Probabilistic approach over Decision Trees for problems with discrete data and a small number of instances 2017 — Norwich, UK

Luke M. Garrigan

Machine Learning, University of East Anglia, UK
l.garrigan@uea.ac.uk

Abstract

Bayesian classifiers are widely known for their optimality when attributes are independent given the class. This paper attempts to prove that small samples of discrete data with arbitrary dependencies are more accurately classified using a probabilistic approach over decision trees.

Index Terms: Probabilistic, Naive Bayes, Decision Trees

1 Introduction

In machine learning a learner algorithm is given a set of training instances with their corresponding class labels, it then produces a classifier. The classifier takes unlabelled testing instances and assigns it to a class. Choosing the best suited algorithm specific to the sample set is not a trivial process.

Probabilistic classification is the application of approximating a joint distribution with a product distribution. Bayes rule is used to approximate the conditional probability of a given class label. Approaches such as *naive* Bayes are among the most popular classifiers used in the machine learning community. Derived from generative probability models they are generally easy to understand and the induction of these classifiers is extremely fast, requiring only a single pass through the data if all attributes are discrete [1]. The *naive* Bayes classifier is the simplest of models in this paper, it assumes that all attributes are independent of each other given the context of the class. Although the *naive* assumption of independence is not true in terms of most sample sets, many papers such as [2] have proven that *naive* Bayes classification accuracy is very competitive when compared with more complex state-of-the-art algorithms.

Decision trees classify instances by sorting them down the tree from the root to some leaf node which represents the classification of the given instance. Nodes specify a test of some attribute of the instance and each branch from that node corresponds to one of the possible values for this attribute. A given instance is classified moving down the tree, the attribute specific to that node is tested. Following down the branch corresponding to the value of the attribute in the given example, this is then repeated until a leaf node is reached and a classification is made. Decision trees are convenient due to their transparency, they explicitly display all possible alternatives and pursues these alternatives to a conclusion. This allows for a comprehensive analysis of the consequences of each decision.

2 Data Description

2.1 Case Study

2.1.1 Dataset Information

Electroencephalography is a domain concerning recording and interpretation of the electroencephalogram (EEG).EEG is a record of the electric signal generated by the cooperative action of brain cells, or more precisely, the time course of extracellular field potentials generated by their synchronous action[3]. EEG detects electrical activity in the brain using small, flat metal discs (electrodes) attached to the scalp. An EEG is used for diagnosing brain disorders, most frequently epilepsy.

For the current study, EEG data was collected 5 times on various days from a healthy right-handed subject of 25 years of age. The data was recorded on a Medelec Profile Digital EEG machine. The settings of high-frequency filter 50 Hz, low frequency filter 1.6 Hz, notch filter 50 Hz, sensitivity 70 micro volts/mm, and a sampling rate of 256 Hz were used for the basic signal processing.

In summary, the subject was asked to lie down in a relaxed position with eyes closed. The EEG recorded for the relaxed state for 5 minutes.Following this, an audible beep of 60 dB for 0.91 seconds was given before and after the subject was asked to mentally plan lifting of the right-hand thumb (no actual movement), after a gap of 5 minutes the same cue is given to repeat the experiment lasting approximately 30 minutes [4].

2.1.2 Attribute Information

By applying wavelet packet analysis on the original signal 12 wavelet coefficients in the 7-13 Hz frequency band were obtained. This is a classification problem to determine whether the subject is relaxing or planning. There are 182 instances; the univariate dataset contains 12 real attributes and a binary class label.

2.2 Discretization

Discretization concerns with the process of transferring continuous data into discrete counterparts. Numeric attributes were discretized into ten equal-length intervals unless the number of uniquely observed values for an attribute was less than 10. This approach was compared in [5] with entropy-based and purity-based methods, which are supervised algorithms. An empirical evaluation showed that the *naive* Bayes algorithm significantly improved accuracy when features were discretized using an entropy-based method. However, due to its simplicity, the unsupervised binning discretization method was used.

2.3 Methods For Accuracy Estimation

2.3.1 K-Fold Cross-Validation

Cross-validation is a computationally expensive algorithm used to estimate performance, it uses all available instances as both training and testing sets. The dataset is split into k equally sized non-overlapping subsets S. Given a fold S_i a model is trained on $S \setminus S_i$, then S_i is used to create the accuracy estimation.

2.3.2 Leave-One-Out Cross-Validation (LOOCV)

Leave-One-Out Cross-Validation is K-fold cross validation where K is equal to the number of instances in the dataset. The classifier is trained on all data except the one instance being left, this is repeated until all instances in the dataset have been the test instance. An average of the data is collected and used to evaluate the classifier.

2.3.3 Bias And Variance Tradeoff

As k increases the less bias the classification is in overestimating the true expected error as the folds will be closer to the total dataset. However, in doing this it induces variance. To minimise the testing bias a large portion of the dataset must be used for training, meaning not much data is used for testing, this ensures that the model will be as close as possible to the model achievable by training using the entire dataset. Minimising the testing variance would mean quite the opposite, a large amount of data would be used for testing, this ensures a more reliable estimate error of the classifier.

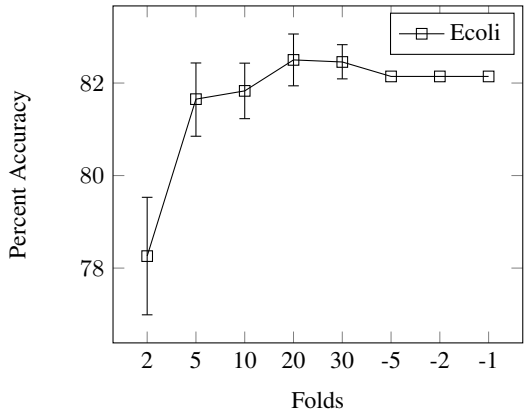

Figure 1: Bias representation of cross-validation. The negative k-folds shows leave-k-out, -1 is LOOCV.

Due to the small sample sizes 10-fold cross-validation was used as an estimator in an attempt to minimise the estimation variance. Figure 1 shows the bias and variance of k-fold cross-validation on arbitrary datasets using *naive* bayes.

3 Classifier Description

3.1 Probabalistic

3.1.1 Naive Bayes

Naive Bayes was selected for testing due to its popularity for classification problems and its presence in the machine learning community. As its name suggests its assumptions are naive and are not generally concordant with the data; it assumes all attributes are independent of each other given the context of the class but has been shown to perform surprisingly well in many classification problems. It is computationally efficient as training is linear in both the number of instances and attributes [6].

Table 1: Details of the data used.

Dataset	Instances	Attributes
Echo cardiogram	131	13
Teaching Assistant	150	6
Seeds	209	8
Planning Relax	180	13
Hepatitis	154	20
Breast Cancer	284	10
Ecoli	223	10
Glass	141	6
Haberman	203	4
Hayes Roth	158	5
Heart	169	14
Lymphography	156	19
Promoters	104	58
Shuttle Landing	252	7
Sonar	137	61
Thyroid	142	6

Let C represent the classification variable, and let C_k be the value of C. According to Bayes Rule, the probability of an instance X with attributes $X = (x_1, x_2, .., x_n)$ having class label C_k is

$$P(C_k \mid X) = \frac{P(X \mid C_k)\, P(C_k)}{P(X)}$$

where $P(C_k \mid X)$ represents the posterior probability of class c given a predictor X. $P(X \mid C_k)$ is the likelihood; the probability of the predictor given C_k. $P(C_k)$ is the prior probability of C_k; the current knowledge of the class distribution and $P(X)$ is the predictor prior probability.

naive Bayes assumes that all attributes are independent given the value of the class label.

$$P(X|C_k) = P(x_1, x_2, .., x_n \mid C_k) = \prod_{i=1}^{n} P(x_i|C_k)$$

So the conditional distribution over the class variable C can be represented by:

$$P(C_k \mid x_1, ..x_n) = \frac{1}{Z} P(C_k) \prod_{i=1}^{n} P(x_i \mid C_k)$$

where Z is a scaling factor dependent on $x_1, x_2, .., x_n$. In order to build a classifier from this, a decision rule is used by picking the classification which is most probably; for two posterior probabilities $P(C_1|X) = 0.54$ and $P(C_2|X) = 0.46$ using the *maximum a posteriori* (MAP) rule, the larger of the two is chosen [7].

3.1.2 Bayesian Networks

Bayesian networks belong to the family of probabilistic graphical models. Graphical model expresses the conditional dependence structure between random variables and are used to represent

knowledge about uncertain domains. Nodes within a Bayesian network represent the random variables, edges represent the conditional dependencies; nodes not connected are conditionally independent of each other. Each node is associated with a probability function that takes as input a particular set of values for the node's parent variables and outputs the probability of the variable represented by the node [8].

In Bayesian networks each node is conditionally independent of any subset of nodes that are not descendants of itself gives its parent, so the value of a node is conditional only on the values of its parent nodes. Let V represent a node in the graph and $par(V_i)$ be the parent of the node:

$$P(V_1, V_2, ...V_n) = \prod_{i=1}^{n} P(V_i \mid par(V_i))$$

So to compute the joint probabilities we must calculate the conditional probability between itself and its parent for each node in the graph. The chain rule is then computer to determine the graphs joint probability functions.

$$P(V_1, V_2, ...V_n) = P(V_1)P(V_2 \mid V_1)P(V_3 \mid V_1, V_2)$$

Prior probabilities are nodes without parents; they are not conditioned on other random variables [9].

Bayesian networks do not necessarily imply a commitment to Bayesian statistics, it is common to use frequentists methods to estimate the parameters of the CPDS.

3.2 Decision Trees

3.2.1 C4.5

C4.5 is an algorithm used to build decision trees, they are created using a divide-and-conquer approach. The tree is formed using information entropy as found in the ID3 algorithm which is a precursor to C4.5, however, this paper only covers the C4.5 algorithm. C4.5 offers a number of improvements over ID3: handling data with missing values, accepts both continuous and discrete attributes, handling attributes with different costs and solves over-fitting by pruning.Ross Quinlan's latest iteration is the C5.0 algorithm which he states is several orders of magnitude faster than C4.5 [10], unfortunately, this software is proprietary thus not compared.

Entropy is a measure of unpredictability, Shannon entropy calculates the level of uncertainty:

$$Entropy(P) = -\sum_{i=1}^{n} P_i(\log_2 P_i)$$

Information gain measures the expected reduction in entropy caused by partitioning according to the given attribute.

$$Gain(S, A) = Entropy(S) - \sum_{vinValues} \frac{|S_v|}{|S|} Entropy(S)$$

C4.5 is implemented recursively, let T be the set of training instances, the algorithm chooses an attribute that best differentiates the instances contained in T, a tree node is created with the value of the chosen attribute. Child nodes are created each link represents a unique value for the given attribute, the child values are then used to further subdivide the instances into subclasses. The subclasses either satisfy the predefined criteria or the remaining attributes choice for the path is null, this is repeated recursively. The classifications for the testing instances is made by following the decision path.

3.2.2 Random Forests

Meta-algorithms are approaches to combine several machine learning techniques into one predictive model in order to decrease the variance (bagging) or bias (boosting) [11].

Ensemble learning methods generate many classifiers and aggregate their results. Random forest incorporates a supplementary layer of randomness to bagging, in addition to constructing each tree using a different bootstrap sample of the data. Random forests construct the classification distinctly using the best among a subset of predictors randomly chosen at that node [12]. Although the approach is somewhat counter-intuitive it has been shown to outperform many state-of-the-art classifiers.

3.2.3 Logistic Model Trees

Logistic Model Trees (LMT) use a combination of a tree structure and logistic regression models to for a single tree. [13] performed experiments showing that LMT produces more accurate classifiers than C4.5, CART, logistic regression, model trees, functional trees, *naive* Bayes trees and Lots.

LMT is a combination of learners which rely on logistic regression models. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables [14]. LMT uses cost-complexity pruning. The compute time of LMT is much greater than the other algorithms.

4 Results

4.1 Supplementary Material

5 Conclusions

5.1 References

References

[1] R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid." in *KDD*, vol. 96. Citeseer, 1996, pp. 202–207.

[2] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine learning*, vol. 29, no. 2, pp. 103–130, 1997.

[3] D. B. Lindsley, "Emotions and the electroencephalogram." 1950.

[4] "Uci machine learning repository: Planning relax data set." [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Planning Relax

[5] J. Dougherty, R. Kohavi, M. Sahami *et al.*, "Supervised and unsupervised discretization of continuous features," in *Machine learning: proceedings of the twelfth international conference*, vol. 12, 1995, pp. 194–202.

[6] E. Frank, M. Hall, and B. Pfahringer, "Locally weighted naive bayes," in *Proceedings of the Nineteenth conference*

Table 2: An example table.

Dataset	*Naive* Bayes	Bayes Net	C4.5	Random Forests	LMT
Echo cardiogram	91.42 ± 0.5	13	0	0	0
Teaching Assistant	150	6	0	0	0
Seeds	209	8	0	0	0
Planning Relax	180	13	0	0	0
Hepatitis	154	20	0	0	0
Breast Cancer	284	10	0	0	0
Ecoli	223	10	0	0	0
Glass	141	6	0	0	0
Haberman	203	4	0	0	0
Hayes Roth	158	5	0	0	0
Heart	169	14	0	0	0
Lymphography	156	19	0	0	0
Promoters	104	58	0	0	0
Shuttle Landing	252	7	0	0	0
Sonar	137	61	0	0	0
Thyroid	142	6	0	0	0

on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc., 2002, pp. 249–256.

[7] H. Zhang, "The optimality of naive bayes," *AA*, vol. 1, no. 2, p. 3, 2004.

[8] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine learning*, vol. 20, no. 3, pp. 197–243, 1995.

[9] F. V. Jensen, *An introduction to Bayesian networks*. UCL press London, 1996, vol. 210.

[10] R. Quinlan, "Data mining tools see5 and c5. 0," 2004.

[11] A. Galkin, "Bagging, boosting and stacking in machine learning." [Online]. Available: https://stats.stackexchange.com/questions/18891/bagging-boosting-and-stacking-in-machine-learning

[12] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[13] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Machine Learning*, vol. 59, no. 1-2, pp. 161–205, 2005.

[14] "What is logistic regression?" [Online]. Available: http://www.statisticssolutions.com/what-is-logistic-regression/