An empirical evaluation of classifier combination schemes for predicting user navigational behavior

Enric Mor and Julià Minguillón

Estudis d'Informàtica i Multimèdia,
Universitat Oberta de Catalunya,
08035 Barcelona, Spain
{emor,jminguillona}@uoc.edu

Abstract. In this paper we present a simple classification system for predicting user behavior when browsing a web site devoted to inform about university degrees. More than building a very accurate classifier, we want to study which kind of combination scheme performs better in front of a complexity constrain. A set of marks embedded in the web pages being visited by each user is used as the input for a classification system which decides whether the user will be interested in accessing other related parts of the web site or not. We compare two different classification systems: the first is built using decision trees for the whole data set, with the aim of studying user profiles and variable importance, while the second, which is an adaptive version, it combines simple classifiers based on small decision trees using a combination of the voting or cascading paradigms, in order to make predictions which evolve during the period of time that the web site is collecting data. Results show that it is possible to extract useful information for studying user profiles and for predicting user behavior using small decision trees.

1 Introduction

Nowadays, Internet has become a common tool that people use for finding information about almost any subject. Companies, institutions and even particular users provide contents which are browsed by millions of people every day. The large amount of contents and services available through Internet makes users to be very sensitive to the way the information is presented and how it can be accessed and browsed. Modern web design takes into account all of these aspects in order to adapt the web site to the way each user uses it.

The web site described in this paper is part of a large educational site devoted to provide information about several university degrees for potential students. Basically, users browse the web site visiting several web pages related to the available options (the different degrees grouped by subject) but sometimes they also visit other web site pages related to general information, thus showing a further interest in a possible future use of the services offered by the university through its web site.

Most users browsing the web site visit only a small number of web pages, between four and six. Therefore, we are interested in making fast and reasonably accurate predictions with simple classifiers, in order to adapt the web page layout, and to reinforce (visually, for instance) all those elements that the user seems to be more interested in. Decision trees [1] allow us to build classifiers that partially fulfill the requirements stated above, simplicity and accuracy. We use orthogonal splits which allow us to obtain a valuable interpretation of the built classifiers for a posterior analysis.

This paper is organized as follows: Section 2 describes the structure of the web mining classification problem and the available data sets and preprocessing. Section 3 describes the classifiers used and the adaptive scheme. Section 4 describes the experiments and Section 5 summarizes the conclusions, and future work in this subject is also outlined.

2 User navigational behavior

Our main goal is to study the different web site user navigational profiles, trying to extract two kinds of information about the web site users: the parts of the web site that they access to and the way they do it. This information could be used in two ways: first, for automating usability evaluation [2] of the web site with the aim of improving its information architecture and interaction design. And second, for providing adaptive web browsing [3, 4].

2.1 Data set

The data set used in this paper has been obtained from a web site[1] devoted to inform about the several options that potential future university students have available. This web site follows a classical design based in frames, a large frame occupying the right size of the screen with the list of available options, and a small left frame with additional information related to the educational model.

This web site has been collecting data for six months, creating a large data set with more than 170000 entries, one for each recorded visit. This data set is not a classical web log file, but a list of special marks which are embedded in the web pages which are being monitored. These marks are then used as a reduced web log file. A similar approach is described in [5]. Using marks instead of classical log files has several advantages for our purposes: first, preprocessing is simplified as only relevant information is present in the marks file, reducing both time and space web site processing requirements. And second, it is easier to track user navigation in order to study a particular user navigational behavior, for example.

Each mark contains information about a user number which identifies each user uniquely, the web page visited by the user, and both the date and time of the visit. The user id field is a combination of the IP address of the first visit, the

[1] `http://www.uoc.es/web/cat/launiversitat/estudis`

date and time of the first visit and a magic number which is randomly assigned to each user the first time he or she visits the web site. Then, a cookie based mechanism is used to track the following user visits to the web site.

Preprocessing is needed in order to remove all those list entries where the cookie based tracking system fails, mainly because of a small percentage of users do not accept cookies. A total of 17 links (14 make reference to the available options and 3 make reference to the additional information) are used as the basic information extracted from each user: which links are visited and how many times. In this stage we also remove duplicated entries, if present. The total of entries is over 160000, showing that more than a 94% of the original data set is used, so only a small percentage of users is not selected for this study.

Then, for each different user in the data set (more than 53000), a binary vector containing 14 variables (whether the user has visited a link or not) is built. The label assigned to each user is also a binary value showing whether such user has visited at least one of the links of the related information area (denoted by 1) or not (denoted by 0). This creates a final data set with 53740 entries, each entry is a 14-dimensional binary vector with a binary label as the outcome for such vector.

The estimated probability of a user accessing to the additional information (that is, $\mathbf{P}\{Y = 1\}$) is around 20%. Nevertheless, for our purposes we suppose that we are building a classifier better than random guessing without any previous knowledge, so we will force *a priori* probabilities of both classes to be 1/2. This is similar to force different misclassification costs for each class, as described in [1].

2.2 Feature selection

As described above, each input sample is a binary vector containing whether a user visits a collection of links or not. Nevertheless, as we are using orthogonal hyperplanes, it is better to compute additional classification features which can be used to find boundaries more complex than simple orthogonal splits.

Furthermore, users usually visit all web pages that are related to the same subject. For example, people interested in psychology studies usually also visit the psychopedagogy web page. Therefore, for every subject involving more than one option, we compute a new classification variable by combining all the classification features related to such subject using an OR function. A total of five additional classification features is computed. As stated in [6], disjunctions require a large decision tree to be described, so adding new classification features may be useful to fight against the replication problem.

The use of binary vectors is very interesting because only one value need to be tested for each classification feature in order to find the best split, that is, internal decision functions are of the form $x_i = 1$. Furthermore, at each stage of the training process, the data set represented by the leaf which is going to be split can be sorted using a simple algorithm with complexity $\mathcal{O}(N)$, so the training process is speeded up dramatically.

3 Combining classifiers

Our goal is not only to build a simple classification system for studying user profiles for the whole collected data set, but also to design a simple and fast adaptive classification system for improving classification performance during the period of time the web site is running and accepting petitions. The main idea is to see how to combine different classifiers which are created using different training sets available for predicting user behavior, and to study the evolution of such predictions. In this paper, we study two classical paradigms of combining classifiers, voting [7] and cascading [8] in order to build an adaptive classification system. We call the first classification system "static", and the second one "adaptive".

Notice that we are not using any online learning algorithm [9], but combining simple classifiers which are built with all available data at the moment of building the final classifier.

3.1 Voting and cascading

Suppose we split the data set according to a temporal criterion (weekly or monthly, for example), and each subset is denoted by d_i, with a total of M data subsets. The union of all data sets starting with d_1 up to d_i inclusive is denoted by D_i. Our goal is to build a classification system T_i any time a new data subset is available (that is, when the collecting process is stopped). Therefore, once the data subset d_i is available, we have several options:

1. Type A (recent history): use the subset d_i to build a decision tree A_i and make $T_i(x) = A_i(x)$.
2. Type B (complete history): use the subset D_i to build a decision tree B_i and make $T_i(x) = B_i(x)$.
3. Type C (voting): use the subset d_i to build a decision tree A_i, D_i to build B_i and make $T_i(x) = V(A_i(x), B_i(x))$ where V denotes a voting scheme.
4. Type D (cascading d_i): use the subset d_{i-1} to build a decision tree A_{i-1}, and use it with cascading with the subset d'_i to build a decision tree A'_i and make $T_i(x) = A'_i(x)$.
5. Type E (cascading D_i): use the subset D_{i-1} to build a decision tree B_{i-1}, and use it with cascading with the subset D'_i to build a decision tree B'_i and make $T_i(x) = B'_i(x)$.

Voting can be a simple majority rule or a weighted scheme using class probabilities. For the voting option, a special "mixed" class may be used to denote that several classifiers do not generate the same outcome, so a partial classification system is built. This value may also be used to label those predictions made by a single decision tree with a small margin. This margin is defined for each leaf as the probability of making a right prediction minus the probability of making a mistake. Therefore, the new labelling rule for a leaf i is

$$l'_i(t) = \begin{cases} l_i(t) & \text{if } P\{l_i(x) = y\} - P\{l_i(x) \neq y\} > \epsilon \\ \text{"mixed"} & \text{otherwise.} \end{cases}$$

where $l_i(x)$ is the computed label using majority voting. This allows us to discard those samples that fall in leaves which contain elements from several classes. A similar approach has been successfully used in [10]. Suppose the number of classes is two, and that p is the probability of the most populated the class in a leaf, so $p > 1/2$. Then, the margin is $p - (1 - p) = 2p - 1$, and a leaf will label elements as "mixed" if $p < 1/2 + \epsilon/2$. The margin is used as additional information for the cascading ensemble, as it is a measure of confidence in the predictions made by the decision tree.

There are more options for constructing a classifier at the i-th stage, as both voting and cascading could be combined, voting could use both the t_i and/or the T_i, and cascading could also be done using the different t_i and not only T_{i-1}. Nevertheless, the options described above are enough to carry out an empirical evaluation of classifier combination for our purposes.

4 Experimental results

In this section we describe the two kinds of classification systems used for evaluating the possibility of building a system that predicts user navigational behavior. In both cases we use the same setup for decision trees: there is a maximum depth constrain of six levels, orthogonal splits are computed using the entropy impurity criterion, and pruning is done using the number of leaves and misclassification error as tree functionals.

4.1 Static classifier

In the first experiment, we build a small decision tree for studying the bias-variance decomposition [11] of the misclassification error. Our goal in this case is to extract any useful information which may be used for the web page usability study, and also to improve user navigation, to identify user profiles and also to design the adaptive classification system. For this experiment we use the following setup: the original data set (53740 samples) is split using N-fold cross validation with $N = 3$, and a total of 25 bootstrap replicates are generated to compute the bias-variance decomposition for each training set. This process is repeated five times and results are averaged. Table 1 shows the confusion matrix for this experiment.

Notice that, assuming an equal *a priori* probability for each class, classification accuracy is only slightly better than random guessing. On the other hand, the bias-variance decomposition describes the misclassification error as $E = B + V$, and for this classifier we obtain $0.448 = 0.441 + 0.007$, showing that a large bias is the main reason of the obtained misclassification error. Both facts seem to indicate that we are facing a difficult classification problem, which cannot be easily solved by using small decision trees with orthogonal splits. The β value for class 1 is low, as only one out of four times a sample is labelled as 1 is really a true user interested in visiting the additional information web part. The average rate of the decision trees is $R = 4.22$, while the maximum depth

class i / j	0	1	**total**	α
0	957819	836431	1794250	53.4 %
1	167070	277930	445000	62.5 %
total	1124889	1114361	2239250	—
β	85.1 %	24.9 %	—	**55.2 %**

Table 1. Confusion matrix for the static classification system.

is $\overline{R} = 5.92$, showing that it is impossible to make any accurate prediction with less than four questions.

Regarding variable importance, it is measured by computing the impurity gain at each split. With the setup defined above, a total of 125 decision trees are built, and 95 of them use the same variables for the first stages, showing that despite the lack of precision in the prediction, users visiting some parts of the web site are more likely to visit the additional information links than the rest. As a possible result of this experiment, the web layout could be redesigned in order to incorporate additional information elements (the links in the left frame) but with a different visual approach (such as buttons or pop-up windows, for example). This new elements would be only present in those web pages related to studies more visited by people accessing the web site which form the target of the classification system. This should be done carefully, though, as classification accuracy is not very high, in order to avoid annoying information which can cause the users to stop visiting the web site.

4.2 Adaptive classifier

As we want to study the possibility of creating a classification system for a real scenario, we do not use N-fold cross validation or any other technique for constructing the training and the test sets for the adaptive case. At each stage we will suppose we only have past data available for training, while future data is used for testing purposes. For a monthly regular basis ($M = 5$), the available training and test sets are:

i	**training** d_i / D_i	**testing**
1	14087 / 14087	42393
2	12890 / 25336	31480
3	8008 / 31700	24775
4	6134 / 36544	19574
5	8561 / 43680	12063

Table 2. Available training and test sets.

Notice that the first decision tree for each kind of classification system will be always the same, as only d_1 is available for training purposes, and $D_1 = d_1$ by definition.

Table 3 shows the results for the simplest classification system, which uses only the data collected during a period of time. R is the average length (that is, the number of questions asked in order to classify a sample), $\overline{R}$ is the maximum depth, and α and β are the sensitivity (the ability to identify those who visit the related areas of the web site) and the percentage of samples labelled as true which are really true visits, respectively.

i	R	$\overline{R}$	**accuracy**	α	β
1	4.25	6	57.5%	54.0%	23.8%
2	1	1	74.2%	15.4%	24.9%
3	5.07	6	54.4%	59.9%	21.9%
4	1	1	75.8%	17.6%	24.1%
5	1.45	3	82.0%	3.1%	29.8%

Table 3. Results for the type A classification system.

Notice that both α and β values show a bizarre behavior, as several decision trees are trivial (just one split), yielding to biased classifiers towards the most populated class. Using only d_i is therefore a simple but inefficient way to build an adaptive classification system. The monthly basis might be also a too restrictive way to build the training sets.

Table 4 shows the results for the type B classification system, which uses all available history to build a small decision tree.

i	R	$\overline{R}$	**accuracy**	α	β
1	4.25	6	57.5%	54.0%	23.8%
2	3.63	5	57.6%	50.1%	23.3%
3	3.70	6	59.8%	50.8%	22.5%
4	3.70	6	56.2%	58.1%	21.9%
5	3.84	6	60.2%	52.0%	22.3%

Table 4. Results for the type B classification system.

In this case classifiers become more accurate as more data is available for training, as expected. Accuracy is slightly higher than in the static case, although the β values are slightly lower. This might be corrected using uneven

misclassification costs, forcing the classification system to put more effort in one class than in the other one.

Table 5 shows the results for the type C classification system, the voting scheme. As we use the mixed class to avoid misclassifying those samples where both classifiers disagree, p shows the percentage of input samples classified.

i	p	**accuracy**	α	β
1	100 %	57.5%	54.0%	23.8%
2	69.8 %	72.8 %	65.2%	24.9%
3	74.7 %	59.5%	67.1%	21.7%
4	66.1 %	74.2%	59.4%	24.1%
5	60.9 %	84.6%	50.2%	29.3%

Table 5. Results for the type C classification system.

Notice that classification accuracy is much better than in the previous cases, as only a percentage of the input samples is classified. Nevertheless the β values are not very accurate, showing the same accuracy problems.

Table 6 shows the results for the type D classification system, the cascading ensemble using the subsets d_i. In this case we observe the same problems than with the type A classification system.

i	R	$\overline{R}$	**accuracy**	α	β
1	4.25	6	57.5%	54.0%	23.8%
2	1.96	2	72.4%	19.3%	24.6%
3	4.07	5	59.6%	54.4%	23.2%
4	1.51	4	77.0%	16.0%	25.3%
5	2.17	6	81.8%	10.2%	40.4%

Table 6. Results for the type D classification system.

Finally, Table 7 shows the results for the type E classification system, which uses the subsets D_i. In this case results (accuracy, concretely) are surprisingly worse than in the previous experiment, but both the α and β values are preferable as they show an increasing behavior.

i	R	$\overline{R}$	**accuracy**	α	β
1	4.25	6	57.5%	54.0%	23.8%
2	3.44	6	57.0%	51.5%	23.3%
3	3.55	6	47.7%	74.9%	21.9%
4	3.57	5	56.3%	65.2%	23.3%
5	2.16	6	56.1%	65.2%	23.0%

Table 7. Results for the type E classification system.

5 Conclusions and future work

In this paper we have presented an adaptive classification system using small decision trees under a combined cascading/voting paradigm for predicting user behavior when browsing a web site. Several conclusions may be drawn:

- The static classification system built using all the data available at the end of the collecting data period is only slightly better than random guessing, with a very large bias in comparison to variance. This is, in fact, partially caused by the intrinsic difficulty of the problem studied in this paper, but also by the limitations imposed on the classifiers used to build the classification system: limited maximum depth decision trees using only orthogonal splits. Nevertheless, this classifier provides useful information relative to user profiles accessing the web site and variable importance.
- On the other hand, the adaptive classification systems are better than the static one, specially the type C which is based on a voting scheme using a "mixed" class for doing partial classification.

In fact, the results obtained in this paper show that users do not follow a simple predictable behavior when browsing the educational part of the web site. Therefore, it is necessary to redefine the mark embedding system and the user task analysis problem in order that more detailed studies could be carried out. Further work is in progress to improve the classification results, but also a more comprehensive definition of the users navigational behavior.

Acknowledgements

This work is partially supported by the Spanish MCYT and the FEDER funds under grant no. TIC2001-0633-C03-03 STREAMOBILE. We thank Genís Berbel for providing the data set used in this paper.

References

[1] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth International Group (1984)

[2] Chi, E.H., Pirolli, P., Pitkow, J.E.: The scent of a site: a system for analyzing and predicting information scent, usage, and usability of a web site. In: Proceedings of the CHI Conference on Human Factors in Computing Systems. (2000) 161–168
[3] Perkowitz, M., Etzioni, O.: Adaptive web sites: Automatically synthesizing web pages. In: Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98). (1998) 727–732
[4] Spiliopoulou, M., Pohle, C., Faulstich, L.: Improving the effectiveness of a web site with web usage mining. In: Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (1999) 142–162
[5] Breese, J., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, USA (1998)
[6] Pagallo, G., Haussler, D.: Boolean feature discovery in empirical learning. Machine Learning **5** (1990) 71–99
[7] Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 226–239
[8] Gama, J., Brazdil, P.: Cascade generalization. Machine Learning **41** (2000) 315–343
[9] Schlimmer, J.C., Granger, R.H.: Incremental learning from noisy data. Machine Learning **1** (1986) 317–354
[10] Minguillón, J., Pujol, J., Zeger, K.: Progressive classification scheme for document layout recognition. In: SPIE Proceedings, Mathematical Modeling, Bayesian Estimation, and Inverse Problems. Volume 3816., Denver, CO, USA (1999) 241–250
[11] Domingos, P.: A unified bias-variance decomposition and its applications. In: Proceedings of the Seventeenth International Conference on Machine Learning, Stanford, CA, USA, Morgan Kaufmann (2000) 231–238