On the Homogeneous Ensembling with Balanced Random Sets and Boosting

Vladimir Nikulin

Department of Mathematical Methods in Economy,
Vyatka State University, Kirov, Russia
vnikulin.uq@gmail.com

Abstract. Ensembles are often capable of greater prediction accuracy than any of their individual members. As a consequence of the diversity between individual base-learners, an ensemble will not suffer from overfitting. On the other hand, in many cases we are dealing with imbalanced data and a classifier which was built using all data has tendency to ignore minority class. As a solution to the problem, we propose to consider a large number of relatively small and balanced subsets where representatives from the both patterns are to be selected randomly. Using different pre-processing technique combined with available background knowledge, which may have subjective treatment, we can generate many secondary databases for training. The relevance of those databases maybe tested with five folds cross-validation (CV5). Further, we can use CV5-results to optimise blending structure. Note that it is appropriate to use different software for CV5 evaluation and for the computation of the final solution. Our model was tested online during an International Carvana data mining Contest on the Kaggle platform. This Contest was highly popular and attracted 582 actively participating teams, where our team was awarded 2nd prize.

Keywords: ensembling, blending, decision trees, boosting, neural nets, cross validation, classification.

1 Introduction

In line with an ensembling theory, we are interested to generate a variety of high quality solutions for the problem, and there are two main directions how to do that. One very popular way is associated with 1) the usage of the different software in application to the same database, or 2) we can apply the same software (named classifier C1) to different databases. Based on our experience, the second direction is a more preferable in the case of Carvana data.

The next fundamental question is how to link (or how to blend) many different solutions in a most optimal way [1]. The answer is a quite straightforward: it appears to be natural to use cross-validation (CV) with fixed design matrix as a criterion for blending, where the number of 5 folds seems to be quite sufficient. However, implementation of the CV5 may represent significant computational

J.T. Yao et al. (Eds.): RSCTC 2012, LNAI 7413, pp. 180–189, 2012.

problem, particularly, in the case if we are dealing with imbalanced data, and have to construct the final solution as a homogeneous ensemble of many base learners, each of which is a function of the randomly selected balanced subset. Note that the main target of the CV is to compare different databases, where the quality of any particular solution may not necessarily be high.

We can implement here very important principle of invariance, which maybe described very briefly as follows. Suppose, we have another classifier, named C2, which is much faster compared to the C1. Note that the quality of the classification (or quality of the patterns separation) by the C2 maybe much poorer compared to the C1, but it is not essential here. Validity of the hypothesis of invariance is a subject of the fundamental importance. According to this hypothesis, the scaling in the quality of performance between C1 and C2 is about the same around all the secondary databases. Based on our experience with Carvana data, the hypothesis of invariance is true. Therefore, we can use C2 to conduct all the necessary experiments with CV5 in application to 10-20 secondary databases. After collection of the CV5 experimental results with fixed design matrix, we can optimise weighting coefficients for blending. Then, we can recompute solutions for the selected secondary databases with C1, and apply blending coefficients in order to calculate the final solution. In this particular project we used GBM in R as C1, and Neural Nets (NNs) in CLOP, Matlab, as C2.

2 Data Pre-processing

Carvana database[1] includes two parts 1) training with 72983 samples, where 8976 are positive (that means problematic), and all the other samples are negative (that means normal); 2) testing with 48707 samples (unlabelled).

The list of 36 original features is given in Table 1, where 3 features (index=0) were excluded from further consideration. Remaining 33 features were divided into 4 parts:

1) numerical (15 features in total including target variable);
2) textual (14 features in total);
3) categorical (3 features in total);
4) PurchDate.

Remark 1. Any missing values were replaced by "-1". "PurchDate" values were transferred to four integer values: 1) year (2009 or 2010); 2) month; 3) day of the week; and 4) day of the month.

2.1 Textual data

Using special software, written in Perl, we created list of all text-units for any feature, and counted the numbers of their occurrences in the training database.

[1] `http://www.kaggle.com`

Subject to the sufficient level (see, column Δ in Table 1) any particular text-unit was given sequential positive index, or zero index, which means infrequent (insignificant) value.

Table 1. List of 36 original features, where index=0 means that the feature was excluded from modelling (3 in total); index=1 - numerical feature (15 in total including target variable); index=2 - date (one feature); index=3 - textual features (14 features in total); index=4 - categorical features (3 features in total)

N	Field Name	Type	Index	Δ
1	RefID	NA	0	
2	IsBadBuy	target	1	
3	PurchDate	date	2	
4	Auction	txt	3	100
5	VehYear	year	1	
6	VehicleAge	num	1	
7	Make	txt	3	20
8	Model	txt	3	40
9	Trim	txt	3	39
10	SubModel	txt	3	20
11	Color	txt	3	50
12	Transmission	txt	3	100
13	WheelTypeID	cat	4	
14	WheelType	txt	3	100
15	VehOdo	num	1	
16	Nationality	txt	3	100
17	Size	txt	3	100
18	TopThreeAmericanName	txt	3	100
19	MMRAcquisitionAuctionAveragePrice	num	1	
20	MMRAcquisitionAuctionCleanPrice	num	1	
21	MMRAcquisitionRetailAveragePrice	num	1	
22	MMRAcquisitonRetailCleanPrice	num	1	
23	MMRCurrentAuctionAveragePrice	num	1	
24	MMRCurrentAuctionCleanPrice	num	1	
25	MMRCurrentRetailAveragePrice	num	1	
26	MMRCurrentRetailCleanPrice	num	1	
27	PRIMEUNIT	txt	3	50
28	AcquisitionType	NA	0	
29	AUCGUART	txt	3	50
30	KickDate	NA	0	
31	BYRNO	cat	4	
32	VNZIP	cat	4	
33	VNST	txt	3	20
34	VehBCost	num	1	
35	IsOnlineSale	num	1	
36	WarrantyCost	num	1	

As a consequence of the above pre-processing transformation/treatment, we produced two completely numerical matrices (for training and for testing) with 35 features each and without any missing values.

3 Synthetic Features

Let us consider 4 features: VehicleAge, VehOdo, VehBCost and WarrantyCost, where the first one is discrete, and the others are continuous.

Continuous features maybe investigated using method of the moving averages, applied to the sorted (according to the selected feature) vector of the target variable. We have found that the "get kicked" probability is an increasing function of VehOdo (V_{15}, see Table 1) and WarrantyCost (V_{36}), and decreasing function of VehBCost (or of any other Cost-related variable). Based on this observation, we can consider the following structure for the new (synthetic) variable:

$$f_{\text{new}} = \frac{V_{23}}{(1 + C_1 V_{36})(C_2 + V_{15} + C_3 V_6)}, \tag{1}$$

where non-negative parameters $C_i, i = 1, \ldots, 3$, were selected (optimised using specially designed software written in Matlab) in order to maximise diversity of the moving average corresponding to (1), see Figure 1(d).

Two sets of the coefficients C are given in Table 2. Additionally, we used third synthetic variable:

$$f_{\text{new}}^{(3)} = \frac{V_{23} + C_4 V_{34}}{(C_5 + V_{36})}, \tag{2}$$

where $C_4 = 1.49, C_5 = 173$.

Table 2. Two sets of coefficients for synthetic variables

C_1	C_2	C_3
0	267	14354
9.8	333	9229

Remark 2. Equation (1) represents just an example. In general terms, definition of the new synthetic variable may include many multipliers. For example, in the case of a very popular Credit Contest on the Kaggle platform we used 13 multipliers, corresponding to the different original variables. After optimisation of the coefficients, we can split the whole new variable by considering sub-products of 2, 3, 4,..., components. As a consequence, we shall create many new synthetic variables, which cannot be replaced by only one variables as a product of all components.

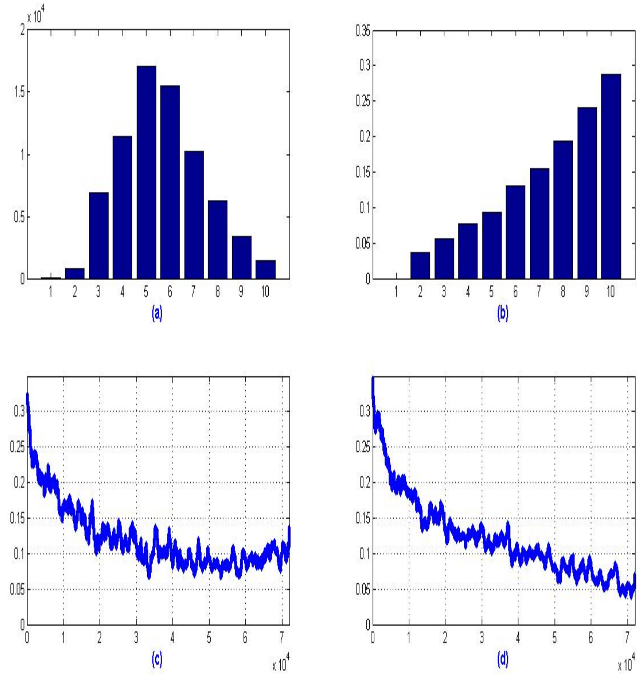

Fig. 1. (a) Numbers of occurrences for VehicleAge; (b) empirical probabilities for VehicleAge (ten values in total); (c) moving averages for VehBCost; (d) moving averages for the synthetic feature (see Section 3)

3.1 On the Comparison between Different Cost-Variables

In the most early stages of the Contest, we had noticed that the differences between Costs are much more informative compared to the Cost-variables themselves. Using database with 37 variables (=35-1+3), as described above, where we excluded VehYear variable (as it replicates VehicleAge) and replaced Cost-variables (NN19-26) by their differences with VehBCost (8 differences in total), we were able to achieve public score on the LeaderBoard 0.26023 in the terms of Gini Index.

As a next step, we decided to replace above 8 Cost-differences by all the 36 Cost-differences (the number of all combinations from 9 by 2):

```
for $i = 1, \ldots, 8$,
    for $j = i + 1, \ldots, 9$,
        $r_{ij} = 1 - \frac{X_i+1}{X_j+1}$
    end;
end;
```

where by $X_j, j = 1, \ldots, 9$, we denote Costs/Prices (see, features NN19-26, 34 in Table 1).

New database includes 65 features. With this database, we observed a very significant improvement: public score on the LeaderBoard 0.26608.

4 Homogeneous Ensembling with Balances Random Sets

In many cases, ensembles have significantly better prediction accuracy compared to their individual members [2]. As a consequence of the diversity between individual base-learners, an ensemble will not suffer from overfitting. On the other hand, in many cases we are dealing with imbalanced data and a classifier which was built using all data has tendency to ignore minority class. As a solution to the problem, we propose to consider a large number of relatively small and balanced subsets where representatives from the both patterns are to be selected randomly [3].

4.1 On the Boosting Principles Applied to the Selection of the Balanced Subsets

In our previous publications [3], [4], we used a sequence of balanced subsets, which were selected from the training set independently (one subset was completely independent from the next, and so on). However, it appears to be logical if we shall apply here principles of boosting [5] based on the latest solution (known, also, as base-learner).

Principle of Complexity in Application to the Labelled Data. Let us describe the proposed boosting model in more details. Suppose that $y_t \in \{0, 1\}$ is the target variable and $s_t^{(\alpha)} \in [0, \dots, 1]$ is the training solution corresponding to the sample t, α is a sequential index of the balanced random subset. Then, we shall select sample t as a prospective to be included in the following balanced subset $\alpha + 1$ subject to the following conditions

$$\xi \leq \xi_1 \ if \ y_t = 1; \tag{3a}$$

$$\xi \leq \xi_2, \ otherwise, \tag{3b}$$

where

$$\xi_i = c_{i1} + (c_{i2} + c_{i3} \cdot \phi) \cdot w(y_t, s_t^{(\alpha)}), i = 1, \dots, 2, \tag{4}$$

$$w(y, s) = |y - s|^{\beta}, \tag{5}$$

where ξ and $\phi \in [0, \dots, 1]$ are standard uniform random variables, $\beta > 0$ and $c_{ij} > 0, i = 1, \dots, 2, j = 1, \dots, 3$, are regulation parameters. For example, we can select $\beta = 0.35$, and the recommended values for coefficients c are given in Table 3.

Table 3. Recommended values for the matrix of coefficients $c_{ij} > 0, i = 1, \ldots, 2, j = 1, \ldots, 3$.

0.25	0.4	0.35
0.12	0.08	0.06

Remark 3. We can see that values in the first row of Table 3, which correspond to the minority class are much bigger compared to the second row, which corresponds to the majority class. As a direct consequence, selection according to (3a) and (3b) will create relatively balanced subset. However, we considered selection in accordance with (3a) and (3b) as just a preliminary. After that, we conducted final adjustment to ensure that the relation between positives and negatives is exactly as required.

Remark 4. The function (5) in (4) represents a very important boosting multiplier to ensure that "difficult" samples will be given higher probability to be selected.

Principle of Simplicity in Application to the Unlabelled Data. We can extend selection (3a) and (3b) to the unlabelled test set. However, there is a fundamental difference between treatment of labelled and unlabelled data. In the case of labelled data, we shall be selecting more complex samples, but in the case of unlabelled data, we shall be selecting simpler samples with stronger indication regarding their classification in accordance with available training solution. We used this semi-supervised approach in application to the Credit Challenge on the Kaggle platform, where the data were stronger imbalanced and the quality of classification according to the AUC was significantly higher compared to the Carvana Challenge.

5 Some Other Ways to Construct Secondary Training Datasets

In Section 3.1 we introduced 36 ratios $r_{ij}, i = 1, \ldots, 8, j = i + 1, \ldots, 9$. Clearly, all those ratios have different importance, and we have found that the following 4 relations are the most influential: 1) $\{19, 23\}$; 2) $\{20, 24\}$; 3) $\{21, 25\}$ and 4) $\{22, 26\}$, where indexes of the involved features are given in Table 1. The next in line were 8 pairs: NN19-26 against N34.

Firstly, we decided to apply another formula to compare different Costs:

$$q_{ij} = \log \frac{X_i + \Delta}{X_j + \Delta}, \tag{6}$$

where we used value $\Delta = 100$ as a smoothing parameter.

Further, we added to the model all possible sums of the first 4 the most influential relations, plus an indicator whether or not all 8 involved Costs are available. In total, we calculated the block $\mathcal{B}$ of $C_4^2+C_4^3+C_4^4+1 = 12$ additional features compared to the previous database with 65 features, which is described in Section 3.1.

With this database (77 features) we observed LeaderBoard public score of 0.26810.

5.1 The Best Single Model

Compared to the previous database with 77 features, we removed feature N35 (see Table 1), also, we removed the indicator of the presence of eight Cost features.

As a very important innovation, we introduces a new definition, which represents a more advanced development compared to (6):

$$\lambda_{ij} = \frac{q_i - q_j}{q_i + q_j}, \tag{7}$$

where

$$q_i = \log \frac{X_i + \mathit{\Delta}}{X_0 + \mathit{\Delta}}, i = 1, \dots, 8,$$

X_0 is feature N34 (VehBCost).

Using above definition (7), we computed 4 new features corresponding to the pairs: 1) $\{19, 23\}$; 2) $\{20, 24\}$; 3) $\{21, 25\}$ and 4) $\{22, 26\}$. Plus, we re-computed 11 features from the block $\mathcal{B}$. Consequently, new database included 79 features (= 77-2+4), and we observed LeaderBoard score 0.26867.

Remark 5. In order to reduce overfitting, we are interested to increase random factor in the model. It appears to be logical to split all the 79 features into 4-5 blocks, where importance of the features within any particular block is about the same. As it was discussed in Section 4, the final classifier represents an average of the base-learners, each of which is based on the randomly selected balanced subset. Importantly to note that the features in the model were, also, selected randomly from any particular block, based on our assessment of how important this block is.

6 Blending of the Different Databases with Neural Nets

In the above section we described three secondary databases with 65, 77 and 79 features. In fact, we created about 20 databases with up to 142 features.

As a next step, we can test any particular database using cross-validation as a standard tool for validation of the classification model, where five folds appears to be quite sufficient. Suppose, we are using the same design (known, also, as splitting) matrix for CV in application to all databases. Then, after computation of the CV-solutions for different databases, we consider performance of the linear combinations of those solutions (known, also, as blend).

After optimisation of the non-negative weighting coefficients, we can compute test-solutions for the selected datasets, which correspond to the sufficiently large weighting coefficients, and compute blend of those particular solutions for a final submission.

An implementation of the above scheme may require a lot of computational time taking into account the fact that any single solution represents a homogeneous ensemble of 100-200 base-learners (each of which corresponds to the randomly selected balanced subset).

6.1 Principle of Invariance

In order to reduce significantly the computational costs, we can implement a principle of invariance. In accordance with this principle, the quality of the CV-solutions are not important in an absolute scale. In contrast, important are relations between different CV-solutions.

We conducted CV5 with Neural Nets function from the Matlab-based CLOP package[2], which is significantly faster compared to the GBM function in R. Evaluation of one particular database with 200 balanced random subsets took about 4 hours time. The following CV5 results were observed: 0.270632 (F79), 0.266423 (F77) and 0.26154 (F65), where numbers in the brackets indicate numbers of the features in the corresponding database.

Remark 6. It is interesting to note that there were several other databases with CV5 result better than 0.26154. However, inclusion of those databases in the blend produced worse result.

The best solution (in both public and private) was a blend of F79, F77 and F65 solutions (used GBM package in R) with the following weighting coefficients: $\{\frac{100}{170}, \frac{55}{170}, \frac{15}{170}\}$. It produces Gini score 0.26885 in public (4th out of 582 participating teams), and 0.26655 in private (3rd place in the Contest).

7 Concluding Remarks

Selection bias [6] or overfitting represents a very important and challenging problem. As it was noticed in [7], if the improvement of a quantitative criterion such as the error rate is the main contribution of a paper, the superiority of a new algorithms should always be demonstrated on independent validation data. In this sense, an importance of the data mining contests is unquestionable. The rapid popularity growth of the data mining challenges [8] demonstrates with confidence that it is the best known way to evaluate different models and systems. Based on our own experience, cross-validation (CV) maybe easily overfit as a consequence of the intensive experiments. Further developments such as nested CV [9] are computationally too expensive [7], and should not be used

[2] http://clopinet.com/CLOP/

until it is absolutely necessary, because nested CV may generate secondary serious problems as a result of 1) the dealing with an intense computations, and 2) very complex software (and, consequently, high level of probability to make some mistakes) used for the implementation of the nested CV. Moreover, we do believe that in most of the cases scientific results produced with the nested CV are not reproducible (in the sense of an absolutely fresh data, which were not used prior).

Generally, we are satisfied with our results, and consider blending model applied to the different databases as a main innovation proposed in this paper. Note, that using conceptually similar method as described in this paper, we were able to achieve 9th place out of 970 actively participating teams in another data mining Contest, named “Credit”, which was, also, based on the Kaggle platform.

References

[1] Koren, Y.: The BellKor Solution to the Netflix Grand Prize, Wikipedia, 10 pages (2009)

[2] Wang, W.: Some fundamental issues in ensemble methods. In: World Congress on Computational Intelligence, Hong Kong, pp. 2244–2251. IEEE (2008)

[3] Nikulin, V.: Classification of imbalanced data with random sets and mean-variance filtering. International Journal of Data Warehousing and Mining 4(2), 63–78 (2008)

[4] Nikulin, V., McLachlan, G.: Classification of imbalanced data with balanced random sets. Journal of Machine Learning Research, Workshop and Conference Proceedings 7, 89–100 (2009)

[5] Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55, 119–139 (1997)

[6] Heckerman, J.: Sample selection bias as a specification error. Econometrica 47(1), 153–161 (1979)

[7] Jelizarow, M., Guillemot, V., Tenenhaus, A., Strimmer, K., Boulesteix, A.-L.: Over-optimism in bioinformatics: an illustration. Bioinformatics 26(16), 1990–1998 (2010)

[8] Carpenter, J.: the best analyst win. Science 331, 698–699 (2011)

[9] Cudeck, R., Browne, M.: Cross-validation of covariance structures. Multivariate Behavioral Research 18(2), 147–167 (1983)