Collaborative Filtering Using Interval Estimation Naïve Bayes

V. Robles[1], P. Larrañaga[2], J.M. Peña[1], O. Marbán[3], J. Crespo[3], and M.S.Pérez[1]

[1] Department of Computer Architecture and Technology, Technical University of Madrid, Madrid, Spain
{vrobles,jmpena,mperez}@fi.upm.es
[2] Department of Computer Science and Artificial Intelligence, University of the Basque Country, San Sebastián, Spain
ccplamup@si.ehu.es
[3] Department of Computer Science, University Carlos III of Madrid, Madrid, Spain
{omarban,jcrespo}@inf.uc3m.es

Abstract. Personalized recommender systems can be classified into three main categories: content-based, mostly used to make suggestions depending on the text of the web documents, collaborative filtering, that use ratings from many users to suggest a document or an action to a given user and hybrid solutions. In the collaborative filtering task we can find algorithms such as the naïve Bayes classifier or some of its variants. However, the results of these classifiers can be improved, as we demonstrate through experimental results, with our new semi naïve Bayes approach based on intervals. In this work we present this new approach.[1]

1 Introduction

Nowadays, the volume of information on the World Wide Web has been arising in an exponential way [2] and the number of pages will be even bigger in a near future. Therefore, people need technology to avoid the problems that this amount of information generates. Due to it, web mining has arose to facilitate the user's information access.

Web mining is a broad term that has been used to refer to the process of information discovery from sources in the Web (Web content), to the process of discovery of the structure of the web servers (Web structure) and to the process of mining for user browsing and access patterns through logs analysis (Web usage) [4].

Three phases [6] are identified in Web using mining: Collecting and preprocessing of data, pattern discovery and pattern analysis.

In the first phase data can be gathered from different sources [9]. On one hand, we can collect data directly from the user. We can ask information through surveys, polls or asking directly to the user. On the other hand, we can obtain information without the user intervention. There are two different ways we can use to gather this user's unconscious information:

[1] The work presented in this paper has been partially supported by UPM project RT-Webp – ref.14495

E. Menasalvas et al. (Eds.): AWIC 2003, LNAI 2663, pp. 46–53, 2003.
© Springer-Verlag Berlin Heidelberg 2003

- Direct pursuit maintains a registry of user's activities, normally located on the client-side such as cookies, beacons, etc.
- Indirect pursuit using log files. Log files can be used in order to find association patterns, sequential patterns and trends of Web accesses.

Therefore, information can be collected and preprocessed depending on its usage. Thus, it would be used to consider the user's preferences, to generate new pages with greater acceptance, for recommendations in the selection of a new product, to diagnose his desires of an information, etc.

In the second phase, once the data have been preprocessed, there are several approaches that can be performed depending on the needs. Many approaches have focused on applying intelligent techniques to provide user personal recommendations. These personalized recommender systems can be classified into three main categories:

1. Content Based Filtering
   In content-based filtering, a user's preference model is constructed for the individual based upon the user's ratings and descriptions (usually, textual expressions) of the rated items. Such systems try to find regularities in the descriptions that can be used to distinguish highly rated items from others. There are three kinds of content based filtering systems: Pure information systems [17], survey or polling or social information systems [16] and content-wise examination information systems [14].
2. Collaborative Filtering
   The main idea of collaborative filtering is to recommend new items of interest for a particular user based on other users' opinions. A variety of collaborative filtering algorithms have been reported and their performance has been evaluated empirically [1] [13] [17] [10]. These algorithms are based on a simple intuition: predictions for a user should be based on the preference patterns of other people who have similar interest.
3. Unified or Hybrid Solution
   Several authors [15] suggest methods for a hybrid solution. They present a combination between collaborative filtering and content-based filtering. They propose a generative probabilistic model for combining collaborative and content-based recommendations in a normative manner.

In this work we present a new approach to collaborative filtering with naïve Bayes. We have developed a new semi naïve Bayes approach based on intervals. This new approach outperforms the simple naïve Bayes classifier and other variants specifically defined for collaborative filtering [10]. We evaluated this algorithm using a database of Microsoft Anonymous Web Data from the UCI repository [12].

The outline of this paper is as follows. Section 2 presents the state of the art in collaborative filtering with naïve Bayes. Section 3 presents a new semi naïve Bayes approach, interval estimation naïve Bayes. Section 4 illustrates the results obtained with the UCI dataset. Section 5 gives the conclusions and suggests further future work.

2 Naïve Bayes Classifiers in Collaborative Filtering

The naïve Bayes classifier [3] [5] is a probabilistic method for classification. It can be used to determine the probability that an example belongs to a class given the values of variables. The simple naïve Bayes classifier is one of the most successful algorithms on many classification domains. In spite of its simplicity, it is shown to be competitive with other complex approaches specially in text categorization and content based filtering.

This classifier learns from training data the conditional probability of each variable X_k given the class label c_i. Classification is then done by applying Bayes rule to compute the probability of C given the particular instance of $X_1, \ldots, X_n$,

$$P(C = c_i | X_1 = x_1, \ldots, X_n = x_n) \tag{1}$$

As variables are considered independent given the value of the class this probability can be calculated as follows,

$$P(C = c_i | X_1 = x_1, \ldots, X_n = x_n) \propto P(C = c_i) \prod_{k=1}^{n} P(X_k = x_k | C = c_i) \tag{2}$$

This equation is well suited for learning from data, since the probabilities $P(C = c_i)$ and $P(X_k = x_k | C = c_i)$ can be estimated from training data. The result of the classification is the class with highest probability.

In [10] Pazzani and Miyahara two variants of the simple naïve Bayes classifier for collaborative filtering are defined:

1. **Transformed Data Model** After selecting a certain number of features, absent or present information of the selected features is used for predictions. That is:

$$P(C = c_i | S_1 = s_1, \ldots, S_n = s_n), \tag{3}$$

   where $r \leq n$ and $S_i \in X_i, \ldots, X_n$. S_i variables are selected using a theory based approach to determinate the most informative features. This is accomplished by computing the expected information gain that the presence of absence of a variable gives toward the classification of the labelled items.
2. **Sparse Data Model** In this model, authors assume that only known features are informative for classification. Therefore, only known features are used for predictions. That is:

$$P(C = c_i | X_1 = 1, X_3 = 1, \ldots, X_n = 1) \tag{4}$$

3 A New Semi-naïve Bayes Approach: Interval Estimation Naïve Bayes

We propose a new semi naïve Bayes approach named interval estimation naïve Bayes. In this approach, in spite of calculate the punctual estimation of the conditional probabilities from data, as simple naïve Bayes does, we calculate interval estimations. After that, by searching for the best combination of values into these intervals, we seek to break the assumption of independence among variables in the simple naïve Bayes. Although we have used this algorithm for collaborative filtering, it can be used in the same problems we use the simple naïve Bayes.

The approach is based on two different steps:

In the **first step**, each parameter is estimated by intervals. Thus, we consider the next interval for the conditional probabilities $\hat{p}^r_{k,i} = \hat{P}(X_k = x^r_k | C = c_i)$

$$\left(\hat{p}^r_{k,i} - z_\alpha \sqrt{\frac{\hat{p}^r_{k,i}(1 - \hat{p}^r_{k,i})}{N}}, \hat{p}^r_{k,i} + z_\alpha \sqrt{\frac{\hat{p}^r_{k,i}(1 - \hat{p}^r_{k,i})}{N}} \right) \tag{5}$$

where

r is the possible values of the variable X_k
$\hat{p}^r_{k,i}$ is the punctual estimation of the conditional probability $P(X_k = x^r_k | C = c_i)$
z_α is the $(1 - \alpha)$ percentile in the $\mathcal{N}$(0,1)
N is the number of cases in dataset.

In the **second step** we make a heuristic search to obtain the best combination of conditional probabilities that maximize a predefined evaluation function. The values for each of this conditional probabilities are found inside each corresponding interval. The evaluation function depends on each specific problem. It does not matter which algorithms we use for the search: Genetic algorithms, simulated annealing, tabu search, etc.

It is important to emphasize three key aspects in interval estimation naïve Bayes:

- In the heuristic search, we take into account all the conditional probabilities at the same time, searching for the best combination. This means that we are breaking the assumption of independence among the variables.
- As validation method we are using leave-one-out cross validation. This method guarantees that no overfitting will occur for these data.
- Normally, the evaluation function will be the percentage of successful classified. However, sometimes, as occurs in this approach, we need a different evaluation function.

Figure 1 shows the pseudocode of interval estimation naïve Bayes.

To make the heuristic search in this work we have used EDAs – estimation of distribution algorithms –. EDAs [11,8] are non-deterministic, stochastic heuristic search strategies that form part of the evolutionary computation approaches, where number of solutions or individuals are created every generation, evolving once and again until a satisfactory solution is achieved. In brief, the characteristic that most differentiates EDAs from other evolutionary search strategies such as GAs is that the evolution from a generation to the next one is done by estimating the probability distribution of the fittest individuals, and afterwards by sampling the induced model. This avoids the use of crossing or mutation operators, and the number of parameters that EDAs requires is reduced considerably.

```
Interval Estimation naïve Bayes
  Calculate every interval estimation $\hat{p}^r_{k,i}$ for $P(X_k = x^r_k | C = c_i)$ from data
    $\left(\hat{p}^r_{k,i} - z_\alpha\sqrt{\frac{\hat{p}^r_{k,i}(1-\hat{p}^r_{k,i})}{N}}, \hat{p}^r_{k,i} + z_\alpha\sqrt{\frac{\hat{p}^r_{k,i}(1-\hat{p}^r_{k,i})}{N}}\right)$
    For $k = 1, \ldots, n \wedge i = 1, \ldots, \|C\| \wedge r = 1, \ldots, \|X_k\|$

  Make a heuristic search to obtain the combination of values that
  maximize a predefined evaluation function
```

Fig. 1. Pseudocode for interval estimation naïve Bayes

4 Experimentation

4.1 Dataset

For the evaluation of our new approach (internal estimation naïve Bayes) for collaborative filtering we have used a dataset of Microsoft Anonymous Web Data from the UCI repository [12].

This data was created by sampling and processing the `www.microsoft.com` logs. The data records the use of `www.microsoft.com` by 32711 anonymous, randomly-selected users. For each user, the data lists all the areas that the user visited in a one week timeframe. Thus, each instance in the dataset represents an anonymous user and each attribute is an area of the `www.microsoft.com` web site. There are a total of 294 areas.

This dataset shows instance independence among each of the records of the database. Taking this into account our leave one out evaluation method will not have overtraining problem.

In this case we have a very sparse data, so areas visited are explicit while non-visited are implicit. Thus, an attribute will have a value 1 if the area has been visited, and a value 0 in other case.

Our task is to predict the areas of `www.microsoft.com` that a user will visit, based on data on what other areas he or she visited.

After the learning and validation we will evaluate prediction accuracy, learning time and speed of predictions. The accuracy will be measured via the leave one out method [7].

4.2 Measuring Prediction Accuracy – Evaluation Function

Most of the times quality of classifiers can be measured by the percentage of successful predictions. However, this measure is not a good idea in this dataset due to the lack of balance between positive and negative cases. Let's see an example.

Suppose that from the 32711 users only 1000 have visited an especific page and the next confusion matrix (see table 1) coming from our classifier, where all the users have been classified as non potential visitors.

Table 1. An example of a confusion matrix

	Classified as	
Real	*0*	*1*
0	31711	0
1	1000	0

Table 2. A generic confusion matrix

	Classified as	
Real	*0*	*1*
0	a	b
1	c	d

Everybody can appreciate that this classifier is really bad, however the accuracy is $31711/32711 = 96.94\%$. That means that % of successful predictions is not a good reference. As we can see in table 3 simple naïve Bayes has this problem in collaborative filtering.

Thus, given a generic confusion matrix (see Table 2), the measure we will use is

$$\left(\frac{a}{a+b} + \frac{d}{c+d}\right)/2 \tag{6}$$

This measure is much better and realistic because we are evaluating the percentage of visitors classified as visitors and the % of non-visitors classified as non-visitors independently. Then we calculate the average of both.

This formula (6) will be used as the evaluation function in interval estimation naïve Bayes. Besides, for calculating the probabilities we have used the idea exposed in the variant Sparse Data Model (see equation 4) defined by Pazzani and Miyahara and only known features are used for predictions.

4.3 Experimental Results

We have run the algorithm in the 18 more visited pages. The most visited page has 10836 visitors and the less visited has 1087 visitors. This range of visitors is enough to analyze the behavior of our new approach.

Table 3 contains the experiment results for simple naïve Bayes, the variant Sparse Data Model defined by Pazzani and Miyahara and interval estimation naïve Bayes approaches. The first two columns in the table have the identifier of the area and the number of visitors in that area. The next two columns have the results for the simple naïve Bayes. The first one with the symbol % is the percentage of successful predictions and the second one the value of the evaluation function. After that we have two columns with the results for the variant Sparse Data Model and the last two columns with the results for interval estimation naïve Bayes.

We must remember that the most significant columns are those with the values of the evaluation function.

Evaluation of interval estimation naïve Bayes:

Table 3. Experiment results for Interval Estimation naïve Bayes

		simple NB		**PazzaniNB**		**IENB Max(feval)**	
Area	*Visitors*	*%*	*f_eval*	*%*	*f_eval*	*%*	*f_eval*
'1008'	10836	72.34	63.03	70.94	70.59	71.23	71.73
'1034'	9383	74.47	64.72	72.35	54.56	72.38	55.93
'1004'	8463	72.31	54.12	72.48	53.95	72.01	55.75
'1018'	5330	86.70	72.08	78.85	75.60	79.52	77.49
'1017'	5108	83.94	62.39	76.68	68.96	77.92	71.15
'1009'	4628	88.45	71.16	84.32	72.11	85.21	73.12
'1001'	4451	88.24	71.75	86.31	77.49	86.42	78.58
'1026'	3220	92.22	71.41	91.78	83.68	91.84	85.84
'1003'	2968	90.03	72.21	86.58	78.39	86.95	79.54
'1025'	2123	91.99	67.24	91.85	55.53	91.88	57.53
'1035'	1791	94.15	75.97	89.22	88.64	88.60	89.54
'1040'	1506	94.89	68.06	92.71	75.45	92.61	79.36
'1041'	1500	94.89	71.74	89.36	79.61	89.44	80.77
'1032'	1446	95.98	57.27	95.98	57.23	96.02	58.47
'1037'	1160	94.22	68.40	90.27	79.26	90.38	80.99
'1030'	1115	94.91	65.26	89.61	71.69	89.24	73.51
'1038'	1110	95.54	73.52	92.55	80.58	92.57	83.66
'1020'	1087	95.40	62.71	92.88	68.60	92.80	70.97
Average		88.93	**67.39**	85.82	**71.77**	85.95	**73.55**

- Prediction accuracy: About the evaluation function the results are clear. The variant of Pazzani and Miyahara outperforms simple naïve Bayes in 4.38% and our new approach, interval estimation naïve Bayes, outperforms the variant in 1.78% and the simple naïve Bayes in 6.16%.
- Learning time: simple naïve Bayes and the variant of Pazzani and Miyahara have a really short learning time. Few seconds are enough for the learning. However, interval estimation must make a heuristic search of the conditional probabilities. The evaluation of each individual takes the same time than the evaluation of the simple naïve Bayes classifier. In conclusion, as thousand of evaluation are needed in interval estimation, the learning time is some hours. However, as learning should be done only once, this is not a relevant issue.
- Speed of predictions: The speed of the predictions is exactly the same for the three algorithms.

5 Conclusion and Further Work

In this work we have presented a new semi naïve Bayes approach named interval estimation naïve Bayes. We have used this new approach for collaborative filtering. Experimental results shown that our approach outperforms the simple naïve Bayes and other variants specifically defined for collaborative filtering.

As this is the first time we use this approach for collaborative filtering many issues remain for future research. For instance, it is possible to change the objective of the heuristic search. We can try to maximize the area under the ROC curve. Another viable idea is to combine interval estimation naïve bayes with a feature subset selection. On a first phase it is possible to make a subset selection, and on a second phase to apply interval estimation to the previous results.

References

1. J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithm for collaborative filtering. Technical report, Microsoft Corporation, Redmond,WA 98052, May 1998. One Microsoft Way.
2. P. Bruemmer. Google: Search technology for the millennium. Search Engine Guide, February 2002. Every 28 days, Google indexes 3 billion Web Documents.
3. R. Duda and R. Hart. *Pattern Classification and Pattern Analysis*. Wiley, New York, 1973.
4. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann publisher, 2001.
5. D.J. Hand and K. Yu. Idiot's Bayes - not so stupid after all? *International Statistical Review*, 69(3):385–398, 2001.
6. M. Deshpande J. Srivastava, R. Cooley and P. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1:12–23, 2000.
7. R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, pages 1137–1145, 1995.
8. P. Larrañaga and J. A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publisher, 2001.
9. O. Marbán, E. Menasalvas, C. Montes, and J. Segovia. *Book: E-Commerce and Intelligent Methods in the Studies in Fuzziness and Soft Computing Series*, chapter CRM in e-Business: a client's life cycle model based on a neural network, pages 61–77. Springer-Verlag, 2002.
10. K. Miyahara and M.J. Pazzani. Collaborative filtering with the simple Bayesian classifier. In *Pacific Rim International Conference on Artificial Intelligence*, pages 679–689, 2000.
11. H. Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5:303–346, 1998.
12. P. M. Murphy and D. W. Aha. UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/`, 1995.
13. P. Resnick, I. Neophytos, S. Mitesh, P. Bergstrom and J. Rieldd. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW94: Conference on Computer Supported Cooperative Work*, pages 175–186, 1994.
14. M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27:313–331, 1997.
15. A. Popescul, L. Ungar, D. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *17th Conference on Uncertainty in Artificial Intelligence*, pages 437–444, Seattle, Washington, August 2–5 2001.
16. G. Salton and M. J. McGill. Introduction to modern information retrieval, 1983.
17. U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217, 1995.