Improvement of naïve Bayes Collaborative Filtering using Interval Estimation

V. Robles(1), P. Larrañaga(2), E. Menasalvas(1), M. S. Pérez(1), V. Herves(1)
(1) School of Computer Science, Technical University of Madrid, Madrid, Spain.
{vrobles, ernes, mperez}@fi.upm.es, {vanessa}@laurel.datsi.fi.upm.es
(2) School of Computer Science, University of the Basque Country, San Sebastián, Spain.
ccplamup@si.ehu.es

Abstract

Recommender systems emerged to help users choose among the large amount of options that e-commerce sites offer. Collaborative filtering is one of the most successful recommender techniques. In this paper we propose an approach to collaborative filtering based on the simple Bayesian classifier. We propose a method of increasing the efficiency of naïve bayes by applying a new semi naïve Bayes approach based on interval estimation. To evaluate our algorithm we use a database of Microsoft Anonymous Web Data from the UCI repository. Our empirical results show that our proposed interval based naïve Bayes approach outperforms typical naïve bayes[1].

1 INTRODUCTION

The Web increasingly becomes a retailing channel of choice for millions of users. Nevertheless, e-commerce sites offer a large amount of options that can detract user to use the site. Automated methods are needed to provide users information to efficiently locate and retrieve information they want in order to rescue the one to one relationship that is in danger in e-commerce sites. Thus, recommender systems have emerged to assist business in treating each customer individually.

Much research has been done in applying intelligent techniques to provide user personal recommendations as an approach to solve the stated problem. One of the earliest and most successful recommender techniques is collaborative filtering [15] [9]. Collaborative filtering is based on the assumption that finding similar users to a new one and examining their usage patterns leads to useful recommendations being made for the newcomer.

The task of collaborative filtering is to predict preferences of a user given a database of preferences usually expressed as numerical scores of other users. In this setting, the data is a collection of pairs of objects where each pair consists of a person and an item (i.e. Web pages, products). In the collaborative filtering task we are interested in making predictions on how likely a person is to be interested in a particular item (page in our case) given information about their and other user's historical behaviors or interests.

Memory-based collaborative filtering algorithms maintain a database of previous users preferences and perform certain calculations on the database each time a new prediction is needed. The most common representatives are neighbor-based algorithms where a subset of users most similar to an active user is chosen and a weighted average of their scores is used to estimate preferences of an active user on other items [9] [17]. In contrast, model based algorithms first develop a description model from a database and use it to make predictions for a user. In these approaches, collaborative filtering is perceived as a classification task. Published systems of this type include Bayesian networks [2] and classification-based algorithms [1]. Neighbor-based collaborative filtering algorithms are known to be superior than models based on terms of accuracy but their high latency can be a drawback in systems with a large number of request to be preprocessed in real time. Moreover, as the number of items evaluated by an active user decreases the prediction accuracy on neighbour, systems deteriorates dramatically. In [14] a simple Bayesian classifier for collaborative filtering is presented. The proposed model is applied both to user-based collaborative and item-based recommendations. According to the authors, the Bayesian classifier outperforms typical correlation-based collaborative filtering algorithms.

The majority of the approaches to the problem of collaborative filtering assume a data representation for each object and focus on a single relationship between the objects. In [5], the authors examine a richer model that makes it possible to reason about many different relations between the objects. This is specially important, in the context of the Web

[1]The research has been partially supported by Universidad Politécnica de Madrid under project WEBP-RT.

where there is often more relational information than the simple person-item relationship. They also focus on model-based methods and review the two side clustering model for collaborative filtering [7]. In [3] a new collaborative filtering algorithm based on factor analysis is presented. The algorithm is based on a probabilistic model of user choice and on a probabilistic sound approach to dealing with missing data. The main drawback of the algorithm is that the model cannot be applied over binary recommendation system.

In spite of all the research that has been conducted on collaborative filtering in the last years, there are two challenges to solve [16]. The first problem of collaborative system is the scalability of the algorithm. Due to the nature of e-commerce and in order to obtain a good recommendation millions of records have to be explored. The second challenge is related to the quality of the recommendation being done to the customer. The recommendation should be valuable. Like any other prediction systems in recommender systems the final decision can fall in one of the next categories: false negatives if the decision is not recommending a product that the user would have liked or false positive when the systems decision is to recommend a products that the user does not like. In e-commerce the most important error to avoid is false positive as the loyalty of consumer is questionable if the site recommends him to acquire something he does not like.

In this work we present a new approach to collaborative filtering based on naïve Bayes. The new approach is named Interval Estimation naïve Bayes (IENB). We propose to calculate the parameters of naïve Bayes with a confidence interval and later to make an heuristic search for the best naïve Bayes classifier that can be extracted from the intervals. We evaluated this algorithm using a database of Microsoft Anonymous Web Data from the UCI repository [13]. This new approach outperforms the simple naïve Bayes classifier and other variants specifically defined for collaborative filtering [11]. However, the learning time of our approach takes two or three hours. Therefore, depending on the task to be made, this algorithm will be useful or not.

The rest of the paper is organized as follows. In section 2 the naïve approach to collaborative filtering is briefly outlined. Section 3 is a brief introduction to statistical inference. Section 4 contains the statistical demonstration of parameter estimation in naïve Bayes and Interval Estimation naïve Bayes. Section 5 introduces the new approach Interval Estimation naïve Bayes. Section 6 shows how the new approach outperforms the previous approaches. Section 7 presents the conclusion and the future lines of work.

2 Naïve Bayes classifiers in collaborative filtering

The naïve Bayes classifier [4] [6] 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 $X_1, \ldots, X_n$,

$$P(C = c_i | X_1 = x_1, \ldots, X_n = x_n)$$

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

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

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

In Pazzani and Miyahara [11] 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:

$$P(C = c_i | S_1 = s_1, \ldots, S_n = s_n), \tag{2}$$

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 or 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:

$$P(C = c_i | X_1 = 1, X_3 = 1, \ldots, X_n = 1) \tag{3}$$

3 Statistical Inference

Statistical inference studies a collection of data based on a sample of these ones. This sample represents the part of population considered in the analysis. Amongst other things, statistical inference studies the problem known as "estimation problem". The estimation of a parameter involves both the usage of sampled data and some statistical tools. There are two ways of accomplishing this task: point estimation and interval estimation. In general, a **parameter estimator** θ is some random variable $\hat{\theta}$ that is expressed according to a random sample and whose goal is to approximate the value of θ, $\hat{\theta}(X_1, \ldots, X_n)$.

3.1 Point Estimation

Point estimation uses a sample with the aim of assigning a single value to a parameter. This value must represent a good presumption of the real value. The assigned value is called a point estimation. Point estimation methods are used in order to obtain estimators that provide good features. Specifically, maximum likelihood method and method of moments are used in this context. Only the first one is described, because it is used in naïve Bayes.

Let X be a random variable whose probability distribution is

$$f(x;\theta)$$

A simple random sample of size n, $X_1, X_2, \ldots, X_n$ has as joint probability distribution

$$f(x_1, x_2, \ldots, x_n; \theta)$$

This function depends on $n+1$ variables. Nevertheless, it may be calculated only in terms of θ, if the x_k values are fixed. This function of θ is known as likelihood function.

In this context, it is interesting to formulate that if a sample is given with particular x_k values, a possible parameter estimation is the one that maximizes the likelihood function.

$$x_1, \ldots, x_n \; given \Rightarrow Likelihoodfunction \equiv \\ V(\theta) = f(x_1, \ldots, x_n; \theta)$$

To maximize a function is equivalent to maximize the logarithm of a function, because the logarithm function is strictly monotonically increasing. For this reason, it is possible to calculate the maximum value through the derivative of the logarithm of the likelihood function with respect to θ, taking as maximum likelihood estimator the one whose derivative is equal to zero.

$$\frac{\partial \log V}{\partial \theta}(\hat{\theta}_{\mathcal{MV}}) = 0 \quad (4)$$

3.2 Interval Estimation

Interval estimation determines a possible range of values (values interval) and their associated probabilities for a parameter value. This parameter is often a proportion in the case of dichotomous variables and the average and the variance in the case of gaussian variables.

This technique calculates for each sample an interval that probably contains the parameter. This interval is called confidence interval.

Obviously, this technique does not always achieve the right result. The probability with which an interval includes a parameter is known as confidence level. Significance level is the probability for the error of this fact.

4 Parameter estimation in naïve Bayes and Interval Estimation naïve Bayes

This section contains a formal analysis about the accurate estimation of parameters in naïve Bayes and the confidence intervals in Interval Estimation naïve Bayes.

4.1 Point estimation of parameters in naïve Bayes

In this section it is formally explained how to learn the parameters in a data modeling naïve Bayes. The obtained results are intuitive.

Let $\mathcal{D} = \{x^\mu, \mu 1, \ldots, N\}$ be a dataset of binary attributes, that is $x_k^\mu \in \{0,1\}$. Every instance x^μ has associated a class label c_i^μ. Based on the class label it is possible to split the instances into different classes: $X^i = \{x | x \; is \; in \; class \; c_i\}$. We only consider the usage of two classes, which is called a Bernoulli process. The case with more classes is also straightforward and is called multinomial process.

For each class the values $P(X_k = 1|C = c_i) \equiv \theta_k^i$ must be estimated. The other probability, $P(X_k = 0|C = c_i)$ is given by the normalization requirement $P(X_k = 0|C = c_i) = 1 - P(X_k = 1|C = c_i) = 1 - \theta_k^i$. Making use of the standard assumption that data are generated in the same manner and independently, the probability with which the model generates the dataset X^i is,

$$P(X^i) = \prod_{\mu \in c_i} P(x^\mu | C = c_i)$$

Taking into account the assumption of independence,

$$P(X_1 = x_1, \ldots, X_n = x_n | C = c_i) = \\ \prod_k P(X_k = x_k | C = c_i) \prod_k (\theta_k^i)^{x_k} (1 - \theta_k^i)^{1 - x_k}$$

It is important to remember that each term in the product above, x_k, is zero or one. Therefore, only one factor will contribute to the calculus. If $x_k = 1$ the factor θ_k^i will contribute and if $x_k = 0$ the factor that will contribute is $1-\theta_k^i$. Joining the two previous formulas and taking logarithms,

$$\log(\theta^i) = \sum_{k,\mu}[x_k^\mu \log \theta_k^i + (1 - x_k^\mu)\log(1-\theta_k^i)]$$

Optimizing with respect to θ_k^i, that is, differentiating with respect to θ_k^i and making it equal zero,

$$\hat{P}(X_k = x_k | C = c_i) = \frac{num\ times\ x_k = 1\ for\ class\ c_i}{num\ times\ class\ is\ c_i} \tag{5}$$

With a very similar development of maximum likelihood estimation, the following result will be obtained

$$\hat{P}(C = c_i) = \frac{num\ times\ class\ is\ c_i}{num\ of\ total\ instances} \tag{6}$$

4.2 Interval estimation of parameters in IENB

In the case of IENB what we want is to calculate the confidence intervals for the parameters.

Given, $X_1, X_2, \ldots, X_n \rightsquigarrow Ber(p)$, we want to estimate the parameter p. The most natural way to make this estimation consists in defining the sum of the variables, which generate a binomial distribution,

$$X = X_1 + \ldots + X_n \rightsquigarrow \mathcal{B}(n, p)$$

and taking as its estimator the random variable

$$\hat{p} = \frac{X}{n}$$

This means that as estimation of p we are taken the proportion of success in the n experiments.

The success distribution is binomial and it may be approximated by a normal distribution when the sample size is big, and p is a value not very close to zero or one.

$$X \rightsquigarrow B(n,p) \Rightarrow X \overset{\approx}{\rightsquigarrow} \mathcal{N}(np, npq)$$

The estimator $\hat{p}$ is a scale transformation of X, consequently,

$$\hat{p} = \frac{X}{n} \overset{\approx}{\rightsquigarrow} \mathcal{N}(0,1) \Rightarrow \frac{\hat{p} - p}{\sqrt{\frac{pq}{n}}} \approx \mathcal{Z} \rightsquigarrow \mathcal{N}(0,1)$$

This expression is quite difficult to compute, so it may be substitute for the next expression,

$$\frac{\hat{p} - p}{\sqrt{\frac{\hat{p}\hat{q}}{n}}} \approx \mathcal{Z} \rightsquigarrow \mathcal{N}(0,1)$$

To find a confidence interval with level of significance α for p we must consider, with a confidence $1 - \alpha$,

$$|Z| \leq z_\alpha$$

that is,

$$\frac{|\hat{p} - p|}{\sqrt{\frac{\hat{p}\hat{q}}{n}}} \leq z_\alpha$$

so,

$$|\hat{p} - p| \leq z_\alpha \sqrt{\frac{\hat{p}\hat{q}}{n}}$$

that can be summarized as,

$$p = \hat{p} \pm z_\alpha \sqrt{\frac{\hat{p}\hat{q}}{n}} \tag{7}$$

having a confidence of $1 - \alpha$

5 Interval Estimation naïve Bayes – IENB

We propose a new semi naïve Bayes approach named interval estimation naïve Bayes (IENB). In this approach, instead of calculating the point estimation of the conditional probabilities from data, as simple naïve Bayes does, we calculate confidence intervals. 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 may be used in the same problems we use the simple naïve Bayes.

There are three main important aspects in IENB algorithm:

- **Calculation of confidence intervals**

  Given the dataset, the first step we must do is to calculate the confidence intervals for each conditional probability and for each class probability. For the calculation of the intervals we must first compute the point estimations of these same parameters (see section 4.1).

  In this way, each conditional probability that must be estimated from the dataset $\hat{p}_{k,r}^i = \hat{P}(X_k = x_k^r | C = c_i)$ must be computed with the next confidence interval, as we demonstrate in section 4.1.

$$\left(\hat{p}^i_{k,r} - z_\alpha\sqrt{\frac{\hat{p}^i_{k,r}(1-\hat{p}^i_{k,r})}{N}}, \hat{p}^i_{k,r} + z_\alpha\sqrt{\frac{\hat{p}^i_{k,r}(1-\hat{p}^i_{k,r})}{N}}\right) \tag{8}$$

$For\ k = 1,\ldots,n \wedge i = 1,\ldots,\|C\| \wedge r = 1,\ldots,\|X_k\|$, where,

r $\equiv$ possible values of variable X_k

$\hat{p}^i_{k,r} \equiv$ point estimation of the conditional probability $P(X_k = x^r_k | C = c_i)$

$z_\alpha \equiv (1-\alpha)$ percentil in the $\mathcal{N}(0,1)$

$N \equiv$ is the number of cases in dataset

Also, in a similar way, the probabilities for the class values $\hat{p}_i = \hat{P}(C = c_i)$ are estimated with the next confidence interval,

$$\left(\hat{p}_i - z_\alpha\sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{N}}, \hat{p}_i + z_\alpha\sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{N}}\right) \tag{9}$$

where,

$\hat{p}^i_i \equiv$ point estimation of the probability $P(C = c_i)$

$z_\alpha \equiv (1-\alpha)$ percentil in the $\mathcal{N}(0,1)$

$N \equiv$ is the number of cases in dataset

- **Search space definition**

Once the confidence intervals are estimated from the dataset, it is possible to generate as many naïve Bayes classifiers as we want. We must only take a value inside each estimated confidence interval.

In this way, each naïve Bayes classifier is going to be represented with the next tupla,

$$\begin{aligned}(\hat{p}^*_1, \ldots, \hat{p}^*_{\|C\|}, \hat{p}^{1*}_{1,1}, \ldots, \hat{p}^{\|C\|*}_{1,1}, \ldots,\\ \hat{p}^{\|C\|*}_{1,\|X_1\|}, \ldots, \hat{p}^{\|C\|*}_{n,\|X_n\|})\end{aligned} \tag{10}$$

where,

$\hat{p}^* \equiv$ selected values inside each confidence interval.

Thus, the search space for the heuristic optimization algorithm is composed of all the valid tuplas. A tupla is valid when it represents a valid naïve Bayes classifier. Formally,

$$\sum_i \hat{p}^*_i = 1 \wedge \forall k \forall i \sum_r \hat{p}^{i*}_{k,r} = 1 \tag{11}$$

Finally, each generated individual must be evaluated with a fitness function. In section 6.2 this function is defined.

- **Heuristic search for the best individual**

Once the individuals and the search space are defined, we must run the optimization heuristic algorithm to search for the best individual.

To make the heuristic search in this work we have used EDAs –estimation of distribution algorithms–. EDAs [12] [10] 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, therefore, the number of parameters that EDAs requires is reduced considerably.

6 Experimentation

6.1 Dataset

In order to evaluate the presented approach (Internal Estimation naïve Bayes) for collaborative filtering a dataset of Microsoft Anonymous Web Data from the UCI repository [13] has been used.

This dataset was created by sampling and processing the `www.microsoft.com` logs and records the use of `www.microsoft.com` by 32711 anonymous, randomly-selected users. Attributes of the table represent each of the 294 areas of the `www.microsoft.com` web site and each record represents all the areas that a user has visited in a one week timeframe. Consequently if a user has visited a certain area along the specified period then the corresponding column will take 1 as the value and 0 otherwise. This ends up with a very sparse and not balanced dataset.

The dataset will be used to predict the areas of `www.microsoft.com` that a new user will visit.

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 [8].

6.2 Measuring prediction accuracy -Evaluation Function-

The percentage of successful prediction is the measure most frequently used to measure the quality of a classifier when we deal with balanced datasets. Nevetheless, when the training dataset happens to be not balanced this measure is not appropriate. As an example let's suppose that only 1000 users from a total of 32711 have visited a certain page. A classifier with the following confusion matrix (see table 1), in which no potential visitor is classified as so, would have an accuracy of $31711/32711 = 96.94\%$, which illustrates the fact that accuracy is not an appropriate measure of quality when dealing with sparse datasets.

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

Table 1. An example of a confusion matrix

Consequently, in order to measure the quality of prediction we propose to use a new measure to balance the results. Given a generic confusion matrix (see table 2), the measure we propose to use is:

$$\left(\frac{a}{a+b} + \frac{d}{c+d}\right)/2 \qquad (12)$$

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

Table 2. A generic confusion matrix

This measure happens to be more realistic than accuracy in the example we are dealing with because it evaluates the percentage of visitors classified as visitors and the percentage of non-visitors classified as non-visitors independently and then calculates the average value. This measure will be used when evaluating results of the interval estimation naïve bayes

It is also worth mentioning that in order to calculate the probability of a user to be classified as a certain class, the idea exposed in the variant Sparse Data Model (see equation 3) defined by Pazzani and Miyahara has been used in our approach.

6.3 Experimental results

The presented approach has been used to predict if a user will visit one of the 18 most visited areas of the site. The most visited area has 10836 visitors and the less visited has 1087 visitors. This range of visitors is enough to analyze the behavior of the interval estimation naïve bayes.
Simple naïve Bayes, the variant Sparse Data Model defined by Pazzani and Miyahara and interval estimation naïve Bayes have been executed on the training dataset. The results are shown in Table 3. The first two columns of the table represent the areas and the number of visitors for each area. The remaining columns shows the results of the experiments. For each algorithm the evaluation function (f_eval) is shown. As it was already mentioned, the evaluation function is the value that will be taken into account when quality is measured.

		simple NB	**PazzaniNB**	**IENB Max**
Area	*Visitors*	*f_eval*	*f_eval*	*f_eval*
'1008'	10836	63.03	70.59	71.73
'1034'	9383	64.72	54.56	55.93
'1004'	8463	54.12	53.95	55.75
'1018'	5330	72.08	75.60	77.49
'1017'	5108	62.39	68.96	71.15
'1009'	4628	71.16	72.11	73.12
'1001'	4451	71.75	77.49	78.58
'1026'	3220	71.41	83.68	85.84
'1003'	2968	72.21	78.39	79.54
'1025'	2123	67.24	55.53	57.53
'1035'	1791	75.97	88.64	89.54
'1040'	1506	68.06	75.45	79.36
'1041'	1500	71.74	79.61	80.77
'1032'	1446	57.27	57.23	58.47
'1037'	1160	68.40	79.26	80.99
'1030'	1115	65.26	71.69	73.51
'1038'	1110	73.52	80.58	83.66
'1020'	1087	62.71	68.60	70.97
Average		**67.39**	**71.77**	**73.55**

Table 3. Experimental results for Interval Estimation naïve Bayes

In order to evaluate the Interval Estimation naïve Bayes the following parameters have to be taken into account: (i) Prediction accuracy: Results from the experiments highlight the fact that the new approach presented outperforms both simple naïve and the variant presented by Pazzani. The variant of Pazzani and Miyahara outperforms simple naïve Bayes in 4.38% and our new approach outperforms the last one in 1.78% and the simple naïve Bayes in 6.16%; (ii) Learning time: In this aspect results are not so clear. 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 as the evaluation of the simple naïve Bayes classifier. In conclusion, the learning time could be in the range of hours. However the learning process has to be executed only once so that this should not be a drawback of the algorithm;(iii) Speed of predictions: Once the learning is made, the speed of the predictions has shown to be exactly the same for all the three algorithms.

7 Conclusion and further work

In this work interval estimation naïve bayes has been proposed as a new approach for collaborative filtering. Experimental results shown that our approach outperforms the simple naïve Bayes and other variants specifically defined for collaborative filtering.

Nevertheless this is the first approach and we think it could obtain better results changing the objective of the heuristic search. We are currently working in the maximization of the area under the ROC curve. We are also working in the combination of the presented approach 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. We think that this enhancement would improve the results of the presented approach.

D. Billsus and M.J. Pazzani. Learning collaborative information filters. In *Proceeding of the 15th International Conference on Machine Learning*, pages 46–54, San Francisco, CA, 1998. Morgan Kaufmann.

J.S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Gregory F. Cooper and Serafn Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52, San Francisco, July 24–26 1998. Morgan Kaufmann.

J. Canny. Collaborative filtering with privacy, 2002.

R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.

L. Getoor and M. Sahami. Using probabilistic relational models for collaborative filtering. In Springer, editor, *Proceedings WEBKDD99*, 1999.

D.J. Hand and K. Yu. Idiot's Bayes - not so stupid after all? *International Statistical Review*, 69(3):385–398, 2001.

T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In Dean Thomas, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99-Vol2)*, pages 688–693. Morgan Kaufmann Publishers, July 31-August 6 1999.

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, pages 1137–1145, 1995.

J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. In *Communications of the ACM*, volume 40, pages 77–87. March 1997.

P. Larrañaga, J.A. Lozano, and E. Bengoetxea. Estimation of Distribution Algorithms based on multivariate normal and Gaussian networks. Technical Report KZZA-IK-1-01, Department of Computer Science and Artificial Intelligence, University of the Basque Country, 2001.

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.

H. Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5:303–346, 1998.

P. M. Murphy and D. W. Aha. UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/, 1995.

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.

P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering in netnews. In *Proceedings of CSCW'94*, Chapel Hill NC, 1994.

B. Sarwar, G. Karypis, J. Konstan, and J. Rieddl. Analysis of recommendation algorithms for e-commerce. In *Proceedings of EC'00*, Mineapolis, October 2000.

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 Volume 1 of papers: Using the Information of Others, pages 210–217, 1995.