Applying Modified Fuzzy Neural Network to Customer Classification of E-Business

Yukun Cao[1], Yunfeng Li[1], and Xiaofeng Liao[1]

[1] Department of Computer Science, Chongqing University, Chongqing, 400044, P.R. China
marilyn_cao@163.com, lyf129@126.com

Abstract. With the increasing interest and emphasis on customer demands in e-commerce, customer classification is in a crucial position for the development of e-commerce in response to the growing complexity in Internet commerce logistical markets. As such, it is highly desired to have a systematic system for extracting customer features effectively, and subsequently, analyzing customer orientations quantitatively. This paper presents a new approach that employs a modified fuzzy neural network based on adaptive resonance theory to group users dynamically based on their Web access patterns. Such a customer clustering method should be performed prior to Internet bookstores as the basis to provide personalized service. The experimental results of this clustering technique show the promise of our system.

1 Introduction

In the recent years, the on-going advance of Internet and Web technologies has promoted the development of electronic commerce. Enterprises have been developing new business portals and providing large amount of product information in order to expand their markets and create more business opportunities. In addition to providing a new channel, e-commerce (EC) has the potential of serving customer better if it can take good advantage of information technology and develop EC-specific marketing strategy. Ec, not just the purchase of goods and services over the Internet, is a broad term. It encompasses all electronically conducted business activities, operations, and transaction processing. With the development of the Inter and EC, companies have changed the way they connect to and deal with their customers and partners. Businesses hence could overcome the space and time barriers and are now capable of serving customers electronically and intelligently. New enterprises should manage a particular customer's Web experiences by customer personalization and retain the communication or interaction with the customer. Such understanding of customers can be applied to transform customer information into quality services or products.

In the approach, we presents a framework that dynamically groups users according to their Web access and transactional data, which consist of the customers' behavior on web site, for instance, the purchase records, the purchase date, amount paid, etc. The proposed system is developed on the basis of a modified fuzzy ART neural network, and involves two sequential modules including: (1) trace customers' behavior on web site and generate customer profiles, (2) classify customers according customer profile using neural network. In second module, we employ a modified fuzzy

X. Deng and Y. Ye (Eds.): WINE 2005, LNCS 3828, pp. 356 – 365, 2005.

ART, that is a kind of adaptive resonance theory neural network, for the following reasons. Similar to other neural network strategies, it can plastically adapt to such complex (often uncertain or inconsistent) and correlated (non-linear and not isolated) situations in market analysis rather than those linear functions such as K-means clustering model. And some of the methods like self-organizing map algorithm is suitable for detailed classification, rather than preliminary clustering, such as customer analysis. And some self-organizing map algorithms need to specify the expected number of clusters in advance, which may affect the clustering results due to subjective parameter setting.

The remainder of the paper is organized as follows. Section 2 presents the relate work. In Section 3, the framework to automatically extract user preference and recommend personalized information is expatiated in detail. Section 4 presents three classifier used in our experiments briefly. Implementation issues and the results of empirical studies are presented in Section 5. Finally, the conclusion can be found in Section 6.

2 A Customer Cluster Framework

In this section, an on-line customer cluster framework is presented, which is performed prior to an Internet bookstore in our experiment. The main idea is that customer preference could be extracted by observing customer behavior, including the transaction records, the transaction time and the products pages customer browsed. Then the result of first module is organized in a hierachical structure and utilized to generate customer profile respectively. Finally customer profile could be grouped into different teams using modified fuzzy ART neural network. The framework includes three modules: customer behavior recording, customer profile generating and customer grouping.

2.1 Customer Behavior Recording

Most personalization systems gather customer preference through asking visitors a series of questions or needing visitors rating those browsed web pages. Although relevance feedback obtained directly from customers may make sense, it is troublesome to customers and seldom done. And since customer interests often change likely, it is important to adjust the user preference profile incrementally. Although relevance feedback is effective, customers are overloaded. In the paper, we present a customer behavior recording module to collect the training data without user intervention through tracking the customers behavior on a e-commerce web site. In the paper, the customer behavior is divided two types: transaction record and customer operation. The transaction record includes the type and number of products customer purchased. The customer operation on product pages or images includes the browsing time, the view frequency, saving, booking, clicking hyperlinks, scrolling and so on.

According to some relate works, visiting duration of a product pages or images is a good candidate to measure the preference. Hence, in our work, each product page or image, whose visiting time is longer than a preset threshold (e.g. 30 seconds), is analyzed and rated. Periodically (e.g. every day), the module analyzes the activities of the previous period, whose algorithm is shown as follows:

```
BEGIN
For each product category Pi a customer browsed
{ if (Pi doesn't exist in customer log file)
       {favorite(Pi)=0;}
  for each (transaction record on Pi)
       {favorite(Pi)= favorite(Pi)+transaction-number*0.03;}
  While (browsing time of product page or image belonging to
Pi) > threshold
       { switch (happened operation)
            {case (saving, booking operation happened):
                 favorite(Pi)= favorite(Pi)+0.02;break;
              default: favorite(Pi)= favorite(Pi)+0.01;break;}
       }
}
updating favorite(Pi) in Customer Log file;
END
```

where the function *favorite*(P_i) measures the favorite degree of a certain product category in e-commerce web site, and the record in customer log file is shown as follows: *product-id, category, favorite*. The *category* element is the category path of a resource, what is a path from the root to the assigned category according the hierarchical structure of Internet bookstore. For example, in a Internet bookstore, "JavaBean" category is a subclass of "Java" category, "Java" category is a subclass of "Programming" category, and "Programming" category is a subclass of "Computer & Internet" category, then the category path of the product pages or images belonging to "JavaBean" is "/JavaBean/Java/Programming/Computer&Internet".

2.2 Customer Profile Generating

In this approach, we employ a tree-structured scheme to represent customer profile, with which customers specify their preference. Generator could organize customer preference in a hierarchical structure according the result of Recorder and adjust the structure to the changes of customer interests. Customer profile is a category hierarchy where each category represents the knowledge of a domain of user interests, which could easily and precisely express customer's preference. The profile enhances the semantic of user interests and is much closer to a human conception. The logical structure of the preference tree is shown as follows:

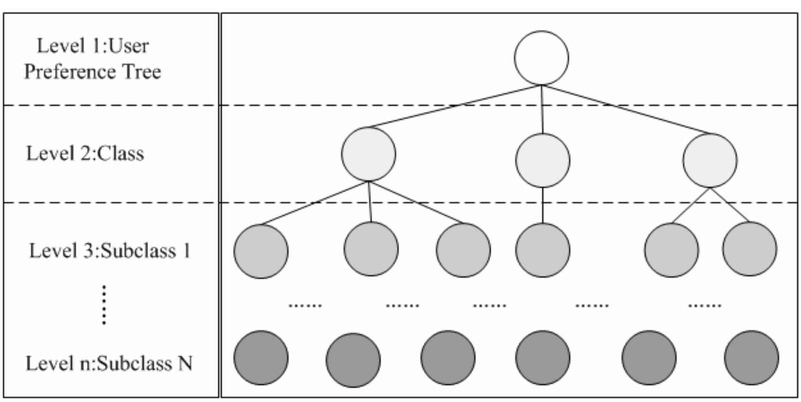

Fig. 1. The Logical Structure of Customer profile

Customer profile is established according the hierarchical structure of a certain e-commerce web site. It means the number of levels and the categories in the profile are similar with the web site. Each node in the tree, representing a category might be interested in, is described by an energy value E_i what indicates the favorableness of a product category. E_i controls the life cycle of a category in a profile. The energy increases when customers show interest in the product category, and it decreases for a constant value for a period of time. Relatively, categories that receive few interest will be abstracted gradually and finally die out. Based on the energy values of categories, the structure of customer profile can be modulated as customers interests change. The algorithm is shown as follows:

```
BEGIN
for each (product Pi in customer log file f)
{ inserting(Pi);
  if ( Energy Ei of Pi) >1 { Ei =1; }
}
if (the days from the last updating) > threshold {updat-
ing(f)}
END
```

To construct customer profile, we employ two Functions: *inserting* and *updating*. The inserting operation is utilized to insert new categories into a profile and adjust the energy values of existing categories. The updating operation is utilized to remove those categories customers don't interest anymore. The two operation just like planting a new kind flower and pruning for a plant in a garden. And we must keep the energy value from 0 to 1, what is expected by the modified fuzzy ART neural network.

2.2.1 Inserting

The Customer Log File mentioned in section 3.1 is considered as the basis of inserting operation to construct the customer profile. For each product in a log file, we first check if the category of the product exists in the preference tree. If the category exists, the *energy* value of the category should be refreshed. If the category does not exist, we will create the category in customer profile, whose *energy* is the value of *favorite*. Then the *energy* value of the new node should be calculated. The following method is used to calculate the new energy value of each category:

$$E_i = \frac{\sum_{p \in P_i^{new}} W_{p,i}}{\left|P_i^{new}\right|} + \lambda \times E_i \tag{1}$$

where E_i is the energy value of product category C_i, P_i^{new} is the set of the products assigned to the category C_i in customer log file , the absolute value $\left|P_i^{new}\right|$ is the number of products in P_i^{new} , and $W_{p,i}$ is the *favorite* of the product *p*. The parameter λ, called *decaying factor*, is set from 0 to 1, hence the older records have less effects to the representation of category. In our experiment, λ is assigned to 0.98.

2.2.2 Updating

Since customer interests often change, it is important to adjust the customer profile incrementally, in order to represent customer interests accurately. In discussion of the changes of customer interests, it is found that there are two types of the user interests. One is the long-term interest and the other is the short-term interest. The long-term interest often reflects a real user interest. Relatively, the short-term interest is usually caused by a hot products event and vanishes quickly. The updating operation is designed to adjust the part reflecting customer short-term interests.

In contrast to the inserting operation that adds the new interesting categories into customer profile, the updating operation is the mechanism to remove the out-of-favor categories. Categories with a continual attention can continuously live, otherwise, they will become weak and finally die out. In customer preference, every category's energy value should be reduced a predefined value ψ periodically (e.g. 15 days). The parameter ψ, called *aging factor*, is used to control the reduction rate.

When no or few products browsed in a category, its energy value will decline gradually. If a category's energy value is less than (or equal to) a pre-defined threshold, we remove the category from user preference tree. To keep a personal view on part to the trend of customer interest, categories with low energy value are removed.

2.3 Customer Cluster

Customer cluster could group customers into different teams according their profiles using adaptive neural network. Nowadays, there are various approaches to cluster analysis, including multivariate statistical method, artificial neural network, and other algorithms. However, some of the methods like self-organizing map algorithm implies some constraints: the need to choose the number of clusters a priori, heavier computational complexity and merging the groups representing the same cluster, because the SOM, by approximating the distribution patterns, finds more than one group representing the same cluster. Moreover, successive SOM results depend on the training phase and this implies the choice of representative training examples. For this reason, we employ a modified fuzzy ART, one of the clustering methods using neural network, for cluster analysis.

The Fuzzy ART [9] network is an unsupervised neural network with ART architecture for performing both continuous-valued vectors and binary-valued vectors. It is a pure winner-takes-all architecture able to instance output nodes whenever necessary and to handle both binary and analog patterns. Using a 'Vigilance parameter' as a threshold of similarity, Fuzzy ART can determine when to form a new cluster. This algorithm uses an unsupervised learning and feedback network. It accepts and input vector and classifies it into one of a number of clusters depending upon which it best resembles. The single recognition layer that fires indicates its classification decision. It the input vector does not match any stored pattern, it creates a pattern that is like the input vector as a new category. Once a stored pattern is found that matches the input vector within a specified threshold (the vigilance $\rho \in [0,1]$), that pattern is adjusted to make it accommodate the new input vector. The adjective fuzzy derives from the functions it uses, although it is not actually fuzzy. To perform data clustering, fuzzy ART instances the first cluster coinciding with the first input and allocating new groups when necessary (in particular, each output node represents a cluster from a group). In the paper, we employ a modified Fuzzy ART proposed by Cinque al. [10] to solve some problems of traditional Fuzzy ART [10,11]. The algorithm is shown as follows:

```
BEGIN
For each (input vector Vi)
{ for each (exist cluster Ci) {C*=argmax(choice(Ci,Vi));}
  if match(C*,Vi)≥ρ {adaptation(C*,Vi);}
  else { Instance a new cluster; }
}
END
```

Function *choice* used in the algorithm is the following:

$$choice\ (C_j, V_i) = \frac{(|C_j \wedge V_i|)^2}{|C_j| \cdot |V_i|} = \frac{(\sum_{r=1}^{n} z_r)^2}{\sum_{r=1}^{n} c_r \cdot \sum_{r=1}^{n} v_r} \tag{2}$$

It computes the compatibility between a cluster and an input to find a cluster with greatest compatibility. The input pattern V_i is an n-elements vector transposed, C_j is the weight vector of cluster J (both are n-dimensional vectors). " $\wedge$ " is fuzzy set intersection operator, which is defined by:

$$\begin{aligned} &x \wedge y = \min\{x, y\} \\ &X \wedge \mathrm{Y} = (\mathrm{x}_1 \wedge y_1, \cdots, \mathrm{x}_n \wedge y_n) = (\mathrm{z}_1, \mathrm{z}_2, \cdots, \mathrm{z}_n) \end{aligned} \tag{3}$$

Function *match* is the following:

$$match\ (C^*, V_i) = \frac{|C^* \wedge V_i|}{|C^*|} = \frac{\sum_{r=1}^{n} z_r}{\sum_{r=1}^{n} c_r^*} \tag{4}$$

This computes the similarity between the input and the selected cluster. The *match* process is passed if this value is greater than, or equal to, the vigilance parameter $\rho \in [0,1]$. Intuitively, ρ indicates how similar the input has to be to the selected cluster to allow it to be associated with the customer group the cluster represents. As a consequence, a greater value of ρ implies smaller clusters, a lower value means wider clusters.

Function *adaptation* is the selected cluster adjusting function, which algorithm is shown as following:

$$adaptation\ (C^*, V_i) = C^*_{new} = \beta(C^*_{old} \wedge V_i) + (1-\beta)C^*_{old} \tag{5}$$

Where the learning parameter $\beta \in [0,1]$, weights the new and old knowledge respectively. It is worth observing that this function is not increasing, that is $C^*_{new} < C^*_{old}$.

In the study, the energy values of all leaf nodes in a customer profile consist an *n-elements* vector representing a customer pattern. Each element of the vector represents a product category. If a certain product category doesn't include in customer profile, the corresponding element in the vector is assigned to 0. Pre-processing is required to ensure the pattern values in the space [0,1], as expected by the fuzzy ART.

3 Other Classifiers Used in Our Experiments

To verify our proposed system, we built traditional fuzzy ART and SOM classifier. In this section, these classifiers are briefly described.

3.1 Traditional Fuzzy ART

Adaptive resonance theory (ART) describes a family of self-organizing neural networks, capable of clustering arbitrary sequences of input patterns into stable recognition codes. Many different types of ART networks have been developed to improve clustering capabilities, including ART1, ART2, ART2A, and fuzzy ART etc. The modified fuzzy ART presented in the paper is similar with traditional fuzzy ART, but employs different *choice* function. The choice function utilized in traditional fuzzy ART is as following:

$$choice\ (C_j, V_i) = \frac{|C_j \wedge V_i|}{\alpha + |V_i|} = \frac{(\sum_{r=1}^{n} z_r)}{\alpha + \sum_{r=1}^{n} v_r} \tag{6}$$

Where α is choice parameter providing a floating point overflow. Simulations in this paper are performed with a value of $\alpha \approx 0$.

3.2 Self-organizing Maps

The self-organizing maps or Kohonen's feature maps are feedforward, competitive ANN that employ a layer of input neurons and a single computational layer. Let us denote by y the set of vector-valued observations, $y = [y_1, y_2, ..., y_m]^T$, the weight vector of the neuron j in SOM is $w_j = [w_{j1}, w_{j2}, ..., w_{jm}]^T$. Due to its competitive nature, the SOM algorithm identifies the best-matching, winning reference vector w_i (or winner for short), to a specific feature vector y with respect to a certain distance metric. The index i of the winning reference vector is given by:

$$i(y) = \arg\min_j \{ \| y - w_j \| \},\ j = 1, 2, ..., n \tag{8}$$

where n is the number of neurons in the SOM, $\|\cdot\|$ denotes the Euclidean distance. The reference vector of the winner as well as the reference vectors of the neurons in its neighborhood are modified using:

$$w_i(t+1) = w_i(n) + \Lambda_{i,j}(t)[x(t) - w_i(t)]\ ,\ t = 1,2,3,... \tag{9}$$

Where $\Lambda_{i,j}(t)$ is neighourhood function, and t denotes discrete time constant. The neighbourhood function $\Lambda_{i,j}$ used in equation (9), is a time decreasing function which determines to which extent the neighbours of the winner will be updated. The extent of the neighbourhood is the radius and learning rate contribution, which should both decrease monotonically with time to allow convergence. The radius is simply the maximum distance at which the nodes from the winner are affected. A typical smooth Gaussian neighbourhood kernel is given bellow in equation (10).

$$\Lambda_{i,j}(t) = \alpha(t) \cdot \exp\left(- \frac{\| r_i - r_j \|^2}{2\sigma(t)} \right) \tag{10}$$

where $\alpha(t)$ is the learning rate function, $\sigma(t)$ is the kernel width function, $\| r_i - r_j \|^2$ is the distance of BMU i unit to current unit j. There are various functions used as the learning rate $\alpha(t)$ and the kernel width functions $\sigma(t)$. For further details about the SOM please refer to [12] and [13].

4 Experiment

In the experiment, we construct an experimental web site and the proposed framework utilizing Java servlet and Java bean. The trial simulated 15 customers behavior on an experiment Internet bookstore over a 30-day period, and they were pre-grouped 4 teams. The experimental web site is organized in a 4-level hierarchy that consists of 4 classes and 50 subclasses, including 5847 book pages and images obtained from www. Amazon.com. As performance measures, we employed the standard information retrieval measures of recall (r), precision (p), and F1(*F1=2rp/(r+p)*).

The trial results were compared with the clustering algorithm performed by SOM and traditional fuzzy ART. All the experiments were conducted by limiting human interaction to adjust parameter with intuitive effects. Comparisons were made in the following context. Traditional fuzzy ART was simulated by an original implementation. It was used in the fast learning asset (with β =1) with α set to zero. Values for the vigilance parameter ρ were found by trials. In the simulation of *k*-means, parameter *K* representing the number of clusters is assigned to 7 by trials. In particular, we used a rectangular map with two training stages: the first was made in 750 steps, with 0.93 as a learning parameter and a half map as a neighborhood, and the second in 400 steps, with 0.018 as a learning parameter and three units as a neighborhood. Map size was chosen by experiments. In the proposed system, decaying factor λ is assigned to 0.95, aging factor ψ is set to 0.03, β is set to 1, and vigilance parameter ρ is assigned to 0.90 by trials. With the growth of vigilance parameter, the amount of clusters is increased too. Figure 2 shows the increase in the number of clusters with increased vigilance parameter values ranging from 0.85 to 0.97.

Figure 3 illustrates the comparisons of three algorithms mentioned before, including precision, recall and F1. The average for precision, recall and F1 measures using the SOM classifier are 81.7%, 78.3%, 79.9%, respectively. The average for precision, recall and F1 measures using the traditional fuzzy ART classifier are 87.3%, 84.8%, 86%, respectively. And the average for precision, recall and F1 measures using the k-means classifier are 81.6%, 76.9%, 79.2%, respectively. In comparison with the proposed system, the precision, recall, and F1 measures are 92.3%, 88.1%, 90.15%, respectively. This indicates that if the parameters are selected carefully, the proposed framework could group users pattern accurately.

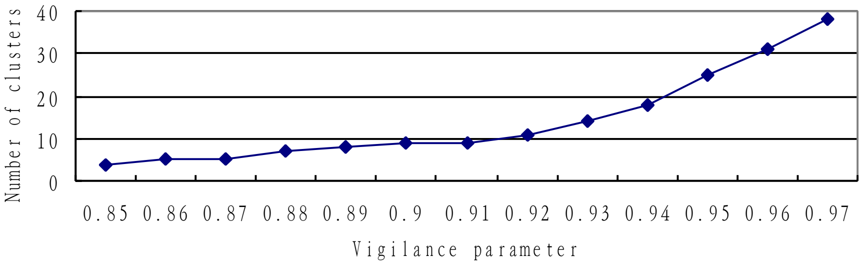

Fig. 2. The vigilance parameter increase with the clusters increasing

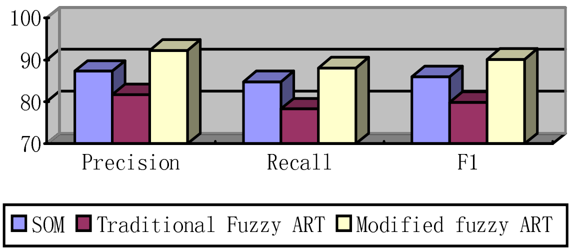

Fig. 3. The comparison of SOM, tranditional ART and modified fuzzy ART algorithm

5 Conclusions

In this paper, we have presented a new framework to automatically track customer access patterns on an Internet commerce web site and group customers using an adaptive neural network. Our approach, essentially based on neural network computation, i.e., learning capacity, satisfies some of its main requirements: fast results, fault and noise tolerance. A pattern grouping module totally independent of the application was also proposed. The cluster system made up of the modified fuzzy ART and the customer pattern track module, was very simple to use. As such system does not use specific knowledge, by adopting the most proper operators, it becomes possible to customize it to different scenarios.

References

1. Brooks C.: Linear and non-Linear forecastability of high-frequency exchange rates, J Forecast, Vol. 16 (1997), pp. 125-145
2. Chen CH, Khoo LP: Multicultural factors evaluation on elicited customer requirements for product concept development, Proceedings of 16th Interneational Conference for Production Research (ICPR-16), July 29-August 3 2001, pp. 15-23
3. Yan W, Chen CH: A radial basis function neural network multicultural factors evaluation engine for product concept development, Expert System, Vol. 18(5), (2001), pp.219-232
4. Cotrell M, Girard B: Forcasting of curves using a Kohonen classification. J Forecast, Vol. 17, (1998), pp. 429-439

5. Curry B, Davies F: The Kohonen self-organizing map: an application to the study of strategic groups in the UK hotel industry, Expert System, Vol. 18(1), (2002), pp. 19-30
6. Sang Chul Lee, Yung Ho Suh: A cross-national market segmentation of online game industry using SOM, Expert Systems with Application, Vol. 27, (2004), pp. 559-570
7. Santosb K. Rangarajan, Vir V. Pboba: Adaptive Neural Network Clustering of Web Users, Computer, Vol. 4, (2004), pp.34-40
8. Tung-Lai Hu, Jiuh-Biing Sheu: A fuzzy-based customer classification method for demand-responsive logistical distribution operations, Fuzzy Sets and System, Vol. 139, (2003), pp. 431-450
9. G.A. Carpenter: Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system, Neural Networks, Vol. 4, (1991), 759-771
10. L. Cinque, G. Foresti: A clustering fuzzy approach for image segmentation, Pattern Recognition, Vol. 37, (2004), pp.1797-1807
11. A. Baraldi, E. Alpaydm: Simplified ART: a new class of ART algorithms, International Computer Science Institute, Berkeley CA,1998
12. Teuvo Kohonen: Self-Organizing Maps, Second Extended Edition. Springer Series in Information Sciences, Vol. 30, Berlin, Heidelberg, New York, (1997)
13. S. Haykin: Neural Networks: A Comprehensive Foundation (2nd Edition). Prentice Hall, (1999)