Predicting Web User Behaviour – a Decision-theoretic Approach

Krzysztof Dembczyński[1], Wojciech Kotłowski[1], Marcin Sydow[2]

[1] Institute of Computing Science, Poznań University of Technology,
60-965 Poznań, Poland
[2] Polish-Japanese Institute of Information Technology, 02-008 Warsaw, Poland

Abstract. In this paper we present our solution to the ECML/PKDD 2007 Challenge Task that concerns prediction of Internet user behaviour by characterising the nature of their Web page visits. Our solution has low time and space complexity, scales well with large datasets and, at the same time, produces high-quality results. Comparison of performance of our ultimate approach with a suit of other approaches that we examined exhibits its superiority as well as hardness of the given datasets.

1 Introduction

The contest objective [4] was to predict Internet user behaviour by characterising the nature of their visits. The visit is defined by categories of visited web pages and the number of page views in each category. The contest was organised into 3 tasks: predicting the number of Web page categories (1 or greater than 1), predicting the first 3 visited categories and predicting the number of pages seen in each of the first 3 categories.

The data was provided by Gemius – a leading Internet market research company in Poland – and divided into: training set (379485 records) and testing set (166299 records – twice less than in the training one). Each record contains `record number`, `user id`, `timestamp`. Additionally, in the training set the records contain the sequence of pairs, each of the form `category`, `#pages`. Each record corresponds to a user session started at timestamp and reflects the category and number of pages seen by that user in chronological order (from left to right). Example: `248 46 1167680792 12,1 8,7 12,3`.

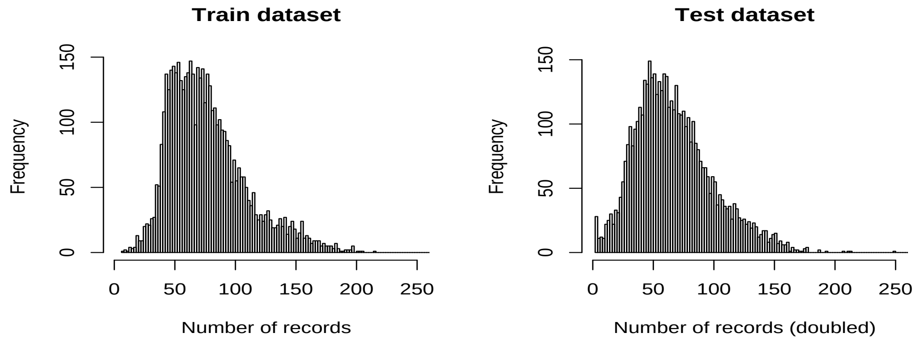
In addition, a third, auxiliary dataset was provided which contains 4882, user-related records contains the following attributes: `user id`, `country`, `region`, `city`, `system`, `sysVer`, `browser`, `browserVer`.

Our exploratory analysis phase is described in section 2. Section 3 describes our first attempts of building *global models*. During our work, we have observed that models constructed separately for each user (*user-models*) give better results than the global ones. The reason of this is a specific nature of data and the formulation of the challenge problems. In order to obtain stable predictions, our methods are strongly based on statistical decision [8, 1] and learning theory [3, 6], section 4 is devoted to this topic. Final approaches to the challenge problems are presented in sections 4.1-4.3. The last section makes a short conclusion.

2 Exploratory Data Analysis

Any serious data mining task concerning unknown real datasets should be preceded by an appropriate *exploratory data analysis* phase which is regarded as being crucial to obtain high quality results in the subsequent phases [2]. All the computations given in this section were performed using the R package [7].

The datasets concern 4882 different users and 20 different Web page categories. All the users were represented in the training as well as the testing dataset. The number of records per user in each dataset is summarised in Figure 1.



dataset	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	200+ records
train	7.00	54.00	72.00	77.73	94.00	497.00	7
test (doubled)	2.00	46.00	64.00	68.13	86.00	430.00	5

Fig. 1. Summary of the statistics for the number of records per user in the training and testing (doubled) datasets. The typical number of records per user lies between 50 and 90 and does not vary very much but is not high enough (given the number of different categories, and other attributes) to build detailed user-level probability model.

We discovered (fig. 2) a group of users (having the highest id numbers) which were "new" to the recording system at the end of the training dataset. Interestingly, the id number growth rate is higher in this group than in the remaining part but also this rate is constant (see fig. 2, top-left), what means that the group is homogeneous. The group is distinct among the users, since we believe that the user numbers must be assigned in some natural (perhaps, chronological) order. In the testing set all the users recorded are present equally through the whole recording period.

Having inspected the data, we assumed that the timestamp attribute represents the number of seconds measured since the beginning of the era.

Timestamp range is the following: training set: 31/12/2006 - 22/01/2007 (about 1 am), testing set: 22/01/2007 - 31/01/2007 (about 1 pm). The testing set is a chronological continuation of the training set (with less than 1 minute

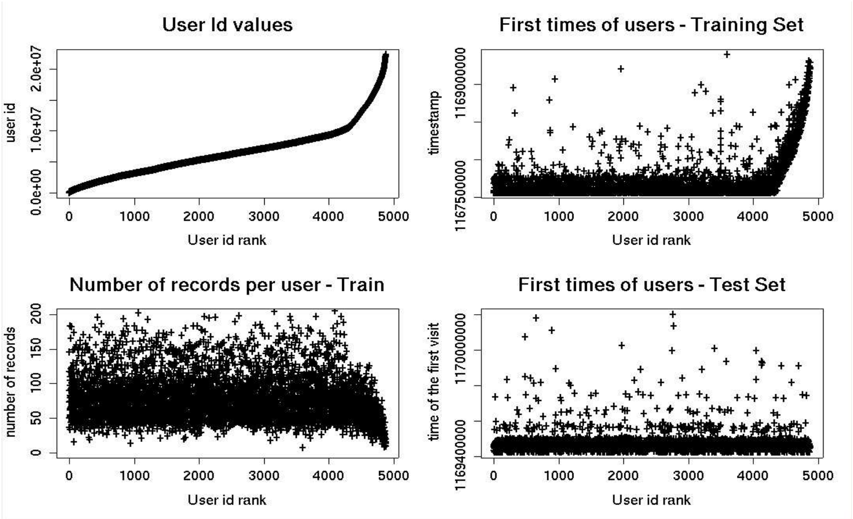


Fig. 2. The users with the highest id numbers are "newcomers" in the training set. The testing set does not contain users with analogous property.

break in between). This chronological relationship was taken into account while selecting our prediction models. It also encouraged to use time-series approach.

We transformed the timestamp attribute to the full date (i.e. `year`, `month`, `month-day`, `week-day`, `hour`, `minute`, `second`) to explore the week and 24-hour periodicity in users' behaviour, among others (fig. 3).

The difference of histograms (fig. 3, the middle column) of week-day-based activity is due only to more Mondays and Tuesdays in the testing set and abnormal activity on the Sunday, 31st December 2006, perhaps due to the New Years Eve greetings traffic, etc. After normalisation, both histograms would be almost flat. Thus, week-day is not a good discriminant at the global level. In contrast, the hour attribute seems to bring valuable discriminative information even on the global level (see fig. 3, left column and the top right histogram).

Much exploratory data analysis effort was devoted to explore the training dataset in order to choose the proper method of solving task 2 and 3.

For each session in the training set the category on the first 3 positions of the visit path were recorded. Subsequently, the above data was aggregated over

Table 1. Statistics concerning the visit length – i.e. the number of consecutive page categories visited in each session. The distribution is extremely right-skewed.

1 category	2 categories	3 cat.	3+ cat.	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
72.9%	11.5%	6.3%	9.25%	1	1	1	2.17	2	200

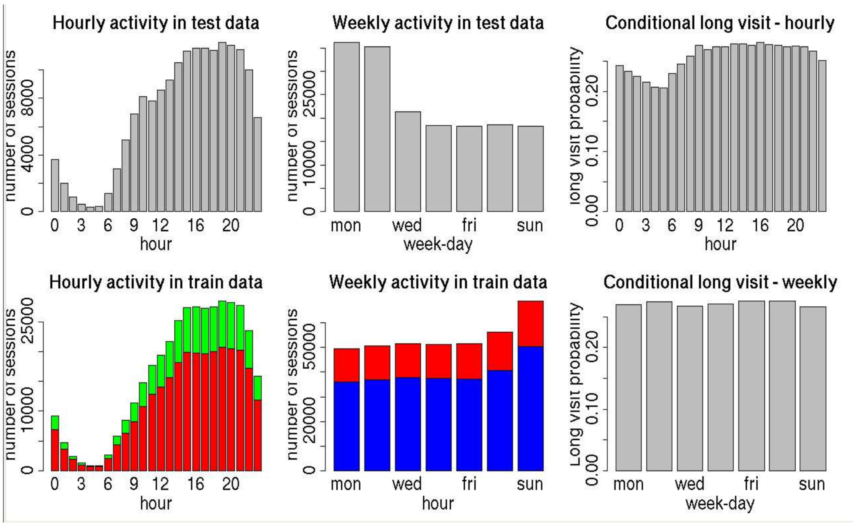


Fig. 3. Periodicity exploratory analysis. For the training set, the shorter bars represent the sessions in which more than 1 category of pages was visited, the longer bars - the other cases (task 1 of the challenge). One can observe (the histograms on the left, and top-right) that hour brings more information that week-day (almost flat bottom-right conditional histogram), on the global level.

separate users and, for each user, three 20-dimensional distribution vectors over categories were computed - for the 1st, 2nd and 3rd category on the visit path (e.g. if a particular user visited only category 12 and 8 on the first position of their session with equal frequency, the corresponding 1st-category distribution vector has entries of 0.5 on positions 8 and 12 and values of 0 elsewhere).

Subsequently, those probability distribution vectors served as the basis for computing entropy (left column on fig. 4), number of non-zero entries (the middle column) and the probability of the most likely category on the position 1, 2 or 3, for a given user (the right column on the figure).

All those measurements served to convince us that simple, user-level model for tasks 2 and 3 is a reasonable solution. Namely, the graphs on Figure 4 clearly show, that for most of users the categories on their 1st, 2nd and 3rd positions are quite easily predictable. In particular, low entropy, low number of categories encountered on the positions under examination and generally very high probability of the most likely category on a given position strongly influenced our decision of choosing very fast, yet simple prediction method for these tasks. One can easily observe from the fig. 4 that especially categories on the 1st and 3rd positions seem to be easily predictable. The phenomenon of the 2nd category being much harder to predict can be explained as follows. In 74.5% of sessions of over 2 categories the third category is the same as the 1st category.

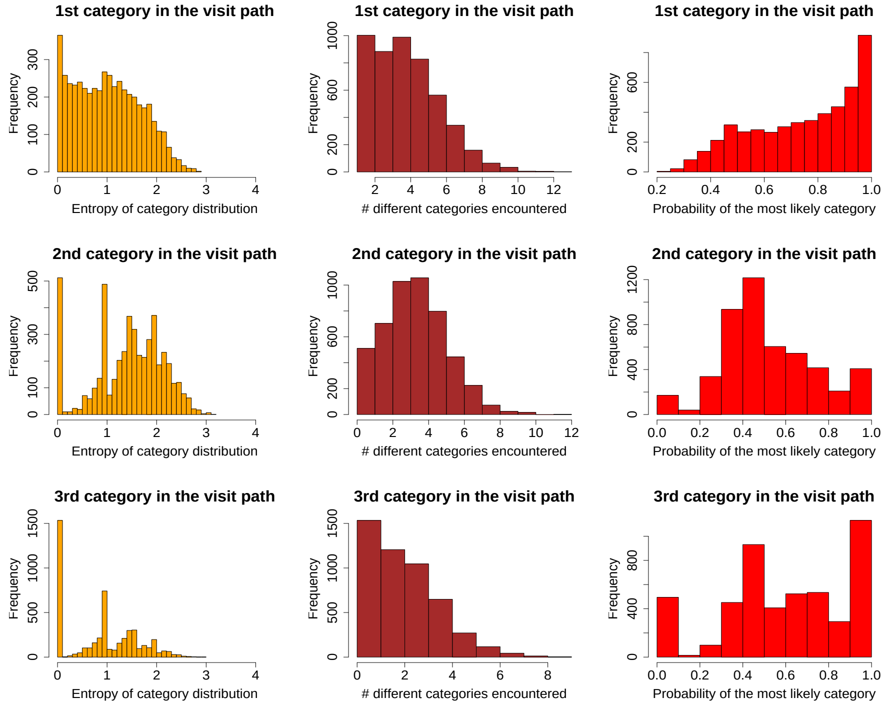


Fig. 4. Analysis focused on assessing feasibility of user-level simple modelling for the Task 2 and 3. Row number corresponds to the position on visit path. Left column: entropy of 20-dimensional probability distribution over categories – notice its relatively low values (maximum entropy for 20-dimensional distribution is 4.32193). Middle column: number of different categories on a given position (notice that it is close to 1). Right column: data-estimated probability of the most likely category on a given position (most of the mass is definitely above the value of 0.5)

3 First Attempts on Global Models

The property which is apparent in the collected dataset is its granularity. The finer granule concerns a single visit (*visit's granule*), while the coarser granule corresponds to a single user (*user's granule*). On the one hand, using visit's granules permits the classifier to evaluate each visit separately, possibly giving different responses in each case, e.g. depending on the position of visit in chronological order, timestamp, etc. On the other hand, for user's granules we are able to do the averaging over all the visits for a given user, thus reducing the variance and giving more reliable responses.

Simple Global Model. Our first attempt was related to visit's granules. We considered the global model, in which we divided the original training set into 2 parts: training 89% and testing 11%. The division reflects the chronological relationship between the original training and testing sets. The first 89% of the recorded sessions for each user constituted the training subset, and the last 11% the testing subset. To train the classifiers, we used features such as user data (`country`, `region`, `city`, `system`, etc.), week day, hour, part of the day, time from the last visit, number of visits during the day, number of visits in last 60, 120, etc. minutes, type of a last visit (whether it was a short or long visit), type of a second last visit, etc. However, obtained results were not satisfactory. The best result for task 1, which we have obtained using j48 (C4.5 implementation in Weka [9]), was 75.7% correctly classified visits.

Enhanced Global Model. Due to the poor quality of the results of the simple global model, we decided to estimate some additional values describing the behaviour of the users. In order to achieve it, we prepared an estimation set isolated from the training set. To reflect the chronological relationship present in data, the first 70% of the recorded sessions for each user constituted the estimation subset, the next 19% – the training subset, and the last 11% – the testing subset. The features calculated on the estimation set were:

- category-based (210 attributes in total): average number of pages seen in a session, average number of groups seen each session, number of different categories seen each session, majority category at position 1 through 3 on the path, average number of pages on each position (1st through 3rd), average number of pages seen in each category, average number of groups of each category, distribution of categories encountered on the 1st-3rd position of the path for each category, average number of pages seen in the 1st-3rd position for each category.
- visit length-based, obtained by considering only two types of visits (short or long) and estimating the probability of long visit for each day of week, for each hour of working day and hour of weekend day. Probability estimates were smoothed using the kernel estimation method (Gaussian kernel).

Notice that all those features represent some average characteristics of each user so that they are related to user's granularity level. In addition, user data (`country`, `region`, etc. - 7 attributes in total) were also included in the dataset.

We experimented with taking subsets of the above attributes. We also tried to take logarithms of some of the above attributes (those which were extremely right-skewed). In this setting, for the task 1, the best results have been obtained with j48 algorithm. The result was 76.7% correctly classified visits.

User Models. In the previous approach, the information about users was incorporated to the model by isolating the estimation set (including most of the observations) and calculating some coefficients for each user by averaging over their visits. In this approach, we decided to use *directly* user id number (attribute `user id`), without extracting any additional information about the users. This approach has been verified to be the most successful, therefore will be described in the next three sections (sections 4.1-4.3), separately for each task. Here we present common features of all the models.

Incorporating user id number as a condition attribute leads to the following problem: user id has nominal scale without any order between values, so that each of its values must be treated separately. It is possible to include such attribute in a general model, but for most of the classifiers, it will be binarised, i.e. changed into 4882 (number of users) binary attributes. Taking into account the size of the dataset, this is not a practical solution. Much more practical procedure, which can simulate conditioning on user id attribute, corresponds to building a separate model for each user. Such a procedure has been used in all of the models described later.

In each of the models user id number is used as one of the predictors (condition attributes). However, in none of the models any other attributes of the user (`country`, `region`, etc.) are included. This is due to the fact, that those attributes functionally depend on user id number, or in other words, user id number determines values on those attributes. Thus, they do not introduce any additional information, or in other words, they do not lead to finer granulation.

4 Final Solutions on User Models

All of the solutions described in this section are based on the statistical decision [8, 1] and learning theory [3, 6]. First, we briefly remind the basic concepts.

In the *prediction problem*, the aim is to predict the unknown value of an attribute y (called *decision attribute*, *output* or *dependent variable*) of an object using known joint values of other attributes (called *condition attributes*, *predictors*, or *independent variables*) $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. The task is to find a function $f(\mathbf{x})$ that predicts value y as well as possible. To assess the goodness of prediction, the *loss function* $L(y, f(\mathbf{x}))$ is introduced for penalising the prediction error. Since $\mathbf{x}$ and y are random variables, the overall measure of the classifier $f(\mathbf{x})$ is the *expected loss* or *risk*, which is defined as a functional:

$$R(f) = E[L(y, f(\mathbf{x}))] = \int L(y, f(\mathbf{x}))\mathrm{d}P(y, \mathbf{x}) \tag{1}$$

for some probability measure $P(y, \mathbf{x})$. The optimal (risk-minimising) decision function is:

$$f^* = \arg\min_f R(f). \tag{2}$$

Since $P(y, \mathbf{x})$ is unknown in almost all the cases, one usually minimises the *empirical risk*, which is the value of risk taken from the set of training examples $\{y_i, \mathbf{x}_i\}_1^N$:

$$R_e(f) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(\mathbf{x}_i)). \tag{3}$$

Function f is usually chosen from some restricted family of functions.

When solving the contest tasks, the problem was to find the best approximation of the optimal decision function.

4.1 Solution to Task 1

The task is to predict whether a visit has page views of only one category (*short visit*), or more categories (*long visit*). We deal here with two classes, thus it is a simple binary classification problem for which the most common loss function is so called *0-1 loss*:

$$L_{0-1}(y, f(\mathbf{x})) = \begin{cases} 0 & \text{if } y = f(\mathbf{x}), \\ 1 & \text{if } y \neq f(\mathbf{x}). \end{cases} \tag{4}$$

Coding short visit by -1 and long visit by 1, the optimal decision function for a given $\mathbf{x}$ is:

$$f^*(\mathbf{x}) = \operatorname{sgn}\left(\Pr(y = 1|\mathbf{x}) - 0.5\right). \tag{5}$$

Of course, we have no information about the probabilities $\Pr(y = 1|\mathbf{x})$, so they must be estimated from training examples (alternatively, one can estimate whether the probability is higher or smaller than 0.5).

Shrinkage. Let $\hat{p}(\mathbf{x})$ be an estimator of the probability $\Pr(y = 1|\mathbf{x})$ (denoted as $p(\mathbf{x})$ from this moment on), which is calculated on the dataset. Our objective is to find the decision function defined as:

$$\hat{f}^*(\mathbf{x}) = \operatorname{sgn}\left(\hat{p}(\mathbf{x}) - \theta\right) \tag{6}$$

where, comparing with (5), we used the estimator $\hat{p}(\mathbf{x})$ instead of real unknown probability $p(\mathbf{x})$ and threshold θ instead of 0.5. The motivation for the latter is based on Bayesian inference. Suppose that we impose some prior distribution τ on parameter $p(\mathbf{x})$. It is easily seen from the data that $E_\tau p$, the expected value of p according to the prior distribution τ, is much less then 0.5, since in 73% of the cases the visit is short. It is a well known fact from Bayesian decision theory that the estimated parameters are shrunk towards the center of prior distribution. We impose such shrinkage by introducing regularised estimate of the probability defined as: $\tilde{p}(\mathbf{x}) = \alpha\hat{p}(\mathbf{x}) + (1 - \alpha)E_\tau p$, where α is chosen to be independent of $\mathbf{x}$ for simplicity. But the condition $\tilde{p}(\mathbf{x}) \geq 0.5 \Leftrightarrow \hat{p}(\mathbf{x}) \geq \theta$ where $\theta = \frac{1}{\alpha}0.5 + \frac{\alpha-1}{\alpha}E_\tau p$, and $\theta > 0.5$ as long as $E_\tau p < 0.5$. θ was chosen empirically to maximise the performance on the testing set.

The crucial thing in estimating $p(\mathbf{x})$ is the chosen vector of predictors (condition attributes) $\mathbf{x}$. Depending on the choice, three different models are considered, described below.

Model I: Simple Classification. The only predictor is user id number, so that $\mathbf{x} \equiv j$, where j is the number of user. For each user j, the fraction of the long visits was taken to be a probability estimator $\hat{p}(j)$. This estimator is constant in all observations for a given user. Therefore, it is characterised by a very small variance, but also a significant bias. The time complexity of the algorithm is linear with the number of visits. The memory complexity is linear with the number of users (not including the memory occupied by dataset).

Model II: Trend prediction. Data for each user were regarded as short time series with values 0 (short visit) or 1 (long visit). The abscissa values (predictors) were timestamps of the observations (normalised, in order to avoid some numerical difficulties due to large numbers). For each user, a polynomial trend was fitted to the time series and was used as a probability estimator. The fitting procedure was regularised least squares (ridge regression). The amount of regularisation was chosen empirically, to maximise the performance of the procedure on the testing set. It appears that models with very strong regularisation (more smoothing) are preferred due to their small variance.

For a given user, the time complexity of the method is dominated by the least squares fitting which is done by Cholesky decomposition and has complexity $O(m^3 + \frac{nm^2}{2})$, where n is the number of visits for a given user and m is the degree of fitted polynomial. Since m is fixed, time complexity is linear in the number of visits, so as the memory complexity.

Model III: Autoregression. The autoregressive model was the most sophisticated one that we used. For each user a separate linear model is fitted to the user's time series, based on the following attributes: normalised timestamp, time from the last visit, length of the last visit, average length of the last 2, 4 and 8 visits. Since the predicted value (length of the current visit) depends on the values in previous moments, such algorithm resembles autoregressive models used in time series analysis [5], but with regularised least squares fitting procedure.

The classification procedure is more complicated here – all the objects must be classified chronologically, since the current value depends on the previous values. This causes the model to be less reliable with predicting the latest observations. That is why strongly regularised models (more smoothing, less variance) were preferred.

The complexity of the method, both in time and memory, is the same as for model II (linear in the number of visits), since least squares are also used as fitting procedure. However, training the autoregressive model takes more time due to the greater number of condition attributes.

Results and conclusions. For all our models, we present 3 following estimates:

1. *training score* – value of the score on the training set. This estimate is thus over-optimistic, since the score is measured on the data which were used for fitting the classifier,
2. *validation score* – the training set was divided into 89% proper training set and 11% validation set. The classifier was learned on training set and the score was calculated on validation set. This estimate was used to choose the best classifier for the contest,
3. *solution score* – value of the score on the testing set. We were able to calculate this estimate using the proper solution sent by organisers after finishing the contest. The classifier was learnt on the whole training set.

The threshold θ was chosen to be equal to 0.55. This value was obtained by repeatedly fitting the classifier for values between 0.5 and 0.6 and choosing

Table 2. Values of the score (accuracy of classifier) for task 1.

Classifier	Score			time [sec.]
	training	validation	solution	
Majority vote	0.7292	0.7285	0.7331	0.021
Simple classification	0.7733	0.7661	0.7669	0.060
Trend prediction	0.7729	0.7696	0.7690	1.793
Autoregression	0.7781	0.7717	0.7687	7.842

the best results (validation score). The value of the score is (in case of task 1) the accuracy of the classifier, i.e. the fraction of correctly classified observations. The results are presented in Table 2. A “majority vote” classifier is also included which always assigns values from the larger class (short visit in this case – results of this classifier coincide of course with values presented in Table 1). We also present computational time for each classifier (calculated on the notebook with 512MB RAM and 2.13GHz Athlon processor), which includes training and classification of the testing set, but which does not include reading both files (training and testing) from disk into memory (it took additional 6.409 seconds).

Notice that although regression results were sent for the contest (the highest validation score), the best results on the testing set were achieved by the trend prediction approach (the highest solution score). All the models were relatively fast, especially majority vote and simple classification. Also all the models, apart from majority vote, have very similar results (almost no change in the score value). This suggests that any other condition attributes based on timestamp or previous visit length are hardly informative. This would also suggest choosing the simplest (parsimonious) model which is simple classification.

In comparison to the global models presented in section 3, the simple classification is only slightly worse than the best result obtained by enhanced global model, while trend prediction and autoregression seem to be better. The user models are simpler (less attributes taken into account), more stable, easier in parameterisation, and for these reasons, much more faster.

4.2 Solution to Task 2

The second task concerns predicting a list of the 3 most probable categories (i.e. the 3 first page categories on the visit path) during a given visit of a given user. A specific score function was defined by the organisers in order to quantitatively measure the goodness of prediction. Assume that $y = (y_1, \ldots, y_m)$ is a sequence of m visited categories. Moreover let $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}))$ be the sequence of the 3 most probable categories predicted by the classifier based on some predictor vector $\mathbf{x}$. In case of y as well as $f(\mathbf{x})$, according to the challenge rules, we assume that when a category appears more than once in the sequence, each time it is regarded as new, different category. The loss function is defined as

Table 3. Values of the score for task 2 and 3

Task	Score			time [sec.]
	training	validation	solution	
2	5.6830	5.6021	5.5606	30.124
3	6.5041	6.3747	6.3147	30.545

negative score and can be written in the following way:

$$L(y, f(\mathbf{x})) = -\sum_{j=1}^{m}\sum_{k=1}^{3} s(j,k) I(y_j = f_k(\mathbf{x})) \tag{7}$$

where

$$s(j,k) = \max\{1, \min\{6-j, 6-k\}\} \tag{8}$$

is the single score value and $I(x)$ is the indicator function equal to 1 if x is true, 0 otherwise. The risk of the classifier has the following form:

$$R(f) = \int \sum_{y} L(y, f(\mathbf{x})) P(y|\mathbf{x}) \mathrm{d}P(\mathbf{x}) \tag{9}$$

In order to find the optimal decision for a fixed predictor $\mathbf{x}$, we must minimise the risk point-wise, i.e. minimise $\sum_y L(y, f(\mathbf{x}))P(y|\mathbf{x})$. Since the probabilities $P(y|\mathbf{x})$ are unknown, we use empirical risk minimisation (3), so that we minimise the loss function (7) on the dataset. However, one can show, that estimating the probabilities $P(y|\mathbf{x})$ by frequencies would lead exactly to the same result.

Thus, the optimal decision function is obtained simply by choosing for each $\mathbf{x}$ value $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}))$, which minimises the empirical risk. However, one does not need to go through the whole dataset for each combination of values of $f(\mathbf{x})$. It is enough to calculate three aggregated coefficients for each category for a given $\mathbf{x}$ and then choosing $f(\mathbf{x})$ is independent of the number of visits.

There is still $\mathbf{x}$ to be chosen to make the model complete. Motivated by our exploratory analysis (see section 2) and the results for task 1, we decided to use parsimonious model taking into account only one predictor - user id number. Thus, the classifier is constant on every visit of the same user, so that corresponds to user's granulation described in section 3. Such model has the advantage of being stable and having small variance. The results obtained using the model are presented in Table 3. They indicate that the model, despite its simplicity, has fairly good accuracy.

The time complexity of the method is linear with the number of visits. The memory complexity is (not including the dataset itself) linear with the number of users.

4.3 Solution to Task 3

The third task was an extension of the second task. Apart from giving a list of the most probable categories in a visit, it concerns giving a range of number of page views in each category. Assume that $y = ((y_1, t_1), \ldots, (y_m, t_m))$ is a sequence of m visited pairs (`category`, `#pages_range`) and

$$f(\mathbf{x}) = ((f_1^c(\mathbf{x}), f_1^t(\mathbf{x})), (f_2^c(\mathbf{x}), f_2^t(\mathbf{x})), (f_3^c(\mathbf{x}), f_3^t(\mathbf{x})))$$

is a sequence of 3 most probable pairs (`category`, `#pages_range`) predicted by the classifier based on some predictor vector $\mathbf{x}$. Then, the loss function is:

$$L(y, f(\mathbf{x})) = -\sum_{j=1}^{m} \sum_{k=1}^{3} s(j,k) I(y_j = f_k^c(\mathbf{x})) + I(t_j = f_k^t(\mathbf{x})) \tag{10}$$

Similarly as in the case of task 2, we minimised the empirical risk (3) to obtain the decision function. Again, only one predictor $\mathbf{x}$ was chosen – user id number. The results are shown in Table 3. The time and memory complexity of the method is the same as for task 2.

5 Conclusion

After an intensive exploratory data analysis phase we examined a few approaches to the contest tasks and chose the solution that is simple, but effective and theoretically well-founded. We found this choice optimal in the context of the limited time. All of our algorithms scale well with large data and have linear time complexity, which is the smallest possible complexity for such problems, since reading the dataset is already linear in its size. The memory complexity never exceeds linear rate and grows linearly with the number of users, not visits.

We believe that there is still some improvement of the result possible and plan to explore it in a subsequent work. We plan to further experiment with our attributes and clustering users in order to obtain larger granules which in turn make it possible to compute estimates for models with richer structure.

References

1. Berger, J.: *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag, New York (1993)
2. Dasu, T., Johnson, T.: *Exploratory Data Mining and Data Cleaning.* Wiley (2003)
3. Duda, R., Hart, P., Stork, D.: *Pattern Classification.* Wiley-Interscience (2000)
4. ECML/PKDD'2007 Discovery Challenge: User's behaviour prediction `http://www.ecmlpkdd2007.org/challenge/` (2007)
5. Hamilton, J. D.: *Time Series Analysis.* Princeton University Press (1994).
6. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning.* Springer (2003)
7. R Development Core Team: *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, `http://www.R-project.org` (2005)
8. Wald, A.: *Statistical decision functions.* Wiley (1950)
9. Witten, I., Frank, E.: *Data Mining: Practical machine learning tools and techniques, 2nd Edition.* Morgan Kaufman (2005)