INSTITUT D'INVESTIGACIÓ EN INTEL.LIGÈNCIA ARTIFICIAL

CSIC

TECHNICAL REPORT TR-2003-01

# The Indifferent Naive Bayes Classifier

Jesús Cerquides
Ramon López de Màntaras

JANUARY 2003

**Abstract**

The Naive Bayes classifier is a simple and accurate classifier. This paper shows that assuming the Naive Bayes classifier model and applying bayesian model averaging and the principle of indifference, an equally simple, more accurate and theoretically well founded classifier can be obtained.

# 1 Introduction

In this paper we apply probability theory as extended logic [5] to the Naive Bayes classifier. By using bayesian model averaging and the principle of indifference, we derive the Indifferent Naive Bayes classifier (IndifferentNB from now on). In section 2 we introduce the Naive Bayes classifier, paying special attention to the estimation of its parameters. In section 3 we do the mathematical development of our classifier. In section 4 we perform the empirical comparison of IndifferentNB with the standard implementation of Naive Bayes and the one proposed in [7] by Kontkanen et al. showing that IndifferentNB reduces the classification error rate and approximates better the probabilities, specially when little data is available. We finish with some conclusions and possibilities for future research in section 5.

# 2 The Naive Bayes classifier

The Naive Bayes classifier [8] is a classification method based on the assumption of conditional independence between the different variables in the dataset given the class. Let $K$ be the number of classes and $A = \{A_1, \ldots, A_N\}$ the set of attributes and represent $Val(A_i)$ as the set of values that $A_i$ can take and $\#A_i$ as the number of elements in $Val(A_i)$. The model has the parameters:

- For each class $c_j$, $1 \leq j \leq K$:
  - $\alpha_j = p(C = c_j)$
  - For each attribute $A_i$, $1 \leq i \leq N$:
    - For each possible value $v$ that $A_i$ can be in:
      $$\theta_{i,v,j} = p(A_i = v \wedge C = c_j)$$

Imagine we need to know the probability of a new observation $S$ being in class $c_l$ given a Naive Bayes model $M$. Applying the Naive Bayes model we have that:

$$p(C = c_l, S|M) = \alpha_l \prod_{i=1}^{N} \frac{\theta_{i,S_i,l}}{\alpha_l} \tag{1}$$

In many cases, the way of explaining and understanding the Naive Bayes classifier has suffered from "the mind projection fallacy", to use the term introduced by Jaynes in [5]. Hence, it has been accepted that what we need to do is to approximate $\alpha_j$ and $\theta_{i,v,j}$ by the frequencies in the data set. Defining $n_j$ as the number of observations for class $c_j$ in the dataset and $n_{i,v,j}$ as the number of observations with class $c_j$ and value $v$ for attribute $A_i$, the Naive Bayes approximates $\alpha_j$, $\theta_{i,v,j}$ as follows:

$$\alpha_j = \frac{n_j}{\sum_{j'=1}^{K} n_{j'}} \tag{2}$$

$$\theta_{i,v,j} = \frac{n_{i,v,j}}{n_j}\ \alpha_j \tag{3}$$

It has been empirically noticed that approximating these probabilities by their frequency in the dataset can lead to a value of zero in expression (1). This can happen if one of the $\theta_{i,S_i,l}$ is zero, that is if the value of one of the attributes $A_i$ of the new observation $S$,

that we are trying to classify, has not been observed, in the dataset, for the class $c_l$. In other words, if the number of observations in the dataset fulfilling $A_i = S_i$ and $C = c_l$ is zero. To avoid this problem, a "softening" consisting in assigning a small probability instead of zero to $\theta_{i,S_i,l}$ can be done. That softening can improve the accuracy of the classifier. A set of *ad hoc* not well founded softening methods have been tried [2, 6].

In [7], Kontkanen et al. propose an approach for Instance-Based Learning (IBL) and apply it to the Naive Bayes classifier. This approach is based on the Bayesian model averaging principle [4]. More concretely, they define $\phi_{i,v,j} = \frac{\theta_{i,v,j}}{\alpha_j}$ and arrive to the conclusion that if we accept a Dirichlet prior for $\alpha_\cdot$ and for each $\phi_{i,\cdot,j}$, that is if $(\alpha_1, \ldots, \alpha_K) \sim \mathrm{Di}(\mu_1, \ldots, \mu_K)$ and $(\phi_{i,1,j}, \ldots, \phi_{i,\#A_i,j}) \sim \mathrm{Di}(\sigma_{i,1,j}, \ldots, \sigma_{i,\#A_i,j})$ where $\mu_i$, $\sigma_{i,v,j}$ are the prior hyperparameters, then the classifier resulting from applying bayesian model averaging can be represented as a Naive Bayes with the following softened approximation of $\alpha_j$, $\theta_{i,v,j}$ [7]:

$$\alpha_j = \frac{n_j + \mu_j}{\sum_{j'=1}^{K} (n_{j'} + \mu_{j'})} \tag{4}$$

$$\theta_{i,v,j} = \frac{n_{i,v,j} + \sigma_{i,v,j}}{n_j + \sum_{v' \in Val(A_i)} \sigma_{i,v',j}} \alpha_j \tag{5}$$

Kontkanen's work sheds some light on why "softening" improves accuracy and shows that accuracy can be further improved if the "softening" has a theoretically well founded basis.

In spite of pointing in the right direction, in our opinion, Kontkanen et al. disregard the fact that the situation allows for the application of the principle of indifference. First enunciated by Bernoulli and afterwards advocated for by Laplace, Jaynes and many others [5], the principle of indifference, also known as the principle of insufficient reason tell us that if we are faced with a set of exhaustive, mutually exclusive alternatives and have no significant information that allow us to differentiate any one of them, we should assign all of them the same probability. In the following section we will present an improvement of the algorithm presented in [7] by using the principle of indifference for objectively setting the prior probability and show that this implies a relationship between the hyperparameters that has not been noticed in [7].

## 3 The Indifferent Naive Bayes Classifier

The Naive Bayes model as described above defines a set of plausible models $\mathcal{M}$. Our objective is calculating $p(C = c_l, S|D, I)$ where $D$ is the set of observations we have to learn from, $S$ the observation we are trying to classify and $I$ represents our state of knowledge. Assuming that we can apply the principle of indifference to the set of models, that is

$$\forall M \in \mathcal{M} \quad p(M|I) = Q$$

where $Q$ is a constant, in Appendix A we show that:

$$p(C = c_l, S|D, I) = \mathcal{K} \left( n_l + 1 + \sum_{i=1}^{N} \#A_i - N \right) \prod_{i=1}^{N} \frac{n_{i,S_i,l} + 1}{n_l + \#A_i} \tag{6}$$

where $\mathcal{K}$ is a normalizing constant, $n_l$ is the number of observations of class $c_l$ in the data, $\#A_i$ is the number of different values that $A_i$ can take and $n_{i,S_i,l}$ is the number of observations in the data fulfilling $A_i = S_i$ and $C = c_l$.

From here it is easy to see that the classifier can be represented by a Naive Bayes classifier that uses the following softened approximations:

$$\alpha_j = \frac{n_j + 1 + \sum_{i=1}^{N} \#A_i - N}{\sum_{j'=1}^{K} (n_{j'} + 1 + \sum_{i=1}^{N} \#A_i - N)} \tag{7}$$

$$\theta_{i,v,j} = \frac{n_{i,v,j} + 1}{n_j + \#A_i} \alpha_j \tag{8}$$

These results can be seen as a particular case of the ones from [7] shown in equations 4,5. In spite of that, it is worth noticing the following two facts:

- In [7], Kontkanen et al. assume a Dirichlet prior distribution with a set of hyperparameters that have to be fixed at some point in time. This means that a methodologic usage of that classifier requires an assessment of the prior hyperparameters for each dataset in which we would like to apply it. Instead, we have used the principle of indifference to obtain a well founded prior without information about the dataset besides the number of attributes and the cardinality of its attributes and class.

- In equations 4 and 5 the hyperparameters $\mu_{\cdot}$ and $\sigma_{i,\cdot,j}$, for $\alpha_{\cdot}$ and $\theta_{i,\cdot,j}$ are not related. Instead, in our approach there is a link between the softening parameters, because the value of $\alpha_i$ in equation 7, depends not only on the number of classes but also on the number of attributes, $N$, and on the number of values of each attributes, $\#A_i$, and the value of $\theta_{i,v,j}$ in equation 8, depends also on the number of values of the attribute, $\#A_i$.

The experimental results in the next section show that this two facts lead to a reduced error rate of the classifier.

# 4 Experimental results

| **Dataset** | Attributes | Instances | Classes | Missing |
|---|---|---|---|---|
| DCREDITS | 5 | 3781 | 15 | few |
| ADULT | 14 | 48842 | 2 | some |
| BREAST | 10 | 699 | 2 | 16 |
| CAR | 6 | 1728 | 4 | no |
| CHESS | 36 | 3196 | 2 | no |
| CLEVE | 13 | 303 | 2 | some |
| CRX | 15 | 690 | 2 | few |
| GLASS | 10 | 214 | 2 | none |
| IRIS | 4 | 150 | 3 | none |
| LETTER | 16 | 20000 | 26 | none |
| MUSHROOM | 22 | 8124 | 2 | some |
| NURSERY | 8 | 12960 | 5 | no |
| OPTDIGITS | 64 | 5620 | 10 | none |
| PIMA | 8 | 768 | 2 | no |
| SOYBEAN | 35 | 316 | 19 | some |
| VOTES | 16 | 435 | 2 | few |

Table 1: Datasets information

| Dataset | BIBL | MAPNB | IndifferentNB |
|---|---|---|---|
| DCredits | 25.15 ± 2.88 | **21.06 ± 2.67** | 23.55 ± 2.64 |
| breast | 4.42 ± 1.59 | 4.60 ± 1.83 | **4.20 ± 1.53** |
| mushroom | 6.89 ± 0.58 | **0.78 ± 0.30** | 6.89 ± 0.57 |
| votes | 11.20 ± 2.97 | 13.21 ± 4.79 | **11.10 ± 3.18** |
| car | 20.95 ± 2.73 | **20.18 ± 2.70** | 20.82 ± 2.59 |
| crx | 16.86 ± 3.13 | 22.37 ± 4.61 | **16.82 ± 3.12** |
| glass | **41.76 ± 9.74** | 73.25 ± 19.49 | 42.45 ± 9.19 |
| iris | 19.37 ± 7.93 | 19.14 ± 7.59 | **18.18 ± 7.05** |
| nursery | 10.56 ± 0.88 | 10.42 ± 0.95 | **10.33 ± 0.89** |
| pima | 29.40 ± 3.74 | **29.36 ± 3.90** | 29.83 ± 3.64 |
| adult | **18.51 ± 0.73** | 18.64 ± 0.67 | 18.89 ± 0.72 |
| chess | 15.90 ± 3.06 | **15.55 ± 2.96** | 15.87 ± 3.03 |
| letter | 39.39 ± 0.57 | 39.70 ± 0.74 | **39.09 ± 0.40** |
| optdigits | **12.33 ± 1.19** | 22.78 ± 1.10 | 12.35 ± 1.13 |
| soybean | 40.22 ± 6.41 | 69.28 ± 10.42 | **38.54 ± 6.34** |
| cleve | 21.78 ± 5.07 | 29.98 ± 7.21 | **21.62 ± 5.17** |

Table 2: Averages and standard deviations of error rates using 10% training data

| Dataset | BIBL | MAPNB | IndifferentNB |
|---|---|---|---|
| DCredits | 302.66 ± 33.07 | 664.45 ± 139.61 | **280.96 ± 29.03** |
| breast | 23.76 ± 11.42 | 197.08 ± 102.64 | **23.70 ± 11.66** |
| mushroom | 442.88 ± 72.67 | **107.58 ± 131.06** | 441.43 ± 71.85 |
| votes | **50.02 ± 21.80** | 230.76 ± 151.23 | 51.42 ± 22.37 |
| car | **169.64 ± 17.79** | 387.42 ± 205.72 | 186.01 ± 17.04 |
| crx | 68.40 ± 17.46 | 345.75 ± 135.47 | **67.78 ± 17.19** |
| glass | 55.88 ± 15.08 | 1294.34 ± 608.31 | **53.08 ± 13.37** |
| iris | 14.25 ± 5.75 | 91.66 ± 64.77 | **13.32 ± 4.99** |
| nursery | 365.17 ± 10.85 | **350.04 ± 9.02** | 383.97 ± 11.59 |
| pima | **99.19 ± 14.89** | 271.02 ± 163.27 | 101.13 ± 15.21 |
| adult | **3084.83 ± 123.53** | 3548.78 ± 221.05 | 3138.38 ± 127.60 |
| chess | 219.02 ± 31.08 | 241.76 ± 50.48 | **218.52 ± 30.86** |
| letter | 16780.41 ± 460.27 | 79666.20 ± 2489.86 | **16616.48 ± 468.10** |
| optdigits | 835.15 ± 109.63 | 9487.80 ± 1253.42 | **828.98 ± 102.78** |
| soybean | 438.03 ± 132.96 | 4524.90 ± 1335.06 | **393.61 ± 120.65** |
| cleve | 34.66 ± 10.25 | 321.26 ± 167.61 | **34.18 ± 10.01** |

Table 3: Averages and standard deviations of *LogScore* using 10% training data

We tested three algorithms over 15 datasets from the Irvine repository [1] plus our own credit screening database. The dataset characteristics are described in Table 1. To discretize continuous attributes we used equal frequency discretization with 5 intervals. For each dataset and algorithm we tested both accuracy and *LogScore* as defined in [7]. We used cross validation, and we made two experiments: taking all of the learning fold and taking only 10 % of it. This is done because the three methods converge to the same model given enough data. Hence, comparing them when the size of the training data is small can provide us with good insight of how they differentiate.

The error rates appear in Tables 2,4, with the best method for each dataset boldfaced. *LogScore*'s appear in Tables 3,5. The columns of the tables are the induction methods and the rows are the datasets. The meaning of the column headers are:

- MAPNB is the standard Naive Bayes algorithm using frequencies as probability estimates, as shown in equations 2 and 3.
- BIBL is the algorithm appearing in [7] and shown in equations 4 and 5 and fixing the hyperparameters to get uniform prior probability distributions.
- IndifferentNB is the Indifferent Naive Bayes as described in equations 7 and 8.

### 4.1 Interpretation of the results

In order to make sense of all the numbers in tables 2, 4, 3 and 5 we have selected some comparisons and have put them into separate graphs.

| Dataset | BIBL | MAPNB | IndifferentNB |
|---|---|---|---|
| DCredits | 17.90 ± 1.44 | **16.77 ± 1.57** | 17.69 ± 1.53 |
| breast | 3.34 ± 1.19 | 3.55 ± 1.30 | **3.31 ± 1.19** |
| mushroom | 4.70 ± 0.44 | **0.38 ± 0.14** | 4.70 ± 0.44 |
| votes | 10.84 ± 2.56 | **10.56 ± 2.67** | 10.81 ± 2.61 |
| car | 14.93 ± 2.02 | 14.57 ± 2.00 | **14.05 ± 1.96** |
| crx | 14.77 ± 2.54 | 15.15 ± 2.66 | **14.76 ± 2.55** |
| glass | 20.79 ± 6.30 | **17.75 ± 5.76** | 20.39 ± 6.28 |
| iris | 13.10 ± 5.14 | **12.61 ± 4.78** | 13.03 ± 4.97 |
| nursery | 9.78 ± 0.83 | 9.78 ± 0.83 | **9.76 ± 0.84** |
| pima | 25.92 ± 3.19 | **25.91 ± 3.20** | 25.95 ± 3.17 |
| adult | **18.49 ± 0.50** | 18.50 ± 0.51 | 18.53 ± 0.50 |
| chess | 12.45 ± 1.38 | **12.41 ± 1.37** | 12.44 ± 1.37 |
| letter | 27.39 ± 0.43 | **25.64 ± 0.67** | 27.35 ± 0.48 |
| optdigits | **8.13 ± 0.35** | 9.09 ± 0.31 | **8.13 ± 0.35** |
| soybean | 11.11 ± 2.29 | **6.17 ± 1.70** | 10.81 ± 2.24 |
| cleve | **17.99 ± 3.76** | 19.22 ± 3.83 | 18.01 ± 3.76 |

Table 4: Averages and standard deviations of error rates using 100% training data

| Dataset | BIBL | MAPNB | IndifferentNB |
|---|---|---|---|
| DCredits | 198.06 ± 21.60 | 204.38 ± 28.13 | **196.15 ± 21.34** |
| breast | **36.14 ± 19.70** | 89.37 ± 58.79 | 36.49 ± 19.89 |
| mushroom | 229.40 ± 34.17 | **13.94 ± 8.25** | 229.27 ± 34.15 |
| votes | **55.28 ± 23.82** | 57.91 ± 25.51 | 55.51 ± 23.92 |
| car | 116.28 ± 9.64 | **110.59 ± 9.33** | 114.33 ± 9.28 |
| crx | 59.23 ± 13.36 | 73.29 ± 28.61 | **59.12 ± 13.33** |
| glass | **20.79 ± 7.45** | 76.38 ± 49.70 | 21.68 ± 7.72 |
| iris | 7.32 ± 3.83 | 15.90 ± 15.28 | **7.30 ± 3.81** |
| nursery | 340.00 ± 13.27 | **339.94 ± 11.71** | 340.86 ± 13.74 |
| pima | **83.03 ± 11.36** | 83.82 ± 11.68 | 83.38 ± 11.40 |
| adult | **3066.40 ± 78.40** | 3087.96 ± 79.24 | 3072.35 ± 78.47 |
| chess | 187.29 ± 13.77 | **187.18 ± 14.22** | 187.26 ± 13.77 |
| letter | 11960.44 ± 378.45 | 14221.51 ± 589.09 | **11958.86 ± 373.91** |
| optdigits | 676.25 ± 59.73 | 1530.88 ± 139.14 | **676.07 ± 59.62** |
| soybean | 107.36 ± 34.04 | **51.23 ± 25.67** | 110.71 ± 35.39 |
| cleve | 29.24 ± 8.32 | 40.76 ± 14.52 | **29.22 ± 8.30** |

Table 5: Averages and standard deviations of *LogScore* using 100% training data

#### 4.1.1 IndifferentNB against BIBL

In this section we compare the accuracy and *LogScore* of IndifferentNB and BIBL. In order to do this we have drawn two graphs, comparing both scores using 10% of the training data.

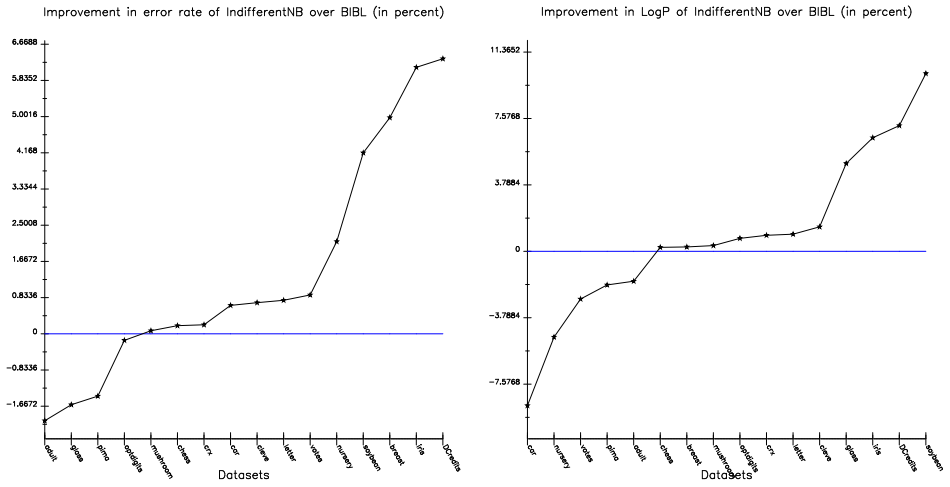
- In Figure 1(a) we can see that accuracy improves for 12 out of the 16 datasets up to a 6% improvement.
- In Figure 1(b) we can see that *LogScore* improves for 11 out of the 16 datasets up to a 11% improvement.

With 100% of the data the difference between both classifiers is in the same direction while not so significant.

#### 4.1.2 IndifferentNB against MAPNB

In this section we compare the accuracy and *LogScore* of IndifferentNB and MAPNB. In order to do this we have also drawn two graphs, comparing both scores using 10% of the training data.

- In Figure 2(a) we can see that accuracy improves for 10 out of the 16 datasets up to a 40% improvement.
- In Figure 2(b) we can see that *LogScore* improves for 14 out of the 16 datasets up to almost a 100% improvement. This is due to the fact that in some cases MAPNB gives probability 0 to the real class. This raises the *LogScore* to infinity.




(a) Error rate

(b) Log score

Figure 1: Comparison of IndifferentNB and BIBL using 10% training data

# 5 Conclusions and future work

We have developed the Indifferent Naive Bayes classifier by applying bayesian model averaging and the principle of indifference. While the objective of the development was to find a dataset independent well founded prior, we have seen that the development also leads to improvements in the error rate, specially when only small amounts of data are available. In future work we would provide the IndifferentNB with the possibility of handling unknown values. We would also like to extend the development to Tree Augmented Naive Bayes [3].

# A Mathematical development

Let $\#A_i$ be the number of different values that $A_i$ can be in, $\mathcal{M}$ the set of possible models, containing the different settings in which the different parameters can be. $D$ the set of observations we have to learn from. In a more detailed level $D$ contains:

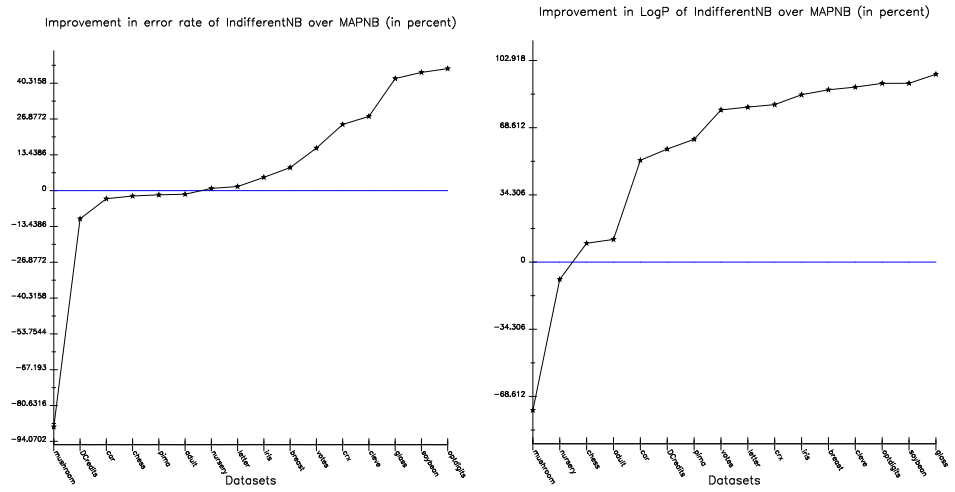
- $n_1, \ldots, n_K$ number of observations of each class in the data.
- $n_{i,v,j}$ number of observations fulfilling $A_i = v$ and $C = c_j$.

$I$ represents our state of knowledge. The observation we are trying to classify is $S$ where $S_i$ is the value of attribute $A_i$ for $S$. Applying Bayesian model averaging [4] we have that

$$p(C = c_l, S|D, I) = \int_{M \in \mathcal{M}} p(C = c_l, S|M, I) p(M|D, I) dM \tag{9}$$

applying Bayes theorem over $p(M|D, I)$:

$$p(M|D, I) = p(D|M, I) \frac{p(M|I)}{p(D|I)} \tag{10}$$




(a) Error rate

(b) Log score

Figure 2: Comparison of IndifferentNB and MAPNB using 10% training data

where the denominator can be expanded as:

$$p(D|I) = \int_{M \in \mathcal{M}} p(D|M, I) p(M|I) dM \tag{11}$$

Here we can apply the principle of indifference, by assigning a uniform constant prior over the set of models, independent of the model parameters, that is:

$$\forall M \in \mathcal{M} \;\; p(M|I) = Q \tag{12}$$

In order to understand what lies behind this assumption, a good explanation of its meaning can be found in chapter 18 of [5]. We can take $Q$ out of the integral sign:

$$p(D|I) = Q \int_{M \in \mathcal{M}} p(D|M, I) dM \tag{13}$$

Substituting 12,13 in 10 we have:

$$p(M|D, I) = \frac{p(D|M, I)}{\int_{M \in \mathcal{M}} p(D|M, I) dM} \tag{14}$$

where the denominator is a constant independent of the model. Using equation 1 for every observation in $D$ we can expand $p(D|M, I)$ as

$$p(D|M, I) = \prod_{j=1}^{K} \alpha_j^{n_j} \prod_{i=1}^{N} \prod_{v \in Val(A_i)} \left( \frac{\theta_{i,v,j}}{\alpha_j} \right)^{n_{i,v,j}} \tag{15}$$

Defining

$$c = \frac{1}{\int_{M \in \mathcal{M}} p(D|M, I) dM} \tag{16}$$

and substituting this expression and 15 in 14 we get

$$p(M|D,I) = c\prod_{j=1}^{K}\alpha_j^{n_j}\prod_{i=1}^{N}\prod_{v\in Val(A_i)}\left(\frac{\theta_{i,v,j}}{\alpha_j}\right)^{n_{i,v,j}} \tag{17}$$

Going back to 9, we have have found a covenient expression for $p(M|D,I)$ and for $p(C=c_l,S|M,I)$ (in equation 1). We introduce the following notation for compactness:

$$\beta_{j,l} = \begin{cases} 1 & j = l \\ 0 & j \neq l \end{cases} \tag{18}$$

$$\beta_{i,v,S_i,j,l} = \begin{cases} 1 & j = l \wedge v = S_i \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

then we have

$$\begin{aligned} p(C=c_l,S|D,I) &= \\ &= c\int_{M\in\mathcal{M}}\left(\prod_{j=1}^{K}\alpha_j^{n_j+\beta_{j,l}}\prod_{i=1}^{N}\prod_{v\in Val(A_i)}\left(\frac{\theta_{i,v,j}}{\alpha_j}\right)^{n_{i,v,j}+\beta_{i,v,S_i,j,l}}\right)dM = \\ &= c\int_{M\in\mathcal{M}}\left(\prod_{j=1}^{K}\alpha_j^{(1-N)(n_j+\beta_{j,l})}\prod_{i=1}^{N}\underbrace{\prod_{v\in Val(A_i)}\theta_{i,v,j}^{\,n_{i,v,j}+\beta_{i,v,S_i,j,l}}}_{\mathcal{A}}\right)dM \end{aligned} \tag{20}$$

Since our model imposes that $\sum_{j=1}^{K}\alpha_j = 1$ and $\forall i\ \forall j\quad \alpha_j = \sum_{v\in Val(A_i)}\theta_{i,v,j}$ we can decompose the previous integral in a set of integrals where our model parameters range from 0 to $\infty$ encoding our model restrictions as $\delta$ functions. Hence, defining $\mathcal{B}(i,j) = \int_{\theta_{i,.,j}}\cdots\int\mathcal{A}\ \delta\big(\sum_{v\in Val(A_i)}\theta_{i,v,j}-\alpha_j\big)d\theta_{i,.,j}$ we have that

$$\begin{aligned} p(C=c_l,S|D,I) &= \\ &c\int_{\alpha_.}\cdots\int\left(\prod_{j=1}^{K}\alpha_j^{(1-N)(n_j+\beta_{j,l})}\prod_{i=1}^{N}\mathcal{B}(i,j)\ \delta\Big(\sum_{j=1}^{K}\alpha_j-1\Big)\right)d\alpha_. \end{aligned} \tag{21}$$

The definite integral appearing in $\mathcal{B}(i,j)$ can be solved by means of Laplace transforms as can be seen in [5]. From the developments in [5] it is easy to infer the following general result:

$$\int_{\gamma_.}\cdots\int\prod_{i=1}^{V}\gamma_i^{m_i}\delta(\sum_{i=1}^{V}\gamma_i - r)d\gamma_. = \frac{\prod_{i=1}^{V}m_i!}{\Big(\sum_{i=1}^{V}m_i+V-1\Big)!}r^{\left(\sum_{i=1}^{V}m_i+V-1\right)} \tag{22}$$

applying this result to $\mathcal{B}(i,j)$ we have

$$\mathcal{B}(i,j) = \frac{\prod_{v\in Val(A_i)}(n_{i,v,j}+\beta_{i,v,S_i,j,l})!}{(n_j+\beta_{j,l}+\#A_i-1)!}\alpha_j^{n_j+\beta_{j,l}+\#A_i-1} \tag{23}$$

Then, substituting in 21 and grouping:

$$p(C = c_l, S|D, I) = c \prod_{j=1}^{K} \prod_{i=1}^{N} \frac{\prod_{v \in Val(A_i)} (n_{i,v,j} + \beta_{i,v,S_i,j,l})!}{(n_j + \beta_{j,l} + \#A_i - 1)!} \times \times \int_{\alpha_\cdot} \cdots \int \prod_{j=1}^{K} {\alpha_j}^{n_j + \beta_{j,l} + \#A_i - 1} \delta\Big(\sum_{j=1}^{K} \alpha_j - 1\Big) d\alpha_\cdot \quad (24)$$

Applying 22 again we get

$$p(C = c_l, S|D, I) = c' \prod_{j=1}^{K} \left( n_j + \beta_{j,l} + \sum_{i=1}^{N} \#A_i - N \right)! \prod_{i=1}^{N} \frac{\prod_{v \in Val(A_i)} (n_{i,v,j} + \beta_{i,v,S_i,j,l})!}{(n_j + \beta_{j,l} + \#A_i - 1)!} \quad (25)$$

where $c' = \dfrac{c}{\left( \sum_{j=1}^{K} n_j + 1 + K(\sum_{i=1}^{N} \#A_i - N) \right)!}$

We can calculate the *Bayes factor* dividing by the probability of a reference class $l'$:

$$\frac{p(C = c_l, S|D, I)}{p(C = c_{l'}, S|D, I)} = \frac{n_l + 1 + \sum_{i=1}^{N} \#A_i - N}{n_{l'} + 1 + \sum_{i=1}^{N} \#A_i - N} \prod_{i=1}^{N} \frac{n_{i,S_i,l} + 1}{n_l + \#A_i} \left( \frac{n_{i,S_i,l'} + 1}{n_{l'} + \#A_i} \right)^{-1} \quad (26)$$

And from here the result appearing in equation 6 follows trivially.

## References


[1] C. Blake, E. Keogh, and C. J. Merz. UCI repository of machine learning databases, 1998.

[2] Bojan Cestnik. Estimating probabilities: A crucial task in machine learning. In *Proceedings of the 9th European Conference on Artificial Intelligence*, pages 147–149, 1990.

[3] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[4] Jennifer A. Hoeting, David Madigan, Adrian E. Raftery, and Chris T. Volinsky. Bayesian model averaging. Technical Report 9814, Department of Statistics. Colorado State University, 1998.

[5] Edwin T. Jaynes. *Probability Theory: The Logic of Science.* published on the net, file://bayes.wustl.edu/Jaynes.book, March 1996 fragmentary edition.

[6] Ron Kohavi, Barry Becker, and Dan Sommerfield. Improving simple bayes. In *Proceding of the European Conference in Machine Learning*, 1997.

[7] Petri Kontkanen, Petri Myllymaki, Tomi Silander, and Henry Tirri. Bayes Optimal Instance-Based Learning. In C. Nédellec and C. Rouveirol, editors, *Machine Learning: ECML-98, Proceedings of the 10th European Conference*, volume 1398 of *Lecture Notes in Artificial Intelligence*, pages 77–88. Springer-Verlag, 1998.

[8] Pat Langley, Wayne Iba, and Kevin Thompson. An Analysis of Bayesian Classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 223–228. AAAI Press and MIT Press, 1992.