Combining Naive Bayes and Decision Tables

Mark Hall[1], and **Eibe Frank**[2]
[1]Pentaho Corporation, 5950 Hazeltine National Drive, Suite 340, Orlando, FL, USA
[2]Department of Computer Science, University of Waikato, New Zealand

Abstract

We investigate a simple semi-naive Bayesian ranking method that combines naive Bayes with induction of decision tables. Naive Bayes and decision tables can both be trained efficiently, and the same holds true for the combined semi-naive model. We show that the resulting ranker, compared to either component technique, frequently significantly increases AUC. For some datasets it significantly improves on both techniques. This is also the case when attribute selection is performed in naive Bayes and its semi-naive variant.

Introduction

Our combined model is a simple Bayesian network in which the decision table (DT) represents a conditional probability table. It can be viewed as a restricted version of Pazzani's semi-naive Bayesian model (Pazzani 1996). The latter greedily joins attributes into multiple groups of dependent attributes—rather than just one group as the method considered here (represented by the DT). This can result in more powerful models, but also increases computational complexity by an order of magnitude. Another difference is that search and evaluation in this paper are based on AUC instead of accuracy.

Learning the combined model

A DT stores the input data in condensed form based on a selected set of attributes and uses it as a lookup table when making predictions. Each entry in the table is associated with class probability estimates based on observed frequencies. The key to learning a DT is to select a subset of highly discriminative attributes. The standard approach is to choose a set by maximizing cross-validated performance. Cross-validation is efficient for DTs as the structure does not change when instances are added or deleted, only the class counts associated with the entries change. Similarly, cross-validation for naive Bayes (NB) is also efficient as frequency counts for discrete attributes can be updated in constant time. In our experiments we used forward selection to select attributes in stand-alone DTs because it performed significantly better than backward selection. Numeric attributes in the training data (including those to be modeled by NB) were discretized using MDL-based discretization (Fayyad & Irani 1993), with intervals learned from the training data.

The algorithm for learning the combined model (DTNB) proceeds in much the same way as the one for stand-alone DTs. At each point in the search it evaluates the merit associated with splitting the attributes into two disjoint subsets: one for the DT, the other for NB. We use a forward selection, where, at each step, selected attributes are modeled by NB and the remainder by the DT, and all attributes are modeled by the DT initally. Leave-one-out cross-validated AUC is used to evaluate the quality of a split based on the probability estimates generated by the combined model. Note that AUC can easily be replaced by other performance measures. We chose AUC to enable a fair comparison to NB (and hence only used two-class datasets in our experiments). AUC was also used to select attributes for the stand-alone DT.

The class probability estimates of the DT and NB must be combined to generate overall class probability estimates. Assuming $X^\top$ is the set of attributes in the DT and $X^\perp$ the one in NB, the overall class probability is computed as

$$Q(y|X) = \alpha \times Q_{DT}(y|X^\top) \times Q_{NB}(y|X^\perp)/Q(y),$$

where $Q_{DT}(y|X^\top)$ and $Q_{NB}(y|X^\perp)$ are the class probability estimates obtained from the DT and NB respectively, α is a normalization constant, and $Q(y)$ is the prior probability of the class. All probabilities are estimated using Laplace-corrected observed counts.

In addition to the method described above, we also consider a variant that includes attribute selection, which can discard attributes entirely from the combined model. To this end, in each step of the forward selection, an attribute can be discarded rather than added to the NB model. In the experiments we compare this technique to NB with the same wrapper-based forward selection (also guided by AUC).

Empirical Results

Table 1 compares DTNB to NB and DTs on 35 UCI datasets. Multi-class datasets were converted into two-class datasets by merging all classes except the largest one. We performed 50 runs of the repeated holdout method, setting aside 66% of the data for training and the rest for testing, and report the mean AUC and standard deviation. Identical runs were used for each algorithm. We used the corrected resampled t-test (Nadeau & Bengio 2003) at the 5% level.

Table 1: Mean AUC and std. dev. w/o attribute selection.

Dataset	DTNB	NB	DT
anneal	0.9970±0.0080	0.9773±0.0138 •	0.9986±0.0037
autos	0.8887±0.0772	0.8613±0.0818	0.9233±0.0569
balance-s	0.9666±0.0192	0.9035±0.0374 •	0.9129±0.0370 •
breast-c	0.6669±0.1090	0.6901±0.1060	0.6432±0.1149
breast-w	0.9922±0.0075	0.9920±0.0076	0.9845±0.0118 •
credit-a	0.9266±0.0318	0.9253±0.0310	0.9199±0.0342
credit-g	0.7554±0.0438	0.7812±0.0522 ○	0.7006±0.0588 •
diabetes	0.8037±0.0573	0.8053±0.0569	0.7971±0.0578
ecoli	0.9868±0.0158	0.9865±0.0150	0.9819±0.0176
glass	0.7485±0.1100	0.7487±0.1036	0.7481±0.1076
heart-c	0.9083±0.0462	0.9109±0.0478	0.8656±0.0524 •
heart-h	0.9206±0.0474	0.9205±0.0487	0.8900±0.0583 •
heart-s	0.8861±0.0612	0.8959±0.0618	0.8777±0.0714
hepatitis	0.8984±0.1063	0.9080±0.1004	0.7767±0.1331 •
horse-c	0.8713±0.0752	0.8365±0.0820	0.8721±0.0478
hypothyroid	0.9950±0.0050	0.9945±0.0035	0.9979±0.0024
ionosphere	0.9533±0.0313	0.9512±0.0302	0.9036±0.0522 •
iris	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000
kr-vs-kp	0.9926±0.0029	0.9525±0.0104 •	0.9946±0.0036 ○
labor	0.9600±0.0762	0.9608±0.0750	0.8633±0.1336
lymphography	0.9202±0.0615	0.9208±0.0584	0.8881±0.0768
mushroom	1.0000±0.0000	0.9981±0.0007 •	1.0000±0.0000
optdigits	0.9909±0.0060	0.9838±0.0066 •	0.9629±0.0132 •
pendigits	0.9919±0.0022	0.9869±0.0028 •	0.9891±0.0038 •
primary-t	0.8777±0.0590	0.8967±0.0503 ○	0.8677±0.0609
segment	0.9992±0.0013	0.9986±0.0020	0.9977±0.0028
sick	0.9560±0.0204	0.9555±0.0199	0.9500±0.0244
sonar	0.8719±0.0725	0.8874±0.0581	0.8255±0.0883
soybean	0.9902±0.0127	0.9656±0.0280 •	0.9649±0.0471
splice	0.9831±0.0048	0.9771±0.0052 •	0.9655±0.0087 •
vehicle	0.9762±0.0144	0.9388±0.0249 •	0.9716±0.0144
vote	0.9886±0.0132	0.9745±0.0191 •	0.9856±0.0129
vowel	0.9967±0.0052	0.9914±0.0107	0.9923±0.0113
waveform	0.9485±0.0100	0.9422±0.0102 •	0.8938±0.0151 •
zoo	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000

•, ○ statistically significant improvement or degradation for DTNB

Table 2: Mean AUC and std. dev. with attribute selection.

Dataset	DTNB$_{AS}$	NB$_{AS}$	DT
anneal	0.9983±0.0075	0.9882±0.0163 •	0.9986±0.0037
autos	0.8934±0.0751	0.8724±0.0848	0.9233±0.0569
balance-s	0.9666±0.0192	0.9669±0.0192	0.9129±0.0370 •
breast-c	0.6615±0.1095	0.6718±0.1083	0.6432±0.1149
breast-w	0.9920±0.0078	0.9910±0.0086	0.9845±0.0118 •
credit-a	0.9298±0.0332	0.9287±0.0318	0.9199±0.0342
credit-g	0.7577±0.0462	0.7788±0.0512 ○	0.7006±0.0588 •
diabetes	0.8024±0.0589	0.8049±0.0570	0.7971±0.0578
ecoli	0.9870±0.0153	0.9871±0.0152	0.9819±0.0176
glass	0.7487±0.1100	0.7493±0.1087	0.7481±0.1076
heart-c	0.9105±0.0468	0.9094±0.0474	0.8656±0.0524 •
heart-h	0.9233±0.0468	0.9197±0.0518	0.8900±0.0583 •
heart-s	0.8831±0.0564	0.8979±0.0633	0.8777±0.0714
hepatitis	0.8960±0.1089	0.8930±0.1045	0.7767±0.1331 •
horse-c	0.8715±0.0757	0.8740±0.0786	0.8721±0.0478
hypothyroid	0.9956±0.0038	0.9968±0.0026	0.9979±0.0024
ionosphere	0.9568±0.0282	0.9596±0.0239	0.9036±0.0522 •
iris	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000
kr-vs-kp	0.9952±0.0024	0.9870±0.0046 •	0.9946±0.0036
labor	0.9575±0.0920	0.9717±0.0822	0.8633±0.1336
lymphography	0.9300±0.0586	0.9185±0.0628	0.8881±0.0768
mushroom	1.0000±0.0000	0.9999±0.0001 •	1.0000±0.0000
optdigits	0.9909±0.0059	0.9927±0.0046	0.9629±0.0132 •
pendigits	0.9936±0.0018	0.9892±0.0026 •	0.9891±0.0038 •
primary-t	0.8770±0.0609	0.8848±0.0567	0.8677±0.0609
segment	0.9994±0.0012	0.9987±0.0019	0.9977±0.0028
sick	0.9544±0.0205	0.9563±0.0196	0.9500±0.0244
sonar	0.8699±0.0703	0.8862±0.0703	0.8255±0.0883
soybean	0.9900±0.0115	0.9930±0.0116	0.9649±0.0471
splice	0.9841±0.0044	0.9823±0.0050 •	0.9655±0.0087 •
vehicle	0.9807±0.0150	0.9680±0.0175 •	0.9716±0.0144
vote	0.9905±0.0096	0.9906±0.0080	0.9856±0.0129
vowel	0.9970±0.0051	0.9941±0.0066	0.9923±0.0113
waveform	0.9479±0.0099	0.9455±0.0098 •	0.8938±0.0151 •
zoo	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000

•, ○ statistically significant improvement or degradation for DTNB$_{AS}$

Table 1 shows that DTNB achieves 11 significant wins against NB and only two significant losses. Against DTs, there are also 11 significant wins and only one significant loss. There are five cases where DTNB is significantly better than both constituent techniques. Four of these datasets are large: *optdigits*, *pendigits*, *splice*, and *waveform*.

Table 2 shows performance when attribute selection is applied to both NB and DTNB. This renders NB's computational complexity quadratic in the number of attributes, lifting it to the same level as that of DTNB. DTNB now achieves seven significant wins against NB, and one significant loss. Compared to DTs, which have built-in attribute selection, DTNB again achieves eleven wins, but this time without a significant loss. For three of the four datasets from above DTNB again improves significantly on both constituent techniques: *pendigits*, *splice*, and *waveform*.

Conclusions

We investigated a simple and efficient semi-naive Bayesian ranking algorithm that splits the set of attributes into two groups: one group assigns class probabilities based on naive Bayes, the other group based on a decision table, and the resulting probability estimates are combined. Empirical results based on AUC show that the combined model performs well compared to stand-alone naive Bayes and decision tables. They also show that this holds true when attribute selection is employed to improve the performance of both naive Bayes and the combined model.

References

Fayyad, U. M., and Irani, K. B. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In *Int. Joint Conf. on AI*, 1022–1027. Morgan Kaufmann.

Nadeau, C., and Bengio, Y. 2003. Inference for the generalization error. *Machine Learning* 52(3):239–281.

Pazzani, M. 1996. Constructive induction of cartesian product attributes. In *Information, Statistics and Induction in Science*, 66–77. World Scientifc.