PERFORMANCE PREDICTION CHALLENGE

Title: Regularized and Averaged Selective Naïve Bayes Classifier

Name, address, email: Marc Boullé,
France Telecom R&D, 2, avenue Pierre Marzin, 22307 Lannion cedex – France
marc.boulle@francetelecom.com

Acronym of your best entry: SNB(CMA) + 10k F(3D) tv

References:

M. Boullé, "Regularization and Averaging of the Selective Naïve Bayes classifier", International Joint Conference on Neural Networks, 2006.
M. Boullé, "MODL: a Bayes Optimal Discretization Method for Continuous Attributes", Machine Learning, to be published.

Method:

Our method is based on the Naive Bayes assumption.
All the input features are preprocessed using the Bayes optimal MODL discretization method.
We use a Bayesian approach to compromise between the number of selected features and the performance of the Selective Naïve Bayes classifier: this provides a regularized feature selection criterion. The feature selection search is performed using alternate forward selection and backward elimination searches on randomly ordered feature sets: this provides a fast search heuristic, with super-linear time complexity with respect to the number of instances and features. Finally, our method introduces a variant of feature selection: feature "soft" selection. Whereas feature "hard" selection gives a "Boolean" weight to the features according to whether they selected or not, our method gives a continuous weight between 0 and 1 to each feature. This weighing schema of the features comes from a new classifier averaging method, derived from Bayesian Model Averaging.
The method computes the posterior probabilities of the classes, which is convenient when the classical accuracy criterion or the area under the ROC curve is evaluated. For the challenge, the Balanced Error Rate (BER) criterion is the main criterion. In order to improve the BER criterion, we adjusted the decision threshold in a post-optimization step. We still predict the class having the highest posterior probability, but we artificially adjust the class prior probabilities in order to optimize the BER criterion on the train dataset.
For the challenge, several trials of feature construction have been performed in order to evaluate the computational and statistical scalability of the method, and to naively attempt to escape the naïve Bayes assumption:

- 10k F(2D): 10 000 features constructed for each dataset, each one is the sum of two randomly selected initial features,
- 100k F(2D): 100 000 features constructed (sums of two features),
- 10k F(3D): 10 000 features constructed (sums of three features).

The performance prediction guess is computed using a stratified tenfold cross-validation.

Results:

In the challenge, we rank 7th as a group and our best entry is 26th, according to the average rank computed by the organizers. On 2 of the 5 five datasets (ADA and SYLVA), our best entry ranks 1st.

Our method is highly scalable and resistant to noisy or redundant features: it is able to quickly process about 100 000 constructed features without decreasing the predictive performance.
Its main limitation comes from the Naïve Bayes assumption. However, when the constructed features allow to "partially" break the naïve Bayes assumption, the method succeeds in significantly improve its performances. This is the case for example for the GINA dataset, which does not fit well the naïve Bayes assumption: adding randomly constructed features allows to improve the BER from 12.83% down to 7.30%.
The AUC criterion, which evaluates the ranking of the class posterior probabilities, indicates high performances for our method.

Dataset	Our best entry					The challenge best entry				
	Test AUC	Test BER	Ber Guess	Guess Error	Test score	Test AUC	Test BER	Ber Guess	Guess Error	Test score
ADA	0.9149	0.1723	0.1650	0.0073	0.1793 (1)	0.9149	0.1723	0.1650	0.0073	0.1793
GINA	0.9772	0.0733	0.0770	0.0037	0.0767	0.9712	0.0288	0.0305	0.0017	0.0302
HIVA	0.7542	0.3080	0.3170	0.0090	0.3146	0.7671	0.2757	0.2692	0.0065	0.2797
NOVA	0.9736	0.0776	0.0860	0.0084	0.0858	0.9914	0.0445	0.0436	0.0009	0.0448
SYLVA	0.9991	0.0061	0.0060	0.0001	0.0062 (1)	0.9991	0.0061	0.0060	0.0001	0.0062
Overall	0.9242	0.1307	0.1306	0.0096	0.1399 (26.4)	0.8910	0.1090	0.1040	0.0079	0.1165 (6.2)

Code: Our implementation was done in C++.

Keywords: Discretization, Bayesianism, Naïve Bayes, Wrapper, Regularization, Model Averaging