Subspace projections – an approach to variable selection and modeling

Thomas Martini Jørgensen[1] and Christian Linneberg[1,2]
[1]Risø National Laboratory
P.O. Box 49, DK-4000 Roskilde, Denmark
Phone: +45-4677-4677, Fax: +45-4677-4565
email:{thomas.martini, christian.linneberg}@risoe.dk
[2]Intellix A/S
H.C. Ørsteds Vej 4, DK-1879 Frederiksberg, Denmark
Phone: +45-7023-3700, Fax: +45-7023-2700

ABSTRACT: This communication describes an approach to data modeling based on detecting significant subspaces in the original variable space. The "quality" of the subspaces is determined from a linear discriminant analysis. The discovered subspaces are used as an alternative representation of the input data and are fed to an n-tuple classifier (also known as a RAM-based neural network). For the present problem a post-analysis shows that the non-linear part of the model structure is not essential. A simple linear discriminant analysis carried out on the input variables gives similar performance (and is actually superior on the test set).

KEYWORDS: Subspaces, n-tuple classifier

INTRODUCTION

The present report describes one of the 43 solutions submitted to the "CoIL challenge 2000" competition. The approach we used is divided in two main parts. The first part seeks to find significant directions in the original variable space with respect to separating the classes present in the problem. Projecting the data set onto the obtained set of subspaces then gives us a new representation of the data, which in the following step is used to train an n-tuple classifier (Bledsoe and Browning (1959)). Essential parameters of the n-tuple architecture are set by applying leave-one-out cross-validation tests. The combined pre-processing and network training is performed three times as we make use of a three-fold cross-validation approach to assess the performance of the obtained models. In this way we also end up with three models which we finally combine in a small ensemble network, which are then applied when classifying the examples of the test set.

Besides adding a number of derived variables the described approach if fully automated in the sense that we have not by manual inspection discovered any patterns, neither have we from inspection decided to omit any variables.

From the post-competition discussions it appears that standard stepwise variable selection approaches are highly suitable for the problem at hand. The basic reasoning for choosing an alternative strategy, which picks several variable at a time, is to have a chance of finding significant combinations of variables that might be missed by applying a stepwise strategy. Furthermore the approach we have used aims at detecting several possible "solutions", which are then combined in order to develop a robust scheme.

As the n-tuple architecture can be considered as an ensemble of decision tree branches, it is actually possible to extract rules from the final architecture that mimics the ones that are obtained from decision trees. For the present study we have however only studied the top three variables that are found to be of most importance for modelling the problem at hand.

The applied model anticipates that a non-linear approach is necessary to obtain a suitable model. However, the discussions that have taken place after the results of the competition were published have indicated that a linear approach might be sufficient in the present case. We have therefore tried to replace the n-tuple classifier with a simple majority voting scheme operating on the outputs of the linear discriminant models that we use when selecting the subspaces. Furthermore we present the results obtained by applying a linear discriminant analysis on the set of input variables.

DATA

The training data set available for this study consists of 5822 customer records from an insurance company, each record containing both sociodemographic data and product ownership data. The task is to find people with a caravan/mobile home policy. The test set consists of 4000 records and the task is to select a subset of 800 from these 4000 customers that are likely to have/be interested in a caravan policy.

Throughout the data processing we have used a three fold splitting of the data. For each fold the parameters of the corresponding model is estimated using leave-one-out cross-validation, whereas the performance of the model is assessed by measuring the performance on the part of the examples not used for training the model.

UNFOLDING CATEGORICAL VARIABLES

The overall modelling scheme is as follows. First the data is unfolded; as an example consider the variable "MOSTYPE" – Customer Subtype – a categorical variable describing to which of a number (41 in total) of subtypes the current customer belongs, e.g. "Stable family" or "Single youth". The variable "MOSTYPE" is unfolded to 41 indicator variables (named "MOSTYPE_1" etc.), each indicating whether or not a given customer belongs to the corresponding subtype. By unfolding all such variables in the data set it is expanded from 85 descriptive variables to 134 variables. This is a standard way of treating categorical variables.

CALCULATED VARIABLES

We have added a few variables, which combine information from the original ones. For all pairs of insurance variables we have calculated the average contribution per insurance, e.g. "dPERSAULT" is calculated as "PPERSAUT" divided by "APERSAUT", i.e. contribution to car policies divided by the number of car policies. Furthermore we added the following summations: "Payment", the total contribution to insurance, given as the sum of the original variables numbered from 44 to 64 and "NumberOfPolicies", the total number of policies, obtained by summing the variables numbered 65 to 85 in the original data. With these two sums and the 21 ratio variables we end up with a total of 157 variables.

PRE-PROCESSING

As an alternative to standard variable selection techniques like e.g. stepwise selection, we have used the following methodology. A large number of simple linear discriminant models are trained on a subset of the training data. All simple models are evaluated and only the best 2 % of the models are kept for further processing. Furthermore, as each model is only allowed to use 10 randomly selected input, this scheme is in effect performing a variable selection, and for each model the corresponding discriminative axis is extracted. Each model is only trained on a random subset (2/3) of the training examples in order to be less influenced by a few outliers. This is the same principle underlying the noise robustness of the Bagging scheme, see Breiman (1996) and Dietterich (2000). As a result the original variables are in general replaced by new ones obtained as weighted sums over the original ones.

With respect to estimating importance of the original variables one can count how often each of the original variables are used in the sub-models. The top-6 variables found by this simple scheme are listed in Table I. The first column indicates how frequent the variable is selected relative to the expected frequency obtained by a random selection of the sub-models. The variables "PPERSAUT", "APERSAUT" and "dPERSAUT" might be considered to contain the same information and they are also only rarely used together, in total 61% of the selected sub-models use at least one of these three variables. This corresponds to an increased likelihood of 9.6 times.

We have also found the most frequent pairs of variables. Table IIa shows the top-ten pairs and IIb shows the top-ten pairs where "Payment" is not one of the variables (As "Payment" is the most frequent single variable, it is also present in most of the most-frequent pairs).

CLASSIFICATION

We have used the so-called n-tuple classifier (originally proposed by Bledsoe and Browning in 1959) to classify the data. Input to this classifier must be converted to binary features and we perform the required discretisation by using the entropy discretisation scheme described in Fayyad and Irani (1993). Furthermore, essential parameters of the architecture are set by applying leave-one-out cross-validation tests, see Stone (1974), as outlined in Linneberg and Jørgensen (1998). As we operate with a three part splitting of the training examples we end up with three models. We would like to make use of all training examples when predicting the outcome of the test set, so we combine the predictions in a small ensemble network, by associating a confidence measure (weighting) to the prediction obtained from each model. The confidence measure is a worst case estimate of what we denote the *critical example number*, defined as the number of examples that should be extracted from the training set in order to change the prediction. Accordingly this number defines a kind of example support for a given decision. This confidence definition was also used to rank the overall predictions of the ensemble model.

VALIDATION

As we used a three-fold cross-validation approach we were also able to estimate the performance of the obtained models on the test set. From these calculations we would expect that between 16 % and 17 % of the selected 800 test examples would actually have a caravan insurance policy, given that the training distribution is representative for the problem. If the 800 examples were selected at random, one would expect to find 6 %.

RESULTS

The submitted ensemble model scored 107 customers that would be interested in buying a caravan insurance policy. As we now know the target values for the test examples, we have repeated the modelling process five times, finding 106, 105, 101, 108 and 105 customers or 105 ± 2.55 in terms of mean and standard deviation. This is slightly lower that expected from the validation process.

Increased likelihood	Variable	Description
798%	Payment	Total payment
518%	PPERSAUT	Contribution car policies
288%	APERSAUT	Number of car policies
230%	dPERSAUT	Average contribution per car policies
204%	PBRAND	Contribution fire policies
204%	NumberOfPolicies	Total number of policies

Table I: The most frequently used variables in the described model.

a)

Increased likelihood	Variable pair	
884%	Payment	AWALAND
839%	Payment	PPERSAUT
680%	Payment	MOSTYPE_08
658%	Payment	dPERSAUT
658%	PBRAND	PPERSAUT
658%	Payment	PBRAND
658%	Payment	PMOTSCO
635%	Payment	MOSHOOFD_10
635%	Payment	MOSTYPE_26
612%	DBRAND	PPERSAUT

b)

Increased likelihood	Variable pair	
658%	PBRAND	PPERSAUT
612%	dBRAND	PPERSAUT
499%	PPERSAUT	MHHUUR
499%	NumberOfPolicies	PPERSAUT
476%	ABRAND	PPERSAUT
476%	PPERSAUT	MOSTYPE_08
454%	dPERSAUT	NumberOfPolicies
454%	NumberOfPolicies	APERSAUT
431%	APERSAUT	MINKGEM
431%	PPLEZIER	PPERSAUT

Table II: a) Most frequently used pair of variables and their increased likelihood. b) As a) but without “Payment”.

A lift chart for the submitted model is depicted in Figure 1. The gain of combining the three individual models is small but existing, especially in the low end of the graph. The three individual models respectively score 96, 101 and 102 caravan policies, whereas the combined model scores 107 caravan policies.

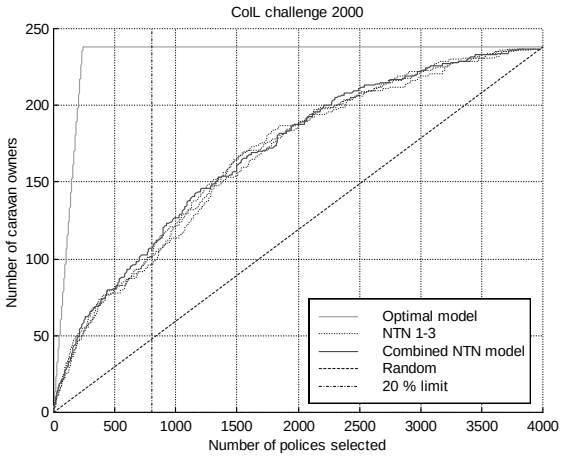

Figure 1: Lift chart for the combined model on the test set.

LINEAR VS. NON-LINEAR POST-PROCESSING

After the results of the competition were available, there were several observations made indicating that simple linear approaches perform quite successful on the present task. As our pre-processing stage actually involves application of a linear discriminant (LD) analysis in different subspaces it is therefore of interest to compare a simple vote decision after the pre-processing stage with the above solution using a non-linear neural network. Instead of projecting the data onto the axis found from the subspace search, we could simply do a majority voting among the set of associated discriminant models or apply a linear model to the projected data. Table III compares the submitted solution to such linear discriminant approaches. The LD schemes typically have around 10 caravan owners more within the top-800 ranked list of the test examples than the submitted neural network solution. It is also interesting to note that the validation result of the 3-fold linear discriminant analysis is comparable to the performance actually achieved on the test set. If one should select the model based on the validation result one would prefer the submitted solution. This might indicate that one should consider the validation results in combination with the complexity of the model.

DESCRIPTIVE TASK

In the search for simple rules for classifying the data we have examined the three most frequent variables used in our subspace models. For each variable, we have based on the training set used an entropy-based splitting criterion to determine the best split value to use. The corresponding performances of the associated rules are listed in Table IV. Using the performance on the training examples, one would use the rule "Payment $\geq$ 8 and PPERSAUT $\geq$ 6". Using this rule on the test set "scores" 102 caravan policies out of the top 800 ranked examples. If we would only use one of the three variables then the rule "Payment $\geq$ 8" would be selected. It scores 90 caravan policies within the top 800 list.

CONCLUSIONS

In the present study it is clear that applying our neural network model to the pre-processed data actually degrades the performance on the test set as compared to a use of linear discriminant analysis on the variables. This shows rather clearly that one should only apply non-linear models where they are needed, and accordingly one must check if they really are. In the present case the validation results however also show that it can be difficult to judge whether anything is gained from applying a non-linear approach.

Model	Train	Validation	Test
Submitted	Not informative for an n-tuple net	16.5 %	13.4 %
3-fold LD on original + 23 extra variables	18.2 %*	15.2 %	15.0 %
LD on original variables	17.1 %	N/A	14.6 %
LD on original + 23 extra variables	17.2 %	N/A	14.9 %
Ensemble of subspace LD models (one of three folds)	Not calculated	13.4 %	14.4 %
LD on projected data	17.0 %	N/A	14.8 %

Table III: Hit rates within the top 20% ranked output lists obtained from the listed models (*The average value over the three folds).

Rule	Train	Test
Payment ≥ 8	11.8 %	11.2 %
PPERSAUT ≥ 6	11.1 %	9.9 %
APERSAUT ≥ 1	9.3 %	8.5 %
Payment ≥ 8 and PPERSAUT ≥ 6	13.8 %	12.7 %
Payment ≥ 8 and PPERSAUT ≥ 6 and APERSAUT ≥ 1	13.8 %	12.7 %

Table IV: Simple rules and performances

REFERENCES

Bledsoe, Woodrow Wilson; Browning, Iben, 1959, "Pattern recognition and reading by machine", Proceedings of the Eastern Joint Computer Conference, pp. 225-232.

Breiman, Leo, 1996, "Bagging Predictors", Machine Learning, vol. 24, pp 123-140.

Dietterich, Thomas G., 2000, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization", Machine Learning, vol 40 (2). pp 139-158.

Fayyad, U. M.; Irani, K. B., 1993, "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning", Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Francisco, pp. 1022-1027.

Linneberg, Christian; Jørgensen, Thomas Martini, 1998, "Cross-validation techniques for n-tuple based neural networks", Proceedings of the Ninth Workshop on Virtual Intelligence/Dynamic Neural Networks, eds. T. Lindblad, M.L. Padgett, and J. Kinser. SPIE vol 3728, pp. 266-277.

Stone, M., 1974, "Cross-validatory choice and assessment of statistical predictions", Journal of the Royal Statistical Society B, vol. 36, pp. 111-147.