15

Soft Computing Paradigms for Web Access Pattern Analysis

Xiaozhe Wang[1], Ajith Abraham[2] and Kate A. Smith[1]

[1] School of Business Systems, Faculty of Information Technology, Monash University, Clayton, Victoria 3800, Australia
{catherine.wang,kate.smith}@infotech.monash.edu.au
[2] Department of Computer Science, Oklahoma State University, 700 N Greenwood Avenue, Tulsa, OK 741060700, USA
ajith.abraham@ieee.org

Abstract. Web servers play a crucial role to convey knowledge and information to the end users. With the popularity of the WWW, discovering the hidden information about the users and usage or access pattern is critical to determine effective marketing strategies and to optimize the server usage or to accommodate future growth. Many of the currently available or conventional server analysis tools could provide only explicit statistical data without much useful knowledge and hidden information. Therefore, mining useful information becomes a challenging task when the Web traffic volume is enormous and keeps on growing. In this paper, we propose Soft Computing Paradigms (SCPs) to discover Web access or usage patterns from the available statistical data obtained from the Web server log files. Self Organising Map (SOM) is used to cluster the data before the data is fed to three popular SCPs including Takagi Sugeno Fuzzy Inference System (TSFIS), Artificial Neural Networks (ANNs) and Linear Genetic Programming (LGP) to develop accurate access pattern forecast models. The analysis was performed using the Web access log data obtained from the Monash University's central Web server, which receives over 7 million hits in a week. Empirical results clearly demonstrate that the proposed SCPs could predict the hourly and daily Web traffic volume and the developed TSFIS gave the overall best performance compares with other proposed paradigms.

Key words: Web Mining, Clustering, Self-Organising Map, Hybrid Systems, Fuzzy Logic, Neural Networks, Genetic Programming

15.1 Introduction and Motivation for Research

The World Wide Web is continuously growing with a rapid increase of in-formation transaction volume and number of requests from Web users around the world. To provide Web administrators with more meaningful information for improving the quality of Web information service performances, the discovering of hidden knowledge and information about Web users' access or usage patterns has become a

Xiaozhe Wang et al.: *Soft Computing Paradigms for Web Access Pattern Analysis*, Studies in Computational Intelligence (SCI) **4**, 233–250 (2005)
www.springerlink.com

necessity and a critical task. As such, this knowledge could be applied directly for marketing and management of e-business, e-services, e-searching, e-education and so on.

However, the statistical data available from normal Web log files or even the information provided by commercial Web trackers can only pro-vide explicit information due to the limitations of statistical methodology. Generally, Web information analysis relies on three general sets of information given a current focus of attention: (i) past usage patterns, (ii) degree of shared content and (iii) inter-memory associative link structures [15], which are associated with three subsets of Web mining: (i) Web usage mining, (ii) Web content mining and (iii) Web structure mining. The pattern discovery of Web usage mining consists of several steps including statistical analysis, clustering, classification and so on [16]. Most of the existing research is focused on finding patterns but relatively little effort has been made to discover more useful or hidden knowledge for detailed pattern analysis and predicting. Computational Web Intelligence (CWI) [20], a recently coined paradigm, is aimed at improving the quality of intelligence in Web technology [14].

In order to achieve accurate pattern discovery for further analysis and predicting tasks, we proposed a Self Organizing Map (SOM) [9] to cluster and discover patterns from the large data set obtained from Web log files. The clustered data were further used for different statistical analysis and discovering hidden relationships. To make the analysis more intelligent we also used the clustered data for predicting Web server daily and hourly traffic including request volume and page volume. By using three Soft Computing Paradigms (SCPs) [19] including the Takagi Sugeno Fuzzy Inference System (TSFIS) [17], Artificial Neural Networks (ANNs) [21] and Linear Genetic Programming (LGP) [4], we explored the prediction of average daily request volume in a week (1 to 5 days ahead) and the hourly page volume in a day (1, 12 and 24 hours ahead). Empirical results (analysis and prediction) clearly demonstrate that the proposed SCPs could predict the hourly and daily Web traffic based on the limited available usage transaction data and clustering analysis results.

We explored the Web user access patterns of Monash University's Web server located at http://www.monash.edu.au. We made use of the statistical data provided by "Analog" [3], a popular Web log analyzer that can generate statistical and textual data and information of different aspects of Web users' access log records, as well as weekly based reports include page requests traffic, types of files accessed, domain summary, operating system used, navigation summary and so on. We illustrate the typical Web traffic patterns of Monash University in Fig. 15.1 showing the daily and hourly traffic volume (number of requests and the volume of pages requested) for the week starting 14-Jul-2002, 00:13 to 20-Jul-2002, 12:22. For more user access data logs please refer Monash Server Usage Statistics [13].

In a week, the university's Website receives over 7 million hits. The log files cover different aspects of visitor details like domains, files accessed, requests of daily and hourly received, page requests, etc. Since the log data are collected and reported separately based on different features without interconnections/links, it is a real challenge to find hidden information or to extract usage patterns. Due to the enormous traffic volume and chaotic access behavior, the prediction of the user access patterns becomes more difficult and complex. The complexity of the data volume highlights the need for hybrid soft computing systems for information analysis and trend prediction.

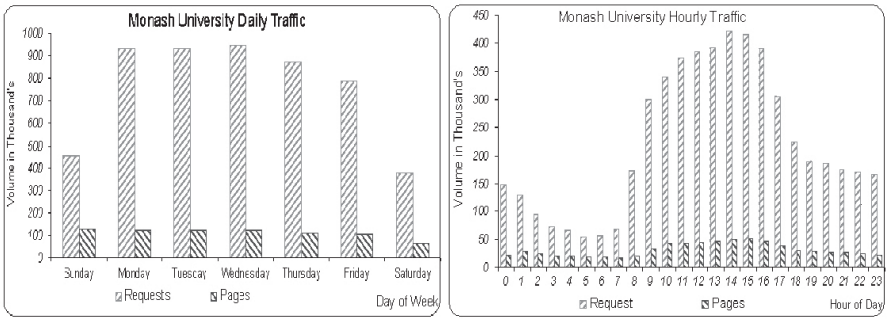

Fig. 15.1. Daily and hourly web traffic patterns of the Monash University main server

In Sects. 15.2 and 15.3, we present a hybrid framework comprising of a SOM to cluster and discover user access patterns. We also employ statistical analysis to mine more useful information using the Web Usage Data Analyzer (WUDA) [18]. In order to make the analysis more intelligent we also used the clustered data to predict the daily request volume and the hourly page volume, as explained in Sect. 15.4. We also pre-sent some basic theoretical background about the three SCPs including TSFIS, ANNs and LGP to predict the usage pattern trends comprising of daily request volume in a week (1 to 5 days ahead) and the hourly page volume in a day (1, 12 and 24 hours ahead). In Sect. 15.5 the experimentation results and comparative performance of the three different soft computing techniques are demonstrated to show the advantages and limitations of each model. Finally, in Sect. 15.6 some conclusions and directions future work are given.

15.2 Web Mining Framework using Hybrid Model

The hybrid framework combines SOM and SCPs operating in a concurrent environment, as illustrated in (Fig. 15.2). In a concurrent model, SOM assists the SCPs continuously to determine the required parameters, especially when certain input variables cannot be measured directly. Such combinations do not optimize the SCPs but only aids to improve the performance of the overall system. Learning takes place only in the SOM and the SCPs remain unchanged during this phase. The pre-processed data (after cleaning and scaling) is fed to the SOM to identify the data clusters. The clustering phase is based on SOM (an unsupervised learning algorithm), which can accept input objects described by their features and place them on a two-dimensional (2D) map in such a way that similar objects are placed close together. The clustered data is then used by WUDA for discovering different patterns and knowledge.

As shown in Fig. 15.3, data X, Y and Z may be segregated into three clusters according to the SOM algorithm. Data X is associated with Cluster 3 strongly only, but data Y and Z have weak associations with the other clusters. For example, data Y is associated with Cluster 2 but can also be considered to have a weak association with Cluster 3. And data Z is associated with both Clusters 2 and 3 even though

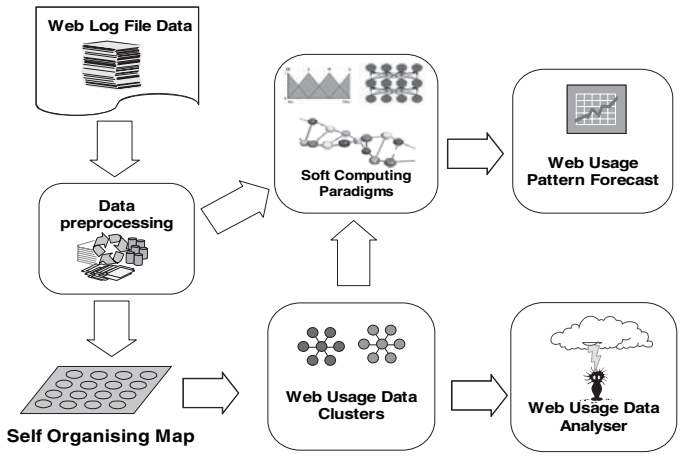

Fig. 15.2. Architecture of the hybrid Web mining model

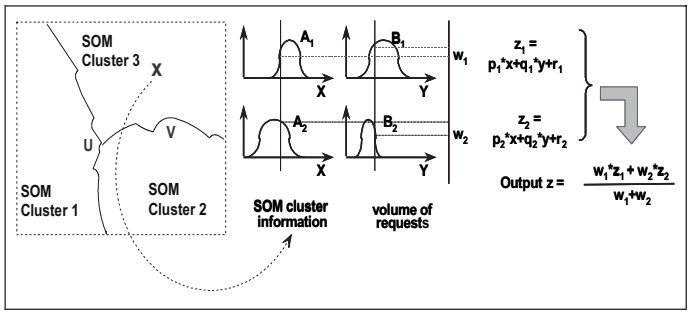

Fig. 15.3. Fuzzy association of data with SOM generated clusters

it is itself within Cluster 1. The degree of association of the data with a particular cluster is modeled as an additional input variable to predict the Web traffic patterns in our pro-posed hybrid system.

Soft computing was first proposed by Zadeh [19] to construct a new generation of computationally intelligent hybrid systems consisting of Neural Networks (NNs), Fuzzy Inference System (FIS), approximate reasoning and derivative free optimization techniques. In this paper, we performed a comparative study of TSFIS, ANNs (trained using backpropagation algorithms) and LGP to predict the hourly and daily Web traffic patterns.

15.3 Web Log Data Clustering and Experimental Analysis using WUDA

Web usage mining normally contains four processing stages including data collection, data preprocessing, pattern discovery and pattern analysis [6]. The data source

selected for our case study is the Web traffic data generated by the "Analog" Web access log analyzer. It is common practice to embed Web trackers or Web log analysis tools to analyze log files for providing useful information to Web administrators. After browsing through some of the features of the best trackers available on the market, it is easy to conclude that other than generating basic statistical data they really cannot provide much meaningful information. In order to overcome the drawbacks of available Web log analyzers, the hybrid approach is proposed to discover hidden information and usage pattern trends which could aid Web managers to improve the management, performance and control of Web servers. In our approach, after selecting the required data from the large data source, all the log data were cleaned, formatted and scaled to feed into the SOM clustering algorithm. The SOM data clusters could be presented as 2D maps for each Web traffic feature, and WUDA was used for detailed user access and usage patterns analysis.

15.3.1 Data Pre-processing

We used the data from 01 January 2002 to 07 July 2002. Selecting useful data is an important task in the data pre-processing stage. After some preliminary analysis, we selected the statistical data comprising the traffic data on the domain on an hourly and daily basis, including request volume and page volume in each data type, to generate the cluster models for finding user access and server usage patterns. To build up a precise model and to obtain more accurate analysis, it is also important to remove irrelevant and noisy data as an initial step in the pre-processing task. Since SOM cannot process text data, any data in text format has to be encoded, according to a specific coding scheme, into numerical format. Further, the datasets were scaled 0–1. Besides the two inputs, "volume of requests (bytes)" and "volume of pages" directly from the original data set, we also included an additional input "time index" to distinguish the access time sequence of the data. The most recently accessed data were indexed higher while the least recently accessed data were placed at the bottom [2]. This is critical because Web usage data has time dependent characteristics.

15.3.2 Data Clustering using SOM

With the increasing popularity of the Internet, millions of requests (with different interests from different countries) are received by Web servers of large organizations. Monash University is a truly international university with its main campus located in Australia and campuses in South Africa and Malaysia. The university has plans to extend its educational services around the globe. Therefore, the huge traffic volume and the dynamic nature of the data require an efficient and intelligent Web mining framework.

In Web usage mining research, the method of clustering is broadly used in different projects by researchers for finding usage patterns or user pro-files. Among all the popular clustering algorithms, SOM has been success-fully used in Web mining projects [10, 12]. In our approach, using high dimensional input data, a 2D map of Web usage pat-terns with different clusters could be formed after the SOM training process. The related transaction entries are grouped into the same cluster and the relationship between different clusters is explicitly shown on the map. We used the Viscovery SOMine to simulate the SOM. All the records after the pre-processing

stage were used by the SOM algorithm and the clustering results were obtained after the unsupervised learning. We adopted a trial and error approach by comparing the normalized distortion error and quantization error to decide the various parameter settings of the SOM algorithm. From all the experiments with different parameter settings, the best solution was selected when minimum errors were obtained.

15.3.3 WUDA to Find Domain Patterns

From the SOM clustering process, five clusters were mapped according to the user access from country of origin or domain. To analyze the difference among the clusters, we have illustrated the comparison of the unique country/domain and the averaged request volume for each cluster in Table 15.1.

Table 15.1. Analysis of request volume for domain cluster map

Cluster Number	Unique Country (or Domain)	Request Volume (Averaged)
1	160	2009.91
2	157	2325.18
3	162	3355.73
4	1 (.au)	199258.93
5	2 (.com & .net)	1995725.00

As evident from Table 15.1, Clusters 4 and 5 are distinguished from the rest of the clusters. Cluster 4 has almost 6 times of the volume of requests compared with the average request volume of the other 3 clusters (Clusters 1, 2 and 3), and Cluster 5 has the maximum number of requests which is nearly 10 times that of Cluster 4. However, by comparing the number of domain countries, Clusters 1, 2 and 3 all have around 150 different domain sources, Cluster 4 contains only Australian domains and Cluster 5 accounts only for *.com and *.net users. The majority of the requests originated from Australian domains followed by *.com and *.net users. This shows that even though Monash University's main server is accessed by users around the globe, the majority of the traffic originates within Australia.

To identify the interesting patterns from the data clusters of Clusters 1, 2 and 3, we have used a logarithmic scale in the Y-axis to plot the time index value in Fig. 15.4 for clarity. For Cluster 1 (marked with "◇"), Cluster 2 (marked with "□") and Cluster 3 (marked with "Δ"), the number of requests of each cluster are very similar and also shared by similar numbers of users from different countries which made it difficult to identify their difference depending on the volume of requests and pages. However, Clusters 1, 2 and 3 can be distinguished with reference to the time of access. So, Clusters 1, 2 and 3 have very similar patterns of access of requests, but their time of access is separated very clearly. Cluster 2 accounts for the most recent visitors and Cluster 3 represents the least recent visitors. Cluster 1 accounts for the users that were not covered by Clusters 2 and 3. Therefore, the different users were clustered based on the time of accessing the server and the volume of requests.

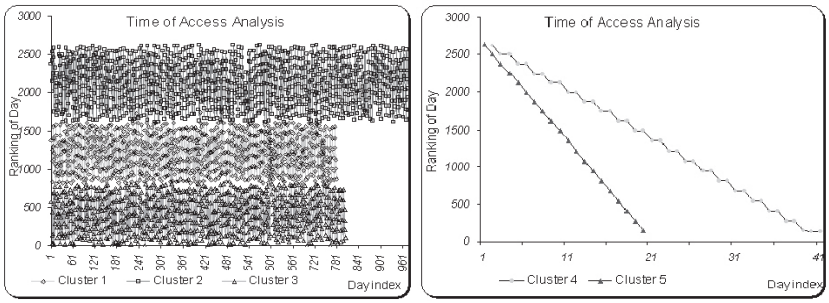

Fig. 15.4. WUDA for time of access in domain cluster map

15.3.4 WUDA to Analyze Hourly Web Traffic Request Patterns

The training process from SOM generated four clusters for the Web traffic requests on the hourly basis. The developed cluster map, indicating the hour of the day when the request was made, is illustrated in Fig. 15.5.

From the developed cluster map depicted in Fig. 15.5, it is very difficult to tell the difference between each cluster, as the requests according to the different hours in a day are scattered. But from Fig. 15.6, it may be concluded that Cluster 2 (marked with "◇") and Cluster 3 (marked with "□") have much higher requests of pages (nearly double) than Cluster 1 (marked with "Δ") and Cluster 4 (marked with "x"). This shows that 2 groups of clusters are separated based on the volume of requests for different hours of the day.

By analyzing the feature inputs of the SOM clusters Fig. 15.6, it is difficult to find more useful information. However, by looking at each hour, as shown in (Fig. 15.7), more meaningful information can be obtained. Clusters 2 and 3 are mainly responsible for the traffic during office hours (09:00–18:00), and Clusters 1 and 4 account for the traffic during the university off peak hours. It is interesting to note that the access patterns for each hour could be analyzed from the cluster results

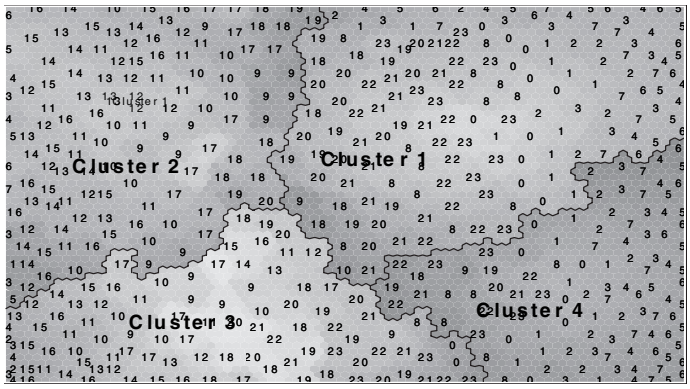

Fig. 15.5. Hourly Web traffic requests cluster map

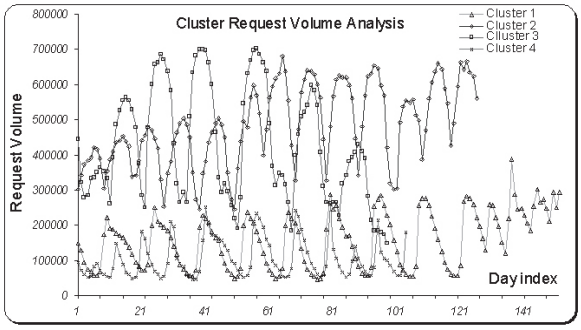

Fig. 15.6. WUDA for request volume in hourly cluster map

with reasonable classification features. By combining the information from Figs. 15.6 and 15.7, the hourly access patterns could be understood very clearly.

15.3.5 WUDA to Discover Daily Requests Clusters

Due to the dynamic nature of the Internet, it is difficult to understand the daily traffic pattern using conventional Web log analyzers. We attempted to cluster the data depending on the total activity for each day of the week using "request volume", "page volume" and "time index" as input features. The training process using SOM produced seven clusters and the developed 2D cluster map is shown in Fig. 15.8.

First, each cluster represents the traffic for only a certain access period by checking the "time index" inputs in each cluster records, and the ranking of the clusters are ordered as 2, 6, 1, 4, 3, 7 and 5 according to the descending order of the access time. In Table 15.2, WUDA reveals that the clusters are further separated according to the day of the week with interesting patterns. Clusters 3 and 6 account for access records which happened during the weekend (Saturday and Sunday). The big group consists of Clusters 1, 2, 4, 5 and 7, which account for the transactions with heavy traffic volume during normal working weekdays (Monday to Friday). With further

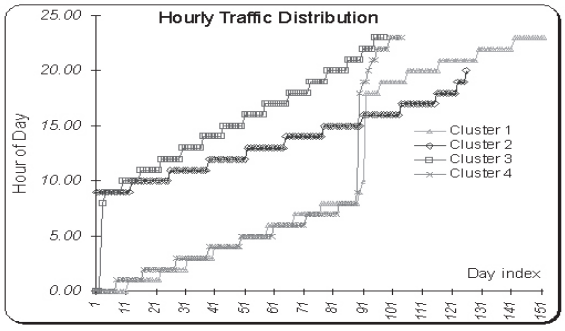

Fig. 15.7. WUDA for hour of day access in hourly cluster map

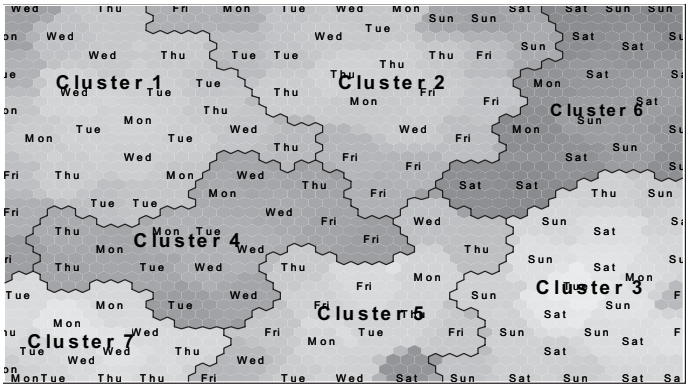

Fig. 15.8. Daily Web traffic requests cluster map

Table 15.2. WUDA for time of access (date) in daily traffic cluster map

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Monday	19.23%	11.54%	4.17%	18.75%	12.50%	11.11%	28.57%
Tuesday	26.92%	15.38%	4.17%	18.75%	12.50%	0.00%	21.43%
Wednesday	23.08%	11.54%	0.00%	31.25%	18.75%	0.00%	21.43%
Thursday	19.23%	15.38%	4.17%	18.75%	18.75%	0.00%	28.57%
Friday	11.54%	30.77%	8.33%	12.50%	31.25%	0.00%	0.00%
Saturday	0.00%	3.85%	37.50%	0.00%	6.25%	50.00%	0.00%
Sunday	0.00%	11.54%	41.67%	0.00%	0.00%	38.89%	0.00%

detailed checking, Clusters 2 and 7 are different from other clusters because Cluster 2 covered heavier traffic on Friday but Cluster 7 missed Friday.

15.4 Soft Computing Paradigms

In contrast with conventional Artificial Intelligence techniques, which only deal with precision, certainty and rigor, the guiding principle of Soft Computing is to exploit the tolerance for imprecision, uncertainty, low solution cost, robustness and partial truth to achieve tractability and better rapport with reality [19]. In this research, we proposed 3 SCPs including TSFIS, ANNs and LGP for developing accurate predicting models for usage pattern trends based on the Website access log data.

15.4.1 Takagi Sugeno Fuzzy Inference Systems (TSFIS)

The world of information is surrounded by uncertainty and imprecision. The human reasoning process can handle inexact, uncertain and vague concepts in an appropriate manner. Usually, the human thinking, reasoning and perception process cannot be expressed precisely. These types of experiences can rarely be expressed or measured using statistical or probability theory. Fuzzy logic provides a framework to

model uncertainty, human way of thinking, reasoning and the perception process. Fuzzy if-then rules and fuzzy reasoning are the backbone of FIS, which are the most important modeling tools based on fuzzy set theory. Fuzzy modeling can be pursued using the following steps:

- Select relevant input and output variables. Determine the number of linguistic terms associated with each input/output variables. Also choose the appropriate family of parameterized membership functions, fuzzy operators, reasoning mechanism, etc.
- Choose a specific type of fuzzy inference system.
- Design a collection of fuzzy if-then rules (knowledge base).

We made use of the Takagi Sugeno Fuzzy Inference Systems in which the conclusion of a fuzzy rule is constituted by a weighted linear combination of the crisp inputs rather than a fuzzy set (Sugeno 1985). A basic TSFIS if-then rule has the following structure, where p_1, q_1 and r_1 are linear parameters:

$$\text{if } x \text{ is } A_1 \text{ and } y \text{ is } B_1, \text{ then } z_1 = p_1 x + q_1 y + r_1 \tag{15.1}$$

A conventional FIS makes use of a model of the expert who is in a position to specify the most important properties of the process. Expert knowledge is often the main source to design FIS. According to the performance measure of the problem environment, the membership functions, rule bases and the inference mechanism are to be adapted. Evolutionary computation [1] and neural network learning techniques are used to adapt the various fuzzy parameters. Recently, a combination of evolutionary computation and neural network learning has also been investigated.

In this research, we used the Adaptive Neuro-Fuzzy Inference System (ANFIS) [11] framework based on neural network learning to fine tune the rule antecedent parameters and a least mean square estimation to adapt the rule consequent parameters of the TSFIS. A step in the learning procedure has two parts. In the first part the input patterns are propagated, and the optimal conclusion parameters are estimated by an iterative least mean square procedure, while the antecedent parameters (membership functions) are assumed to be fixed for the current cycle through the training set. In the second part the patterns are propagated again, and in this epoch, back propagation is used to modify the antecedent parameters, while the conclusion parameters remain fixed. Please refer to [11] for more details.

15.4.2 Artificial Neural Networks (ANNs)

ANNs were designed to mimic the characteristics of the biological neurons in the human brain and nervous system [21]. Learning typically occurs by example through training, where the training algorithm iteratively adjusts the connection weights (synapses). Back propagation (BP) is one of the most famous training algorithms for multilayer perceptrons. BP is a gradient descent technique to minimize the error E for a particular training pattern. For adjusting the weight (w_{ij}) from the ith input unit to the jth output, in the batched mode variant the descent is based on the gradient ∇E ($\frac{\delta E}{\delta w_{ij}}$) for the total training set:

$$\Delta w_{ij}(n) = -\varepsilon * \frac{\delta E}{\delta w_{ij}} + \alpha * \Delta w_{ij}(n-1) \tag{15.2}$$

The gradient gives the direction of error E. The parameters ε and α are the learning rate and momentum respectively.

15.4.3 Linear Genetic Programming (LGP)

LGP is a variant of the Genetic Programming (GP) technique that acts on linear genomes [5]. The LGP technique used for our current experiment is based on machine code level manipulation and evaluation of programs. Its main characteristic in comparison to tree-based GP is that the evolvable units are not the expressions of a functional programming language (like LISP), but the programs of an imperative language (like C). In the automatic induction of machine code by GP [5], individuals are manipulated directly as binary code in memory and executed directly without passing an interpreter during fitness calculation. The LGP tournament selection procedure puts the lowest selection pressure on the individuals by allowing only two individuals to participate in a tournament. A copy of the winner replaces the loser of each tournament. The crossover points only occur between instructions. Inside instructions the mutation operation randomly replaces either the instruction identifier, a variable or the constant from valid ranges. In LGP the maximum size of the program is usually restricted to prevent programs without bounds.

15.5 Daily and Hourly Traffic Patterns Prediction using SCPs

Besides the inputs "volume of requests" and "volume of pages" and "time index", we also used the "cluster location information" provided by the SOM output as an additional input variable. The data was re-indexed based on the cluster information. We attempted to develop SCPs based models to predict (a few time steps ahead) the Web traffic volume on an hourly and daily basis. We used the data from 17 February 2002 to 30 June 2002 for training and the data from 01 July 2002 to 06 July 2002 for testing and validation purposes. We also investigated the daily web traffic prediction performance without the "cluster information" input variable.

15.5.1 Takagi Sugeno Fuzzy Inference System (TSFIS)

We used the popular grid partitioning method (clustering) to generate the initial rule base. This partition strategy requires only a small number of membership functions for each input.

Daily Traffic Prediction

We used the MATLAB environment to simulate the various experiments. Given the daily traffic volume of a particular day the developed model could predict the traffic volume up to five days ahead. Three membership functions were assigned to each input variable. Eighty-one fuzzy if-then rules were generated using the grid based partitioning method and the rule antecedent/consequent parameters were learned after fifty epochs. We also investigated the daily web traffic prediction performance

Table 15.3. Training and test performance for web traffic volume prediction

	Root Mean Squared Error (RMSE)			
	FIS (With Cluster Input)		FIS (Without Cluster Input)	
Predicting Period	Training	Test	Training	Test
1 day	0.01766	0.04021	0.06548	0.09565
2 days	0.05374	0.07082	0.10465	0.13745
3 days	0.05264	0.06100	0.12941	0.14352
4 days	0.05740	0.06980	0.11768	0.13978
5 days	0.06950	0.07988	0.13453	0.14658

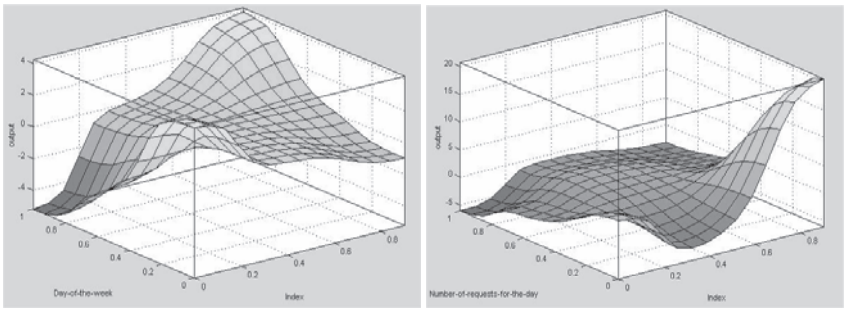

Fig. 15.9. Surface showing day of the week/index and number of requests/index

without the "cluster information" input variable. Table 15.3 summarizes the performance of the fuzzy inference system for training and test data, both with and without cluster information. Figure 15.9 illustrates the learned surface between the output and different input variable (the day of the week/index and number of requests/index).

Figure 15.10 depicts the test results for the prediction of daily Web traffic volume one day, two days, three days, four days and five days ahead.

Hourly Traffic Prediction

Three membership functions were assigned to each input variable. Eighty-one fuzzy if-then rules were generated using the grid based partitioning method and the rule antecedent/consequent parameters were learned after 40 epochs. We also investigated the volume of hourly page requested volume prediction performance without the "cluster information" input variable. Table 15.4 summarizes the performance of the FIS for training and test data.

Figure 15.11 illustrates the test results for 1 hour, 12 hours and 24 hours ahead prediction of the volume of hourly page volume.

From Tables 15.3 and 15.4 it is evident that the developed TSFIS could predict the patterns for several time steps ahead even though the models gave the most accurate results for 1 time step ahead. Hence our further study is focused on

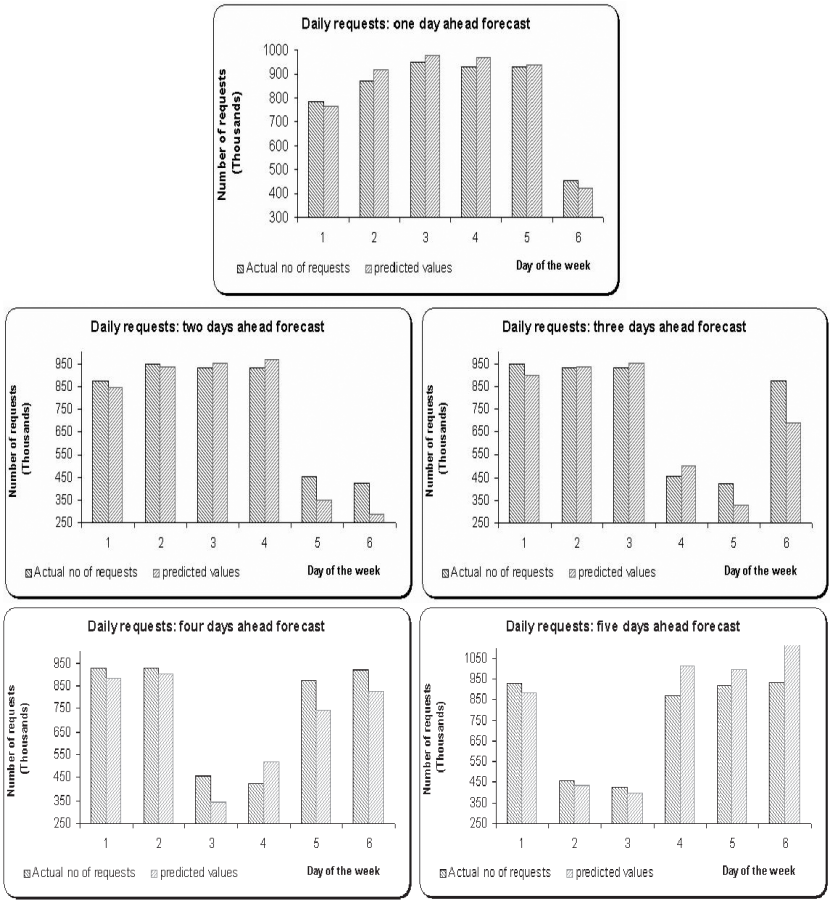

Fig. 15.10. Test results of daily prediction of Web traffic volume

Table 15.4. Training and test performance for hourly page volume prediction

	Root Mean Squared Error (RMSE)			
	FIS (With Cluster Input)		FIS (Without Cluster Input)	
Predicting Period	Training	Test	Training	Test
1 hour	0.04334	0.04433	0.09678	0.10011
12 hours	0.06615	0.07662	0.11051	0.12212
24 hours	0.05743	0.06761	0.10891	0.11352

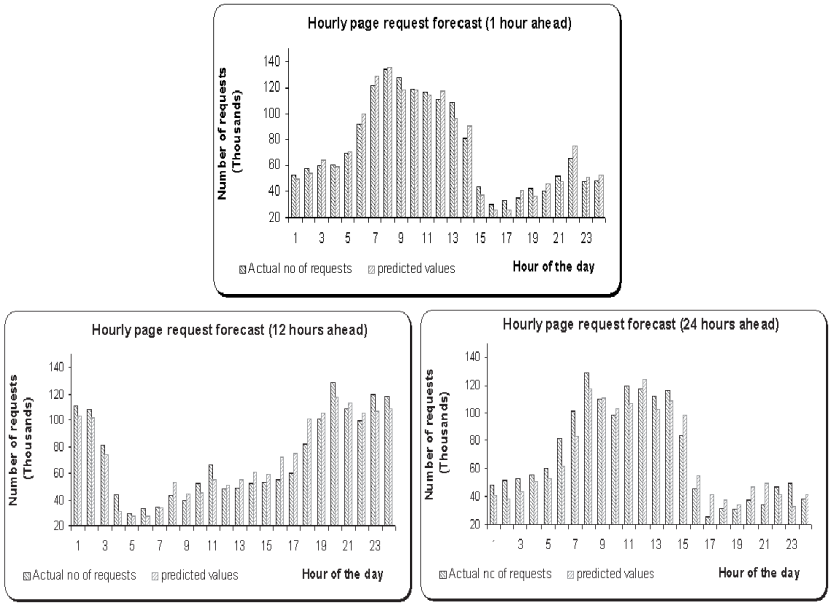

Fig. 15.11. Test results of hourly prediction of volume of page requested

developing ANN and LGP models to predict 1 time step ahead (hourly and daily access patterns).

15.5.2 Artificial Neural Networks (ANNs)

We used a feedforward NNs with 14 and 17 hidden neurons (single hidden layer) respectively for predicting the daily and hourly requests. The learning rate and momentum were set at 0.05 and 0.2 respectively and the network was trained for 30,000 epochs. The network parameters were decided after a trial and error approach. Meantime, the inputs without cluster information were also explored in the experiments. The obtained training and test results are depicted in the Table 15.5.

15.5.3 Linear Genetic Programming (LGP)

We used the "Discipulus" simulating workbench to develop the model using LGP. The settings of the various parameters are of the utmost importance for successful performance of the developed model. We used a population size of 500, 200000 tournaments, a crossover and mutation rate of 0.9 and a maximum program size of 256. As with the other paradigms, both with and without cluster information for the inputs were experimented on and the training and test errors are depicted in Table 15.5.

Table 15.5. Training and test performance of the 3 SCPs

		Predicting Period					
		Daily (1 day ahead)			Hourly (1 hour ahead)		
		RMSE			RMSE		
SCPs	Cluster Input	Train	Test	CC	Train	Test	CC
FIS	Without	0.0654	0.0956	0.9876	0.0654	0.0956	0.9768
FIS	With	0.0176	0.0402	0.9953	0.0433	0.0443	0.9841
ANN	Without	0.0541	0.0666	0.9678	0.0985	0.1030	0.8764
ANN	With	0.0345	0.0481	0.9292	0.0546	0.0639	0.9493
LGP	Without	0.0657	0.0778	0.0943	0.0698	0.0890	0.9561
LGP	With	0.0543	0.0749	0.9315	0.0654	0.0516	0.9446

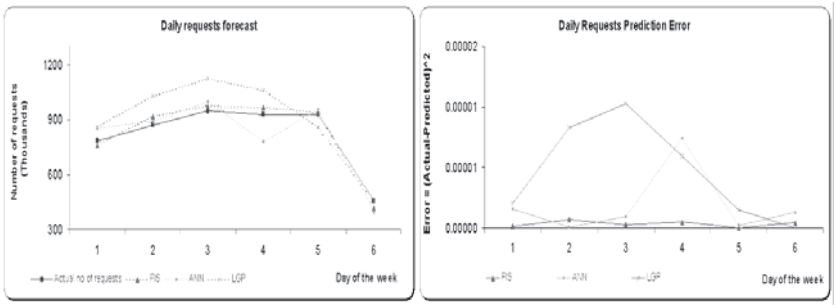

Fig. 15.12. One day ahead request prediction and error using SCPs

15.5.4 Experimentation Results Comparison with Three SCPs

Figure 15.12 illustrates the comparison of the performance of the different SCPs for 1 day ahead daily prediction of Web traffic volume and the error obtained using the test dataset. Figure 15.13 depicts the comparative performance of the SCPs' 1 hour ahead prediction of Web traffic volume and the error obtained using the test dataset.

15.6 Conclusions and Discussions

The discovery of useful knowledge, user information and access patterns allows Web based organizations to predict user access patterns and helps in future developments, maintenance planning and also to target more rigorous advertising campaigns aimed at groups of users [7]. Our analysis on Monash University's Web access patterns reveals the necessity to incorporate computational intelligence techniques for mining useful information. WUDA of the SOM data clusters provided useful information related to user access patterns. As illustrated in Tables 15.3, 15.4 and 15.5 all the three considered SCPs could easily approximate the daily and hourly Web access

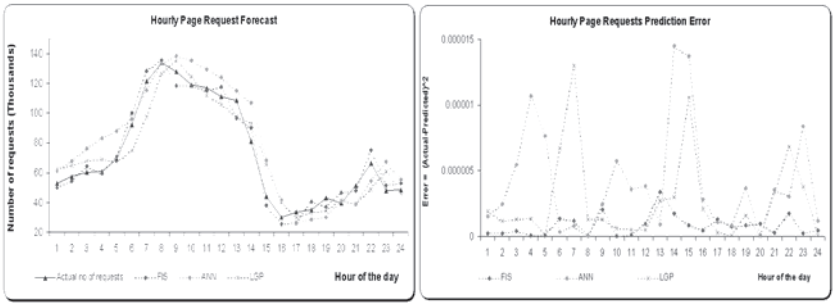

Fig. 15.13. One hour ahead request prediction and error using SCPs

trend patterns. Among the three SCPs, the developed FIS predicted the daily Web traffic and hourly page requests with the lowest RMSE on test set and with the best correlation coefficient. As discussed in [8], neuro-fuzzy hybrid systems for time series analysis and prediction can take advantages of fuzzy systems and neural networks to give excellent prediction accuracy. When compared to LGP, the developed NNs performed better (in terms of RMSE) for daily prediction but for hourly prediction LGP gave better results. Our experiment results also reveal the importance of the cluster information to improve the predict accuracy of the FIS. These techniques might be useful not only to Web administrators but also to Website tracker software vendors.

We relied on the numerical/text data provided by the Web log analyzer that generates statistical data by analyzing Web logs. Due to incomplete details, we had to analyze the usage patterns for different aspects separately, preventing us from linking some common information between the different aspects, trends, patterns etc. For example, the domain requests and the daily or hourly requests are all standalone information and are not interconnected. Therefore, a direct analysis from comprehensive Web logs that covers different interlinked features might be more helpful.

In this research, we considered only the Web traffic data during the university's peak working time. Our future research will also incorporate off-peak months (summer semesters) and other special situations such as unexpected events and server log technical failures. We also plan to incorporate more data mining techniques to improve the functional aspects of the concurrent neuro-fuzzy approach.

References

1. Abraham A, Nath B (2000) Evolutionary Design of Fuzzy Control Systems – An Hybrid Approach. In: Wang JL (ed) Proceedings of the Sixth International Conference on Control, Automation, Robotics and Vision, (CD ROM Proceeding), Singapore
2. Aggarwal C, Wolf JL, Yu PS (1999) Caching on the World Wide Web. IEEE Transaction on Knowledge and Data Engineering 11(1): 94–107
3. Analog (2002) Website Log Analyser. at URL: http://www.analog.cx

4. Banzhaf W, Nordin P, Keller RE, Francone FD (1998) Genetic Programming: An Introduction On The Automatic Evolution of Computer Programs and Its Applications. Morgan Kaufmann Publishers, Inc.
5. Brameier M, Banzhaf W (2001) A comparison of linear genetic programming and neural networks in medical data mining, Evolutionary Computation. IEEE Transactions on Evolutionary Computation, 5(1): 17–26
6. Chang G, Healey MJ, McHugh JAM, Wang JTL (2001) Web Mining. In: Mining the World Wide Web – An Information Search Approach. Kluwer Academic Publishers, pp. 93–104
7. Cooley R, Mobasher B, Srivastava J (1997) Web Mining: Information and Pattern Discovery on the World Wide Web. In: Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97). Newport Beach, CA, pp. 558–567
8. Dote Y (2003) Intelligent Hybrid Systems for Nonlinear Time Series Analysis and Prediction Using Soft Computing. Invited Talk in the 3rd International Conference on Intelligent Systems, Design and Applications. Tulsa, USA.
9. Kohonen T (1990) The Self-Organizing Maps. In: Proceedings of the IEEE. Vol. 78, pp. 1464–1480
10. Honkela T, Kaski S, Lagus K, Kohonen T (1997) WEBSOM – Self Organizing Maps of Documents Collections. In: Proceedings of Workshop on Self-Organizing Maps (WSOM'97). Espoo, Finland, pp. 310–315
11. Jang R (1992) Neuro-Fuzzy Modeling: Architectures, Analyses and Applications, PhD Thesis, University of California, Berkeley, USA
12. Kohonen T, Kaski S, Lagus K, Salojrvi J, Honkela J, Paatero V, Saarela A (2000) Self Organization of a Massive Documents Collection. IEEE Transaction on Neural Networks 11(3): 574–585
13. Monash (2002) Server Usage Statistics, Monash University. Australia, at URL:http://www.monash.edu.au
14. Pal SK, Talwar V, Mitra P (2002) Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions. IEEE Transaction on Neural Networks 13(5): 1163–1177
15. Pirolli P, Pitkow J, Rao R (1996) Silk from a Sow's Ear: Extracting Usable Structures from the Web. In: Proceedings of Conference on Human Factors in Computing Systems. Vancouver, British Columbia, Canada, pp. 118–125
16. Srivastava J, Cooley R, Deshpande M, Tan PN (2000) Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. SIGKDD Explorations 1(2): 12–23
17. Sugeno M (1985) Industrial Applications of Fuzzy Control. Elsevier Science Pub Co.
18. Wang X, Abraham A, Smith KA (2002) Web Traffic Mining Using a Concurrent Neuro-Fuzzy Approach. In: Computing Systems: Design, Management and Applications. Santiago, Chile, pp. 853–862
19. Zadeh LA (1998) Roles of Soft Computing and Fuzzy Logic in the Conception, Design and Deployment of Information/Intelligent Systems. In: Kaynak O, Zadeh LA, Turksen B, Rudas IJ (eds) Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications. pp. 1–9

20. Zhang YQ, Lin TY (2002) Computational Web Intelligence: Synergy of Computational Intelligence and Web Technology. In: Proceedings of 2002 World Congress on Computational Intelligence, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'02). Honolulu, Hawaii, USA
21. Zurada JM (1992) Introduction to Artificial Neural Systems. West Publishing Company