymous web site visits. These patterns are then used to underpin the personalised customer product recommendation module that identifies real time customer navigation behaviour and matches it to the discovered offline patterns. This two phase architecture is shown in Figure1.

2.0 Data Mining

Data mining has recently emerged as a growing field of multidisciplinary research. It combines disciplines such as databases, machine learning, artificial intelligence, statistics, automated scientific discovery, data visualization, decision science, and high performance computing. It is our contention that data mining techniques can be used to provide the knowledge required to assist users in locating relevant information on the web, through automating the analysis of the current users navigation and combining this with data-mined knowledge of multiple users' buying behaviour.

While a customer is visiting a web site they leave a trail (which characterises their requirements) in the form of a web log. Data mining provides the facilities for automated discovery of the knowledge within these logs to enable predictive modelling of the current customer's navigation behaviour [3]. In addition, this type of system can provide an improved understanding of a wide-ranging customer base, typically helping to define

types of customer groups. This has potential as useful knowledge for supporting strategic planning activities, typically used in targeting particular customer segments for new product planning and design.

Web data mining can be defined as applying data mining techniques to automatically discover and extract useful information from the World Wide Web. In general, there are three research areas in web data mining;

- **Web structure mining** is concerned with the discovery of the hyperlink structure of the web in order to extract relevant connectivity patterns. It focuses on the field of mining HTML or XML tags.
- **Web content mining** is concerned with the automatic discovery of patterns within Web information such as HTML pages, images, audios, videos and Emails.
- **Web usage mining** is concerned with the discovery of usage patterns from web data, in order to understand the user's online navigation behaviour.

Current approaches focus on web content mining and web structure mining in order to reveal the semantic relationships among search strings. Our work is concerned with web usage mining techniques to analyse search-driven web sites.

3.0 Related Work

Personalised product recommendation systems have the potential to help users find information and services that enable them to decide which products to purchase.

Within these systems two approaches of information filtering are usually adopted: content filtering and collaborative filtering. Content filtering refers to recommending items based on analysis of the contents of a product's supporting information [4], while in collaborative or social-based filtering, the items are recommended based on the recommendations of other users [5, 6, 7]. Social-based filtering relies on users' ratings; therefore, the success of the system depends on the current number of users on the system and the rating activities among the users.

3.1 Content filtering

WebSIFT (Web Site Information Filter System) is an online information system that employs Web usage mining [8]. The system is built for web personalisation, and a typical usage mining process is examined in three phases: Data Pre-processing, Pattern Discovery and Pattern Analysis. The web topology and content data are considered as valuable domain knowledge which is used to construct what is referred to as a belief set ie, sets of web pages that are "related" if they are linked together and/or share some similarities in content. The patterns discovered using data mining, that were previously unknown to the belief set, are considered to be the more interesting patterns because it is likely that they will be unexpected to the user.

Another web page recommender system called FAB is proposed in [9, 10]. This content-based system relies on multiple agents to learn users' preferences by analysing the way they access web pages. In this work a new performance measure called the Normalized Distance-based Performance Measure (NDPM) is proposed. NDPM is the distance between the user's ranking of a set of documents and the system's ranking of the same documents, normalised to lie between 0 and 1. Although their performance on the collection of user's preferences shows potential, there are disadvantages in comparison with other recommendation approaches, as their system requires users to select their desired area of interest when they register. This information is then used as the initial user profile to which items in the database are matched and recommendations given.

A further example of content-based filtering is presented by Mooney and Roy, in their paper detailing a recommendation system for online bookstores [11]. Their approach utilizes information extraction and a machine learning algorithm for text categorization. The advantage of their approach is that it is able to recommend previously unrated items

to users with unique interests and to provide explanations for its recommendations. However, it has the drawback that it does not also provide collaborative recommendations from other users.

In [12], Frias-Martinez et.al. present a new approach through constructing "E-libraries" that can customise the online users' experience when searching for books, journals, articles etc. Their content filtering system can automatically learn user preferences and goals to create an adaptive interface which delivers a tailored service to the users. Within their system, both supervised and unsupervised data mining techniques have been employed. Within the paper the authors compare a customer driven active personalisation approach with their passive personalization. The work demonstrates that greater customer value is gained where customers are unable to express their preferences. However, their system is incapable of understanding the temporal nature of a customer's behaviour and cannot track their concept drift

3.2 Collaborative filtering

In [13], clustering techniques are used to improve the performance of collaborative filtering. It is difficult to apply the distance-based clustering techniques on web usage mining, and this is partly because it is hard to define the distance between user sessions, which are often in a high dimensional format. Other clustering methods have been explored for partitioning, rather than the distance-based clustering approach that is used to build up usage profiles. Typically ARHP (Association Rule Hyper-graph Partitioning), which is a partitioning technique based on the frequent item sets generated by association rule mining and PACT (Profile Aggregations based on Clustering Transactions). AHRP is used to group and summarise user transactions.

The notion of privacy within a probabilistic model for a collaborative filtering system is examined by Canny [14]. The system he introduces adopts a protocol for encrypting data to protect people's privacy. Experimental results obtained using the algorithm are shown to be satisfactory and the method is considered accurate when compared to other collaborative filtering systems. It also has advantages in terms of speed and size of data set over previous collaborative filtering algorithms.

Latent Semantic Indexing (LSI) [15] has been used by Sarwar [16] to address the problem of poor relationships among users data. In the MovieLens project, the authors utilise LSI with singular value decomposition (SVD) as their underlying matrix factorisation algorithm to fit into the collaborative filtering system for dimension reduction. SVD technology has been used to produce a lower-dimensional representation of the original customer-product space, and to capture latent relationships between customers. However, their experiments show that the results of SVD are unsatisfactory over a high dimensional data set.

A further example of the collaborative filtering technique is presented in [17], where the author develops a system called e-Vzpro. The aim is to overcome problems of too little information, or too much unrelated information. The system comprises a recommendation tool powered by association mining algorithms, which are used to address the problem of sparse binary data. The system uses a two-phase approach: in the first phase, customer historical data are pre-processed and analysed through association mining algorithms to generate rule sets; in the second phase, a scoring algorithm is used to rank the recommendations online. Their experimental results are compared to other dependency networks and item-based algorithms with cosine similarity measures, and e-

Vzpro is shown to outperform these approaches. However, this system does not make real time recommendations based on the current behaviour of the user.

In [18], Jansen presents a three-step methodology for the study of Web-searching, this three-stage process is composed of data collection, preparation, and analysis of web server access logs. A web based application, to record client-side user's interactions, that supplements the analysis methodology is also presented.

The system is capable of dealing with transaction logs in a non-intrusive manner, collecting interaction data about the Web searching process from multiple users, through employing a collaborative approach that improves scalability. However, their system exposes the limitation that the log data may not be complete due to caching of server data on the client machine or proxy servers. In addition, their application cannot capture qualitative aspects of the log that define the behaviour of the user, that can be used for predictive modelling.

3.3 Content and Collaborative filtering

An approach which considers both content-based and collaborative filtering techniques is described in [19]. Here a recommender system called Yoda has been developed to support large-scale web-based applications that require accurate recommendations in real time. The two key contributions of the work are: 1) improvements of the traditional Nearest Neighbour algorithm when used for collaborative filtering and; 2) a novel filtering mechanism to extend the Locality Sensitive Hashing technique, by incorporating a novel distance measure to make the offline process scalable. This aggregation function is used in an offline process to generate predefined recommendations, called cluster-wish lists, for each class of user. The current implementation focuses on recommending music

CDs. However, the drawback of this approach is that it requires the content of items to be specified, i.e., music CDs, and represented using a specific format.

In general, content filtering and collaborative filtering techniques power most current web personalisation systems. Although, these personalisation and recommender systems have been deployed with some success, their wide use has exposed some of their limitations. I.e, [9,10,11,12] have proposed content-based systems which learn user preferences by analysing their accessing patterns and then recommend items specific to the users' interests. Such systems are solely based on matching information within a user's registration document with established user demographic profiles. Limitations include the problem of relying on the subjective input of the online registration process, where the majority of online users prefer browsing websites anonymously and may even fill in false information that leads to inaccurate recommendations. Another clear shortcoming of todays web recommendations is their focus on the users past interests, rather than those of their current browsing session.

3.4 Problem, focus and approach

An important shortcoming for collaborative filtering is the so called "curse of scalability". For these systems, intelligence-intensive tasks need to be performed as online processes in real time. For a large data set, this may lead to unacceptable system resource-consumption and very slow response times. Recent research effort has focussed on this problem. I.e. in [13,16,17], the authors endeavour either to exploit new algorithms or optimise an established approach in order to cope with a high-dimensional data space in an acceptable time frame.

It is clear from the published liturature that although web-based recommendation systems are increasingly common, there are still several problem areas which need to be addressed. Our work is therfore based on the requirement to establish a flexible product recommendation system suitable for search-driven web sites that is able to personalise the web experience for each anonymous visitor. This is achieved through:

1) the alleviation of the "curse of scalability" problem, through combining offline pattern discovery with online pattern matching.
2) the removal of the dependancy on user registration through employing web usage mining techniques in a collaborative filtering system acting upon user's click stream data.

The principal "engine" within our system is a neural network. Neural networks divide into the supervised learning model or the unsupervised learning model where the former needs training samples with known solutions to construct a classification model, and the unsupervised model does not. To enable the predictive modelling of the customer's navigation behaviour, our work uses an unsupervised model to create and train a special type of neural network called a Kohonen network or self-organizing map (SOM) [20,21].

3.5 The self-organising map

In 1981, Tuevo Kohonen proposed and demonstrated a new form of neural network architecture called a self-organizing map (SOM), which has proved extremely useful when the input data is of high dimensionality and complexity. The underlying theory behind the operation of a SOM is covered in [22, 23] where in essence the basic idea is shown to be:

1. "The represention of high-dimensional data in a low-dimensional form without loosing any of the 'essence' of the data." [23], and;
2. "Organisation of data on the basis of similarity by putting entities geometrically close to each other."[23]

The SOM approach is used to discover associations in a data set and clustering of data (i.e. similar expression patterns) where the model creators cannot predict the nature of the classifications, or they feel there may be more than one way to categorise the characteristics of a data set.

4.0 Constructing a Personalised Recommendation System

The objective of our system is to recommend a unique set of objects to satisfy the needs of each active user. The set of recommendations is based on the user's current behaviour, so if a user appears to be searching for hard disc drives on this visit, these are the types of items he will be recommended. This is to avoid irrelevant recommendations based on previous visits to the site, (which can happen with other existing recommendation systems), when, for example, the user may have been interested in printers. The objects recommended could include dynamic links, promotional advertisements or services tailored to each user's preferences. The recommendation engine collects the active user's visit trail (the list of "click" actions taken by the user during their current browsing session) characterises its behaviour and compares it to known patterns of previously classified user group behaviour. The active user is then mapped to a particular user group profile. To make appropriate real time recommendations, these profiles have been predetermined by our system's offline Usage Pattern Discovery Module (Figure 1). The offline usage patterns are discovered through Web usage mining and classified in terms of usage profiles. The profiles provide an aggregate representation of the common activities or interests of each particular user group.

This paper focuses on the structure of the offline Usage Pattern Discovery Module, and its application to an active e-Business, trading in computer products [24]. The module is

tested through comparison with a traditional K-Means clustering model. The following three subsections provide details of the major components of the offline module.

4.1 Data Pre-processing

Data pre-processing is the process of cleaning and transforming raw data sets into a form suitable for web mining. The task of the data pre-processing module is therefore to obtain usable datasets from raw web log files, which, in most cases, contain a considerable amount of incomplete and irrelevant information. The server access log is the main source of data.

A server access log is a text file in which every query made to the web server is recorded. The fields recorded in the log file typically include:

- IP address or domain name;
- user ID and password;
- date and time of the request;
- request, including the request type, query strings, and the protocol;
- cookie, (a general mechanism to both store and retrieve information on the client side of the connection) [25].

In general, data pre-processing in our system includes the following steps, data cleaning and selection, data integration and data transformation. The following subsections provide details of each of these steps in our implementation.

4.1.1 Data Cleaning and Selection

Crude log files always contain a large amount of erroneous, misleading and incomplete attributes which need to be filtered out, including:

- Error requests or requests reset by users
- Requests using "POST" as the method attribute in a browsers HTML <form> tag

- Requests involving image, audio, video and other type of media
- Requests generated by web agents
- Requests generated by proxies

4.1.2 Data integration

Data integration is applied to map the log entries onto semantically meaningful activities. This task is challenging and important work, especially when dealing with search-driven websites. In a search driven web site, client-side components communicate to a web server via encrypted query strings, and the server responds by returning the relevant stem page directly. Consequently, the significance of the page views remains hidden from the web usage log. For example, page views corresponding to a product are considered more significant than others, but such events are missed out from the usage log. Consequently, in our system, the search strings in the web logs are mapped onto corresponding fields of the URL. These search strings are often in the form of a set of product codes that need to be transformed into semantically meaningful names via a link with their product database.

4.1.3 Tasks of Data Transformation

Data transformation plays the most important role in the process of data collection and pre-processing, since, all subsequent data mining tasks are based on the results of this process. A user's navigation behaviour can be depicted as a series of clicks delimited in time sequence and this series of clicks can be defined as a user click stream. Furthermore, the click stream can be separated into a number of smaller sets of clicks with meaningful descriptions, known as sessions. A definition of a session was given by W3C Working Draft (http://www.w3.org/1999/05/WCA-terms/) as: A collection of user clicks to a single web server, also called a visit.

To extract the meaningful sessions, cookies are first used to identify individual users. For each identified user we get a sequence of clicks with a time stamp. These click streams have to be further broken down into a number of sessions to enable subsequent analytical data mining tasks. The approach of identifying sessions by time-out has been well accepted in the research community [8, 26]. If the time interval between two consecutive clicks is less than a predefined threshold, then those two clicks can be considered to be included in one session. In [26], the authors calculated the time-out between every two clicks of one user. Their results indicate that the mean value of those time-outs was 9.3 minutes. By adding 1.5 deviations from the mean, 25.5 minutes was used as the maximum time-out of two adjacent clicks in one session. The authors of [8] also adopted this approach. Their WebSIFT used 30 minutes as the threshold.

Through the processes of data cleaning, selection, integration and transformation, the raw web log is transformed into well-structured user sessions which are ready for subsequent data mining tasks. For a user session, we create an n-dimensional vector over the space of all query strings and all time intervals between each two consecutive searches, as defined in the following subsection.

4.2 Pattern Discovery

When the data pre-processing and transformation steps described in section 4.1 have been completed, we are left with a large number of vectors each representing a different user session. We now need to group users by determining which users have shown similar search behaviour in these sessions. Many different web usage mining techniques can be used for pattern discovery, including clustering, association rules mining, and navigational pattern mining [27]. As explained in section 3.5, we have experimented with neural networks to determine patterns of user behaviour from the pre-processed web logs.

4.2.1 Our Approach

Neural networks have been chosen because for our application they are considered to have advantages over the traditional k-means clustering and k-nearest-neighbour approach. For example, when comparing a Kohonen neural network with a memory-based k-nearest-neighbour (KNN) approach, the neural network enables a clearer visualization that not only has the traditional capacity of clustering data sets into user profile groups, but is also capable of presenting the relationships between the source data and the discovered clusters. The experimental work described in this paper adds to the body of knowledge that supports this argument.

In our work we train a neural network using web log data that describes multiple visitors' browsing activity in retrieving a series of web pages. The trained network is used off-line to discover user group profiles and is then used in real time to examine active user data and make a match to a specific user group.

4.2.2 The SOM Neural Network to Derive Usage Patterns

As explained in section 3.4, we have used a Kohonen network or self-organizing map (SOM) [20, 21], to predictively model the customer's navigational behaviour. This unsupervised model is used to group clusters of queries related to user sessions from a web log, where each cluster represents a group of users with common characteristics. This enables the predictive model to find the web links or products that a current user is potentially interested in.

Like the traditional KNN approach, a SOM neural network also needs input vectors. These are obtained during data preparation and transformation by defining a binary vector for all query strings using one bit per query string.

The input vector can be denoted with the arrays of neuron units, as follows;

For a user session s, $s = < q_1, q_2, \ldots qn, t_1, t_2, \ldots tn\text{-}_1 >$.
Where q1 represents a query (see section 4.1) and t1 represents a time-interval (see section 4.1.3).

Results are in the form of a two-dimensional output grid. Figure 2 illustrates the structure of this two-dimensional Self Organization Map.

During training, each unit on the grid competes with all of the others to "win" each record. When a unit wins a record, its weights (along with those of other nearby units, collectively referred to as a neighbourhood) are adjusted to better match the pattern of predictor values for that record. As training proceeds, the weights on the grid units are adjusted so that they form a two-dimensional "map". Once the learning process is complete, similar inputs will be strengthened from those unit area, thus, these similar inputs can be identified and grouped as user query clusters.

The weight update for an output of j is given by the formula below:

$$W_j(new) = W_j(old) + \alpha [X_n - W_j(old)]$$
$$= \alpha X_n + (1- \alpha) W_j(old)$$

Where:

α is a momentum term used in updating the weights during training to keep the weight changes moving in a consistent direction, normally it is a value between 0 and 1;

Xn represents input vectors, and;

Wj (old) stands for weight before update for an output j.

4.3 Post-processing and Recommendation

The basic procedure for generating recommendations for users is based on a "Top-N Recommendation" approach (also known as a "Most-Frequent Recommendation"). This approach looks at the neighbourhood N of a group of users sharing a common interest, scans through their product-retrieval data and performs a frequency count. The system will sort the results based on the frequency count, and return the N most frequent as recommendations, where their count is greater than a pre-specified threshold. Note that the recommendation list must not include products that the current user has already browsed.

5 System Evaluation

This section, describes the experiments conducted to evaluate our system. In order to evaluate the quality of our SOM statistical model based approach, a traditional K-Means clustering model is also established. The systematic evaluation is based on a performance comparison between these two statistical models which will be referred to as the "hypothesis model" and the "comparison model".

5.1 Evaluating the hypothesis model with three metrics

The experimental approach is based on three metrics that are widely used in the information retrieval community, and are commonly known as the "recall", "precision" and "F1" metrics [28, 29, 6, 30].

We have tuned the definitions of these three metrics to suit the domain of web usage mining.

- Recall: In our system, "recall" is the ratio of the number of patterns correctly identified (size of hit set) over the total test data as follows;

$$\text{recall} = \frac{|test \cap top - N|}{|test|}$$

- Precision: In our system, "precision" is the ratio of the number of patterns correctly identified (size of hit set) over the top N set size as follows.

$$\text{precision} = \frac{|test \cap top - N|}{N}$$

Where the top N set is the top-N recommendations according to the pre-discovered user query clusters.

The hit set is a special set comparing members of the top N set that occur in both the training set and the test set. This method can be found in [6]. There are some drawbacks with only using the two metrics of "recall" and "precision". For example, with increasing N, "recall" will increment, while "precision" will decrement. The "F1-measure" has been used to alleviate these problems through applying the harmonic average of "precision" and "recall" which is defined as follows.

$$\text{F1} = \frac{2 \times (precision \times recall)}{precision + recall}$$

In this study, the number of user queries to be recommended by the top-N recommendation model has been set at 6. The number 6 has simply been chosen as a compromise value to enable a good range of recommendations to be offered to the users, since some types of search may result in many options, whilst others result in very limited options. This can be seen by examining the limited size of some of the clusters shown in Figure 3.

As the hypothesis model may generate more than one cluster and the output on each cluster, in terms of the three measurement metrics may vary, it is reasonable to introduce the average measurement of the three metrics in order to evaluate the overall performance

of the system. Hence, the Mean Recall, Mean Precision, and Mean F1 Measure were used.

5.2 Evaluating the hypothesis Model with statistical accuracy metrics

The metrics for statistical accuracy are used to evaluate the success of our numeric prediction, and in the domain of web usage mining, this numeric prediction can be viewed as comparing the ideal value of a prediction against the actual "hit" scores of the recommendation. The hit score is of the value for comparing members of the top N set that occur in both the training set and the test set, with the ideal value of prediction, which is equal to the top N that is pre-discovered within the training set. In this study, performance is evaluated using the mean absolute error and the correlation coefficient.

- **The mean absolute error (MAE)**

MAE computes the average absolute difference between the actual and predicted numeric output and maintains the dimensionality of the errors without them being affected by large deviations between the actual and predicted numeric output.

The MAE Ei of an individual hypothesis model i is evaluated by the equation:

$$E_i = \frac{1}{n}\sum_{j=1}^{n}\left|P_{(ij)} - T_j\right|$$

where P(ij) is the value predicted by the individual hypothesis model i for the cluster j (where j is one of n clusters pre-discovered in the dataset); and Tj is the actual value for cluster j of the test dataset. Normally, P(ij) is equal to the top N within the training set; and Tj is the actual "hit" score of the recommendation.

For a perfect fit, P(ij) = Tj and Ei = 0. So, the Ei index ranges from 0 to infinity, with 0 corresponding to the ideal.

- **The correlation coefficient**

The correlation coefficient is used to measure the statistical correlation between the actual and predicted numeric output and to explore the consistency correlation between the actual and predicted testing results.

The correlation coefficient Ci of an individual hypothesis model i is evaluated by the equation:

$$C_i = \frac{Cov(T,P)}{\sigma_t \cdot \sigma_p}$$

where Cov(T,P) is the co-variance of the target and hypothesis model outputs; and σ_t and σ_p are the corresponding standard deviations, which are given by the formulas:

$$Cov(T,P) = \frac{1}{n}\sum_{j=1}^{n}(T_j - \overline{T})(P_{(ij)} - \overline{P})$$

$$\sigma_t = \sqrt{\frac{\sum_{j=1}^{n}(T_j - \overline{T})^2}{n}}$$

$$\sigma_p = \sqrt{\frac{\sum_{j=1}^{n}(P_{(ij)} - \overline{P})^2}{n}}$$

where P(ij) is the value predicted by the individual hypothesis model i for the cluster j; and Tj is the actual value for cluster j of the test dataset. Normally, P(ij) is equal to the top N within the training set; and Tj is the actual "hit" score of the recommendation.

$\overline{T}$ and $\overline{P}$ are given by the formulas:

$$\overline{T} = \frac{1}{n}\sum_{j=1}^{n}T_j$$

$$\overline{P} = \frac{1}{n}\sum_{j=1}^{n}P_{(ij)}$$

The correlation coefficient is confined to the range [-1, 1]. When Ci = 1, there is a perfect positive correlation between T and P. When Ci = -1, there is a perfect negative correlation between T and P, that is, they split in opposite ways (when T increases, P decreases by the same amount). When Ci = 0, there is no correlation between T and P. Intermediate values describe partial correlations and the closer to 1 or -1 the better the model.

These two statistical accuracy metrics are considered to be the appropriate performance measures for our work. They enable us to gain a comprehensive measurement of the quality of the work through their complementary nature.

5.3 Data set and Parameter Tuning

- **Data set**

We have evaluated the quality and performance of the top-N recommendations produced by our SOM clustering model, through using a training set and a test set. The data sets were obtained through the server access log provided by our collaborating eCommerce company [24]. The training data-set uses a server access log consisting of 238,676 page clicks, comprising 9,935 unique user's sessions accessing 32,391 content page views. The test data-set uses a server access log consisting of 222,118 page clicks, where 12,447 unique user sessions are identified accessing 33,652 content page views.

- **Parameter setting for the hypothesis model**

The SOM network training is split into two phases. Phase1 is a rough estimation phase, used to capture the gross patterns in the data, and phase2 is a tuning phase, used to adjust the map to model the finer features of the data.

The experimental parameter η is the learning rate, which is a weighting factor that decreases over time, such that the network starts off encoding large-scale features of the data, and, then gradually focuses on finer-level detail. Here, we set the starting value for η at 0.3 for phase1, and 0.1 for phase2 in order to capture finer features. During phase1, η starts at phase1 initial η and decreases to phase2 initial η. During phase2, η starts at phase2 initial η and decreases to 0. In this research, we set 150 cycles for the first phase of training and 30 cycles for the second phase of training. We used an exponential learning rate decay (rather than a linear learning rate decay) during each training cycle to enable a faster training process.

The SOM clustering model was performed on input neuron arrays of size 2,047 from the test dataset and output onto a two-dimensional map with a 5x5 grid (25 nodes). Twenty generic clusters are generated where each cluster represents a group of objects with common characteristics i.e. groups of user sessions showing similar behaviour. The grid size of the output map is tuned during the experimental stage. A final grid size of 5x5 was selected as our experimental results from the particular data set generated twenty clusters. A smaller or larger grid size will result in either missed clusters or wasted training time. The visualisation using SOM allows us to get a good overall view of the clusters, as seen in Figure 3.

The top-N recommendation engine looks into the neighbourhood of each cluster, scans through their test data and performs a frequency count where results are sorted and the 6 most frequently occurring products are returned as recommendations. The top 6 recommendations for the training set were also generated and a hit set was computed based on comparing members of the top N set that occur in both the training set and the test set.

- **the comparison model**

When using the K-means clustering algorithm, an initial number of cluster seeds are required. In order to facilitate a consistent basis for comparison, the number of seeds for the K-means clustering algorithm needs to be the same as the number of clusters generated by the SOM model, in our case this number was twenty.

Through the stages of pattern discovery in the K-means data mining process, nineteen generic clusters are generated. One cluster containing only one candidate has been filtered out. Hence, each of these nineteen clusters can be viewed as a virtual user profile representing common browsing behaviour.

The post-process procedure also needs to be applied in the K-means model to examine the neighbourhood of each cluster, and to perform a frequency count where results are sorted and the 6 top ranked products are selected as recommendations. The top 6 recommendations for the training set are also generated and a hit set can also be obtained by comparing members of the top N set that occur in both the training set and the test set.

5.4 Experimental Results

The size of a cluster can have a significant impact on the recommendation quality; hence we build up our experiment by varying the size of a cluster and validating how efficient the recommendations are by calculating the F1 metrics. The results are illustrated in Figure 4 and Figure 5.

Figure 4 clearly shows that cluster size does influence the top N recommendation quality. The quality drops with increasing cluster size. When the cluster size is under 100 the quality decreases rapidly, it gradually levels and approaches a convergence point. Based on this observation, the optimal size of cluster will be in the range of 100-300.

In general we are looking for evidence that our system can produce useful recommendations based on real time behaviour. To evaluate the performance of our approach we must keep in mind the current internet browsing experience for the user who is looking for a diferent product type to that of his previous visit, which is generally characterised by recommendations that are zero % useful. Consider the results for each cluster related to the three evaluation metrics shown in Table1 and Table 2. The precision rate is considered as the more important metric, reflecting the quality of the recommendations that the system discovered. Based on observations from Table 2, the average precision rate is considered satisfactory, as most of the mean precision rate is of 43.1%, which means users can potentially take advantage of almost three useful recommendations from the six recommended products provided by the system.
In comparison with the precision rate, the recall rate is relatively low, this is due to the small number of N that has been selected and also the size of the test data set which impacts the value of the "recall" rate.

- **Model comparison**

An important issue that affects the evaluation of our work is the degree of benefit derived from the discovered patterns, using the different models. Unpredictable recommendations that could not be derived through common sense, are considered to be the more interesting because it is likely that they will be unexpected to the user. Also, a visit profile is better if the cluster profile is more distinct as the separation reflects a distinct behaviour rather than random selection. In Figure 6, the SOM model can be seen to provide better predictions as it shows the capability of exploiting the dependence among clusters, leading to a wider variety of more interesting predictions. The largest cluster of the K-means model makes up over 50% of the population of the observed classifications (see Figure 7). This reflects

the problem that the K-means model is likely to combine a large number of a user's navigation clicks into one unique group, leading to a limited number of useful predictions. In the extreme case this could lead to a single cluster containing 100% of the population resulting in a single recommendation, thus providing very little interesting advice for the user. In contrast to this, the SOM model tends to keep the size of different clusters balanced (see Figure 6), which will create a better interpretation, capturing wider variations in the user's navigation behaviour.

When comparing the two clustering procedures based on the statistical accuracy metrics, Table 3 shows the results of the two different numeric prediction technologies on a given set, measured using cross-validation. It turns out that the experimental results show the same ordering in terms of prediction quality no matter which statistical accuracy metric is used. Once again the SOM model performs better on this dataset: it leads to a small misclassification error according to mean absolute error (MAE) and a larger correlation coefficient. The value of the correlation coefficient of the SOM model, which scores twice the value of the K-means result, indicates that the SOM model can significantly outperform the K-means model in terms of consistency correlation.

6 Conclusions and Future Work

Our work demonstrates that web usage data mining has strong potential, with the capacity to extract knowledge that can be used for driving personalised recommendation services on the web .

Many approaches to creating personalised web based recommendation systems have already been proposed, however, they lack scalability and capability when dealing with search-driven web sites in real time. Indications are that our approach may overcome

some of these limitations. The advantage of discovering usage patterns offline, can be exploited by scheduling this operation during off peak times. Our online recommendation engine then only needs to collect the active user's click trail data and match this to the discovered offline patterns in order to generate a set of recommendations.

In general, the results in Table3 indicate that adoption of the SOM clustering model can lead to helpful recommendations that out perform the results generated through use of the K-means model.

The proposed system provides a basis for real time personal recommendation whilst overcoming the problem of bottlenecks caused by system computing load when dealing with scaled web sites at peak visiting times.

Our results show that a personalised product recommendation system powered by the SOM predictive model is able to produce, useful recommendations. In most cases the precision rate, or quality of recommendation, is equal to or better than 50%. This figure may apear unimpressive, however, in terms of the application domain it is significant. It means that at least 50% of the products recommended to a user will be in line with his immediate requirements, bringing genuine support to the browsing process rather than a simple reminder of what the user was interested in on his previous visit to the site.

We believe that a framework to underpin the system presented in this paper can be improved by implementing the new technique based on an open and expandable three-tier architecture that is designed for reliability and even greater scalability. This work is currently underway in the Wolfson School at Loughborough.

7.0 References

[1] Mulvenna, M. D., Anand. S, AND Buchner, A. G. 2000. Personalization on the net using web mining. Commun. ACM, 43, 8 (August), pp. 123-125.

[2] Schafer, J, B. Konstan, J A and Riedl, J, (2001). "E-commerce recommendation application," Data Mining and Knowledge Discovery, vol. 5, no. 1/2, pp. 115-153.

[3] Anderson, C R,(2003). "A Machine Learning Approach to Web Personalization" PHD Thesis by University of Washington, 2003.

[4] Basu, C. Hirsh, H and Cohen, V, (1998), "Recommendation as classification: Using social and content-based information in recommendation," Proc. of the Fifteenth National Cont.. on Artificial Intelligence, pp. 714-720, 1998.

[5] Resnick, P. Iacovou, N. Sushak, M. Bergstrom, P. and Riedl, J, (1994) "Grouplens: An open architecture for collaborative filtering of Netnews," Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work, pp. 175-186, 1994.

[6] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of Recommendation Algorithms for E-Commerce. Paper presented at the ACM E-Commerce 2000 Conference, pp.158-167

[7] Shardanand, U. and Maes, P. (1995). "Social information filtering: Algorithms for automating word of mouth," Proc. of the Annual ACM SIGCHI on Human Factors in Computing Systems, pp. 210-217, 1995.

[8] Cooley, R., Tan, P, N, and Srivastava, J. (2000). Discovery of interesting usage patterns from web data. International Workshop on Web Usage Analysis and User Profiling pp.163-182 ISBN:3-540-67818-2

[9] Balabanovic, M, (1997), "An adaptive Web page recommendation service," Proc. of the First Int. Conf. on Autonomous Agents, pp. 378-385, 1997.

[10] Lieberman, H. Van Dyke N. W and Vivacqua, A. S, (1999) "Let's Browse: A collaborative browsing agent," Knowledge-Based Syst. vol. 12, no. 8, pp. 427-431, 1999.

[11] Mooney R. J and Roy, L, (2000). "Content-based book recommending using learning for text categorization," Proceedings of the Fifth ACM Conference on Digital Libraries, pp. 195-204, 2000.

[12] Frias-Martinez, E., Magoulas G., Chen, S. & Macredie, R. (2006). Automated user modeling for personalized digital libraries. International Journal of Information Management, 26, 3, pp. 234-248.

[13] Mobasher, B., Cooley, R., and Srivastava, J. (1999). Creating adaptive web sites through usage-based clustering of URLs. In Proceedings of the third IEEE International Knowledge and Data Engineering ExchangeWorkshop, Chicago, Illinois, pp.19-25.

[14] Canny, J. (2002), “Collaborative Filtering with Privacy via Factor Analysis”. Paper presented at the 25th annual International ACM SIGIR conference on Research and Development in Informaiton Retrieval, Tampere, Finland pp.238-245.

[15] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 41(6), pp. 391-407.

[16] Sarwar, B. M. Karypis, G. Konstan, J. A. and Riedl, J. T, (2001) "Item-based collaborative filtering recommendation algorithms," Proceedings of the 10th International World Wide Web Conference., pp. 285-295, 2001.

[17] Demiriz, A. (2002). Enhancing Product Recommender Systems on Sparse Binary Data. Demiriz, A. Available: http://www.rpi.edu/~demira/productrecommender.pdf 2002.

[18] Jansen, B. J.(2006) Search log analysis: What it is, what's been done, how to do it: Library & Information Science Research 28 (2006), pp. 407–432

[19] Shahabi, C. Banaei-Kashani, F. Chen, Y.-S and McLeod, D, (2001) "Yoda: An accurate and scalable Web-based recommendation system," Proceedings of the 9th International Conference on Cooperative Information Systems, pp.418-432, September 05-07, 2001

[20] Kohonen T, (1981). Construction of similarity diagrams for phonemes by a self-organizing algorithm, Technical Report TKK-FA463, Helsinki University of Technology, Espoo, Finland 1981.

[21] Kohonen T, (1982). Self-organized formation of topologically correct feature maps, Biological Cybernetics 43 1982 pp.59-69.

[22] Willamette,(2006),http://www.willamette.edu/~gorr/classes/cs449/Unsupervised/SOM.html, viewed Feb. 2nd, 2006.

[23] UCL, (2006), http://www.ucl.ac.uk/oncology/MicroCore/HTML_resource/SOM_Intro.htm, viewed Feb. 2nd, 2006

[24] A2Z (2005) , http://www.a2zcomputerproducts.com, data acquired Jan 2005.

[25] Netscape, 1999. Support documents PERSISTENT CLIENT STATE HTTP COOKIES Preliminary Specification - Use with caution http:// home.netscape.com/newsref/std/cookie_ spec .html, 1999.

[26] Catledge, L. D, Pitkow, J. (1995). “Characterizing Browsing Strategies in the World-Wide Web”, Computer Networks and ISDN Systems 27(6): 1065-1073, 1995.

[27] Chang, G, Healey, M,J, McHugh, J. M, Wang, J,T.L. (2001),"Mining the World Wide Web An Information Search Approach" ISBN 0-7923-7349-9 pp. 93-104, 2001.

[28] Yang, Y., & Liu, X. (1999). A Re-examination of Text Categorization Methods. The 22th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99) pp. 42-49.

[29] Yang, Y (2001),"A study on threshold strategies for text categorization", Proc. of the Nth ACM Int. Conf. on Research and Development in Information Retrieval, pp. 137-145.

[30] Shih Y & Liu R .(2005). "Hybrid recommendation approaches: collaborative filtering via valuable content information" Proceedings of the 38th Hawaii International Conference on System Sciences, p.217b, 2005.

1. Figures and Tables

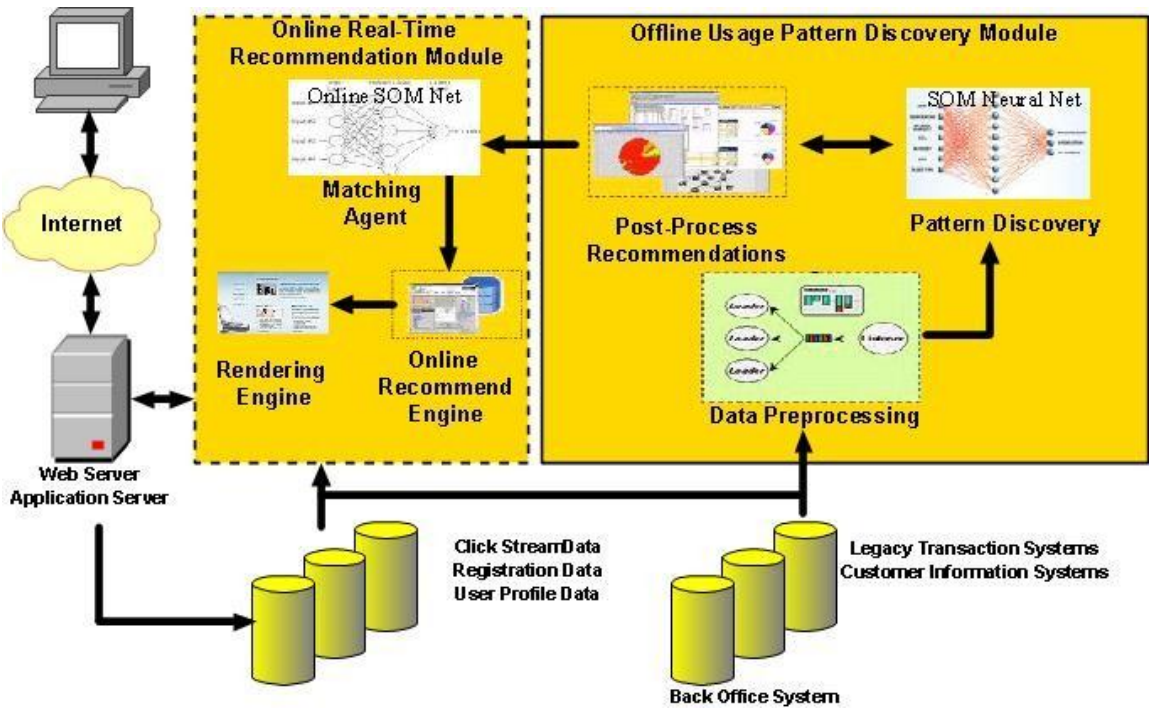

Figure 1 The architecture and process flow within the overall system

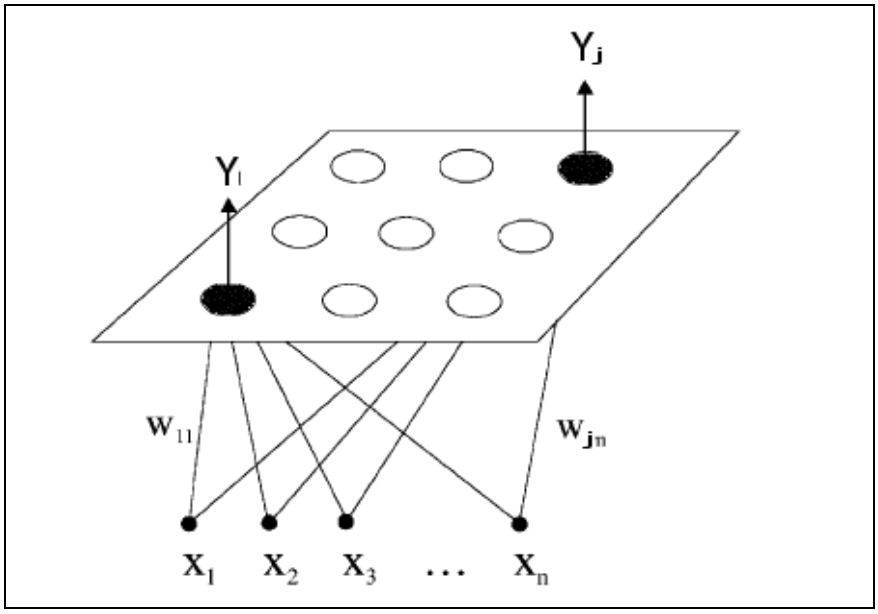

Figure 2 Kohonen Self-Organizing Map in Two-Dimensions

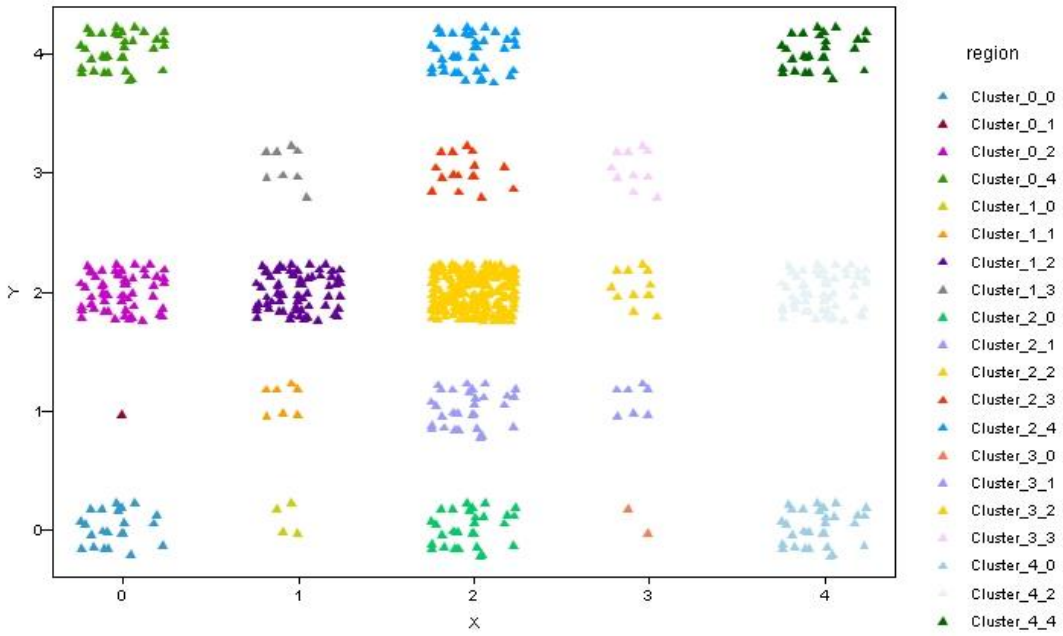

Figure 3: Layout of the off-line generated user profile clusters

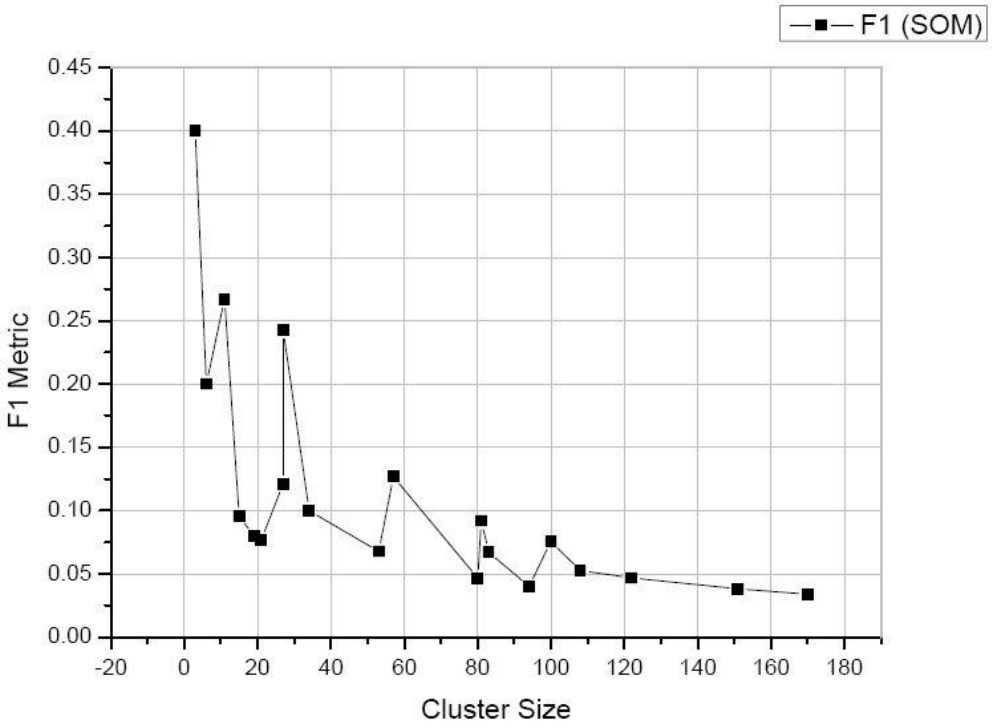

Figure 4: Impact of cluster size on recommendation quality based on SOM model

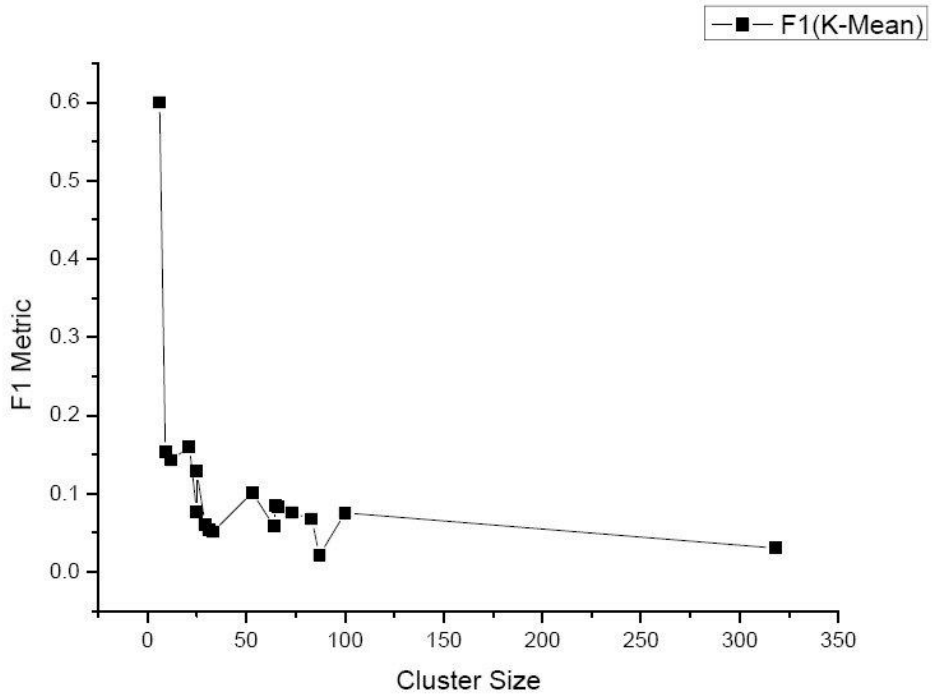

Figure 5: Impact of cluster size on recommendation quality based on K-means

Region	Precision	Recall	F1
Cluster_0_0	33.33%	3.77%	0.068
Cluster_0_1	50.00%	33.33%	0.400
Cluster_0_2	50.00%	2.46%	0.047
Cluster_0_4	66.67%	4.00%	0.075
Cluster_1_0	50.00%	18.18%	0.267
Cluster_1_1	16.67%	6.67%	0.095
Cluster_1_2	33.33%	2.50%	0.047
Cluster_1_3	16.67%	5.26%	0.080
Cluster_2_0	66.67%	4.94%	0.092
Cluster_2_1	66.67%	7.02%	0.127
Cluster_2_2	50.00%	1.76%	0.034
Cluster_2_3	33.33%	5.88%	0.100
Cluster_2_4	50.00%	2.78%	0.053
Cluster_3_0	25.00%	16.67%	0.200
Cluster_3_1	20.00%	4.76%	0.077
Cluster_3_2	33.33%	7.41%	0.121
Cluster_3_3	66.67%	14.81%	0.242
Cluster_4_0	33.33%	2.13%	0.040
Cluster_4_2	50.00%	1.99%	0.038
Cluster_4_4	50.00%	3.61%	0.067

Table 1: Performance of the trained SOM model on the test dataset

Model Metrics	K Means Model	SOM Model
Mean Precision	40.30%	43.10%
Mean Recall	7.40%	7.50%
Mean F1	0.111	0.114

Table 2: Performance comparison based on average three-metrics criteria

Model Numeric Prediction Metrics	K-Means Model	SOM Model
Mean absolute error (MAE)	3.316	3.15
Correlation coefficient	0.218	0.485

Table 3: Performance comparison based on statistical accuracy metrics

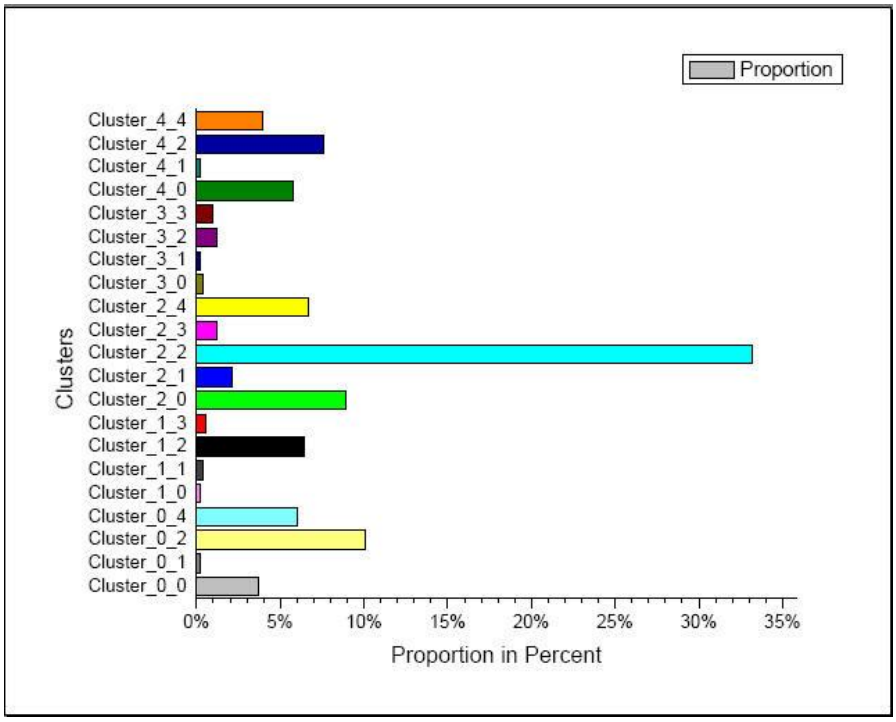

Figure 6: Distribution of cluster proportion for SOM model

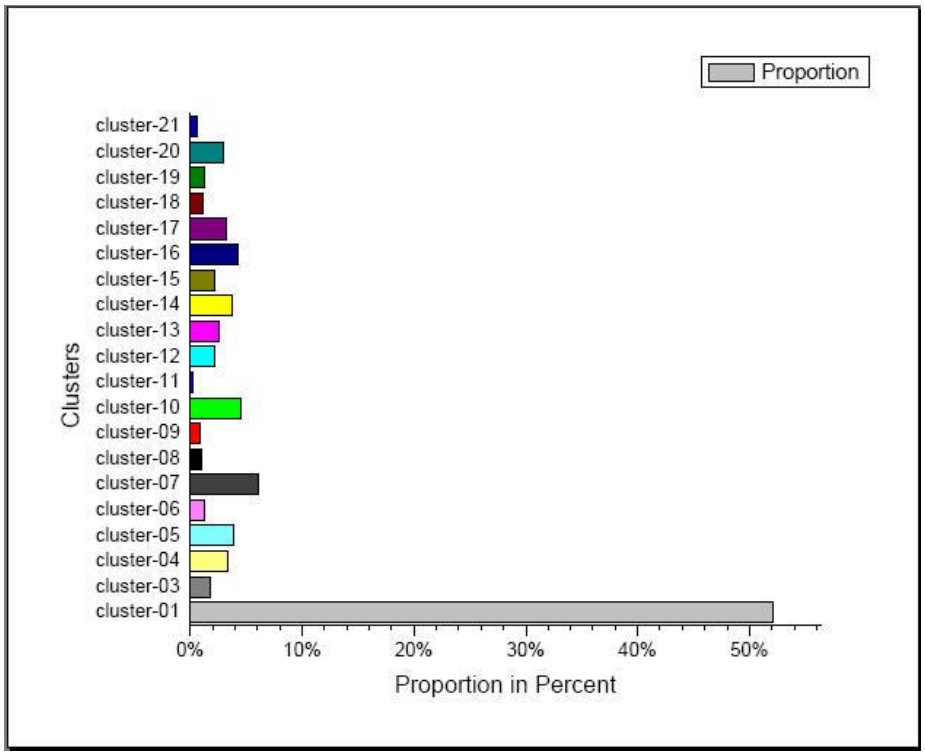

Figure 7: Distribution of cluster proportion for K-means model

.