Selective Naive Bayes Regressor with Variable Construction for Predictive Web Analytics

Marc Boullé
Orange Labs
2 avenue Pierre Marzin
22300 Lannion, France
marc.boulle@orange.com

ABSTRACT

We describe our submission to the ECML/PKDD 2014 Predictive Web Analytics discovery challenge, where the objective is to predict the number of visits and messages on Twitter and Facebook after 48 hours, given a time series of such observations collected during the first hour after publication of each URL. It exploits a selective naive Bayes regressor, together with automatic feature construction from the input time series. The data of the challenge is represented using a multi-tables schema, with pages as the main statistical units and the time series records in a secondary table. Using a small set of construction rules, one thousand of new features are created automatically to enrich the representation of pages. These features are then preprocessed to assess their relevance and a small subset of them are selected using the selective naive Bayes regressor. Our submission, obtained almost automatically, was ranked 3$^{\text{rd}}$ on each task.

1. INTRODUCTION

The ECML/PKDD 2014 Predictive Web Analytics discovery challenge [1] is related to a problem of regression from time series input data. A corpus of 60,000 URL coming from 100 websites is provided, with a representation consisting of a times series of number of visits, average time per visit, number of messages on Twitter and Facebook in the first hour after publication of the URL (series of 12 observations, with 5 minutes window). The objective is to predict the number of visits and messages on Twitter and Facebook after 48 hours, which turns into three regression tasks. In this paper, we present the method we used at the discovery challenge. It mainly exploits the results of a Selective Naive Bayes regressor trained for each of the three target variables, together with variable construction from the time series input data (see Section 2). The challenge was an opportunity to evaluate the variable construction framework introduced for supervised classification in [2] on a new dataset and for a regression task. In Section 3, we describe our submission and evaluate our results. Finally, Section 4 summarizes the paper.

ECML/PKDD Discovery Challenge on Predictive Web Analytics.
Nancy, France, September 2014.

2. METHOD USED IN THE CHALLENGE

In this section, we summarize the method applied in the challenge. It first uses the Naive Bayes (NB) rank regressor described in [3], which aims at predicting the rank of a numerical target variable given a set of numerical or categorical input variables. This regressor is improved using regularized variable selection, detailed in [4]. Finally, the time series input data available in the challenge is represented using a multi-table formalism and exploited using the automatic variable construction method introduced in [2].

2.1 Naive Bayes Rank Regression

The considered supervised learning task consists in predicting the normalized rank of a numerical variable. It exploits a probabilistic approach to estimate the distribution of the target rank conditionally to each input variable, then combines the univariate preprocessings using a naive Bayes approach.

Preprocessing. The preprocessing task discretizes both the input and output variables to estimate the conditional distribution of the target variable. This task is turned into a model selection problem. For that, a bivariate partitioning model family is defined, based on discretizing both the input and target variables. The input variable is discretized into intervals (numerical case) or partitioned into groups of values (categorical case), jointly with the discretization of the numerical target variable. Using a Bayes model selection approach, an analytical criterion is derived to select the partition with the highest posterior probability. The optimized criterion is $p(M)p(D|M)$, where $p(M)$ is the prior probability of a preprocessing model and $p(D|M)$ the conditional likelihood of the data given the model.

Univariate evaluation. Taking the negative log of the criterion, $c(M) = -(\log p(M) + \log p(D|M))$, the approach receives a Minimim Description Length (MDL) [5] interpretation, where the objective is to minimize the coding length of the model plus that of the data given the model. The null model $M_\emptyset$ is the preprocessing model with one single interval or group of values, which represents the case with no correlation between the input and output variables. We then introduce the *Level* criterion in Equation 1 to evaluate the univariate importance of a variable.

$$Level = 1 - \frac{c(M)}{c(M_\emptyset)}. \qquad (1)$$

The *Level* grows with the importance of an input variable. It is a between 0 and 1, 0 for irrelevant variables uncorrelated with the target variable.

Naive Bayes regression. The obtained discretisation or value grouping of the input variables resulting from the preprocessing can be used to build univariate regressors and multivariate ones under the naive Bayes assumption. More precisely, the obtained regressors are probabilistic rank regressors: they are able to predict the rank of the numerical target value, as well as the full conditional distribution $p(rank(y)|x)$ of the rank of the target value given the input value.

2.2 Selective Naive Bayes Rank Regression

The Naive Bayes (NB) classifier has proved to be very effective in many real data applications [6, 7]. It is based on the assumption that the variables are independent within each class, and solely relies on the estimation of univariate conditional probabilities. The naive independence assumption can harm the performance when violated. In order to better deal with highly correlated variables, the Selective Naive Bayes approach [8] exploits a wrapper approach [9] to select the subset of variables which optimizes the classification accuracy. Although the Selective Naive Bayes approach performs quite well on datasets with a reasonable number of variables, it does not scale on very large datasets with hundreds of thousands of instances and thousands of variables, such as in marketing applications or text mining. The problem comes both from the search algorithm, whose complexity is quadratic in the number of variables, and from the selection process which is prone to overfitting. In [4], the overfitting problem is tackled by relying on a Bayesian approach, where the best model is found by maximizing the probability of the model given the data. The parameters of a variable selection model are the number of selected variables and the subset of variables. A hierarchic prior is considered, by first choosing the number of selected variables and second choosing the subset of selected variables. The conditional likelihood of the models exploits the Naive Bayes assumption, which directly provides the conditional probability of each output value. This allows an exact calculation of the posterior probability of the models. Efficient search heuristic with super-linear computation time are proposed, on the basis of greedy forward addition and backward elimination of variables. The work described in the case of classification [4] has been applied in the case of regression: instead of predicting a set of classes, the task consists in predicting a set of ordered values. The regressor resulting from the best subset of variables is the MAP (maximum a posteriori) Naive Bayes.

2.3 Automatic Variable Construction

In a data mining project, the data preparation phase aims at constructing a data table for the modeling phase [10, 11]. The data preparation is both time consuming and critical for the quality of the mining results. It mainly consists in the search of an effective data representation, based on variable construction and selection. Variable construction [12] has been less studied than variable selection [13] in the literature. However, learning from relational data has recently received an increasing attention. The term Multi-Relational Data Mining (MRDM) was initially introduced in [14] to address novel knowledge discovery techniques from multiple relational tables. The common point between these techniques is that they need to transform the relational representation. Methods named by propositionalisation [15, 16, 17] try to flatten the relational data by constructing new variables that aggregate the information contained in non target tables in order to obtain a classical tabular format.

In [2], an automatic variable construction method is proposed for supervised learning, in the multi-relational setting using a propositionalisation-based approach. Domain knowledge is specified by describing the structure of data by the means of variables, tables and links across tables, and choosing construction rules. For example, Figure 1 describes the structure of the data for the challenge. The statistical unit is the *Page*, where the target variable is *sum_visits_48h* (resp. *sum_twitter_48h*, *sum_facebook_48h*). The *Series* table contains the secondary records per page, with one record per time of the time series. The construction rules used in the challenge are detailed at the beginning of Section 3.2, with examples of variables that can be constructed by applying these construction rules.

The space of variables that can be constructed is virtually infinite, which raises both combinatorial and over-fitting problems. When the number of original or constructed variables increases, the chance for a variable to be wrongly considered as informative becomes critical. A prior distribution over all the constructed variables is introduced. This provides a Bayesian regularization of the constructed variables, which allows to penalize the most *complex* variables. An effective algorithm is introduced as well to draw samples of constructed variables from this prior distribution. Experiments show that the approach is robust and efficient.

3. CHALLENGE SUBMISSIONS

3.1 Challenge description

The ECML/PKDD 2014 Predictive Web Analytics discovery challenge (ChartBeat challenge) aims at predicting the number of visits and messages on Twitter and Facebook after 48h, given a time series of such 12 observations (collected every 5 minutes) during the first hour after publication of each URL. This data is described using a multi-table representation, as shown in Figure 1. The root table is defined by two identifier variables, two descriptives variables (posted weekday and hour) and a table of 12 observations (series1h) for the number of visits, messages on Twitter and Facebook and average time per visit, every 5 minutes. Three target variables are considered: number of visits, messages on Twitter and Facebook cumulated in the first 48 hours after the publication of the page.

The evaluation criterion is the RMSE (Root Mean Squared Error) in the prediction of the challenge function $log(1 +$

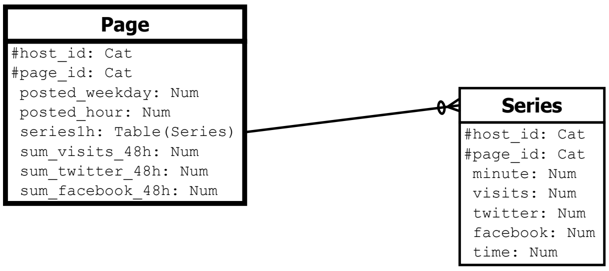

Figure 1: Multi-table representation for the data of the ChartBeat challenge

sum_target_48h) (with $target = visits, twitter, facebook$).

3.2 Submissions

We use the automatic variable construction framework presented in Section 2.3 to create new variables from the observations in the table Series, using the following variable construction rules.

- *Selection(Table, Num)→Table*: selection of records from the table according to a conjunction of selection terms (membership in a numerical interval),
- *Count(Table)→Num*: count of records in a table,
- *Mean(Table, Num)→Num*: mean value,
- *Median(Table, Num)→Num*: median value,
- *Min(Table, Num)→Num*: min value,
- *Max(Table, Num)→Num*: max value,
- *StdDev(Table, Num)→Num*: standard deviation,
- *Sum(Table, Num)→Num*: sum of values.

Using the data structure presented in Figure 1 and the previous construction rules, one can for example construct the following variables ("name" = *formula*: comment) to enrich the description of a *Page*:

- "Sum(series1h.visits)" = *Sum(series1h, visits)*: total number of visits in the first hour,
- "Min(series1h.twitter) where minute <= 32.5" = *Min(Selection(series1h, minute <= 32.5), twitter)*: total number of twits in the first 32 minutes.

3.2.1 First submission

In a first submission, we directly apply the method of Section 2 and generate 1000 variables, evaluate them using bivariate preprocessing and select a subset of variables for the SNB rank regressor. The output of the SNB rank regressor is the full conditional distribution $p(rank(y)|x)$ of the rank of the target value given the input value. For each instance with input data x, we integrate over this distribution to get our prediction $\widehat{y}$, according to $\widehat{y} = \int f(y)p(rank(y)|x)\,\mathrm{d}y$, where $f(y) = \log(1+y)$ with $y = sum\_target\_48h$. Using this first submission, we obtained the following leaderboard scores: 1.124 for the number of visits, 0.686 for the number of Twitter messages and 1.442 for the number of Facebook messages.

3.2.2 Second submission

In a second submission, we change our problem formulation, both in the input data representation and the target variable. We add three ratio variables in the Series, by dividing the number of visits or messages per window of 5 minutes by the total number in the first hour (e.g. $visitsRatio_n = visits_n / \sum_{series1h} visits_i$). We also change the target variable, by exploiting the equation $y = y1 + (y - y1)$ (with $y = sum\_target\_48h$ and $y1 = sum\_target\_1h$). As the first term $y1$ is known from the input data, predicting the second term may improve the final score. We rewrite the objective function

$$\begin{aligned} f(y) &= \log(1+y), \\ &= \log(1+y1)(1+(y-y1)/(1+y1)), \\ &= \log(1+y1)+\log(1+(y-y1)/(1+y1)), \end{aligned}$$

which turns into the new objective function $g(y) = \log(1 + (y-y1)/(1+y1))$. We then use the same method as in the first submission (1000 constructed variables and SNB rank regressor). Using this second submission, we obtained the following leaderboard scores: 1.053 for the visits, 0.675 for Twitter and 1.426 for Facebook.

3.3 Challenge Results

Our second submission was ranked 3$^{\text{rd}}$ on the three tasks, not far from the winner, as shown in Table 1.

It is noteworthy that our results were obtained almost automatically using the method summarized in Section 2. This is illustrated in Figure 2, which shows all results obtained by participants on the leaderboard during the challenge, for the results better than twice the baseline result provided by the organizers. Our two solutions were submitted at the same time on July, 8, 2014, and obtained competitive results w.r.t. the distribution of all results. The other participants submitted more solutions and choose the best one in the end (in this challenge, the leaderboard dataset is the same as the final dataset used to evaluate the final results).

3.4 Evaluation of our Best Submission

One advantage of our method is its robustness and interpretability.

On the train dataset, we obtained the following scores: 1.061 for the visits, 0.673 for Twitter and 1.409 for the Facebook. These scores are very close from the related leaderboard scores obtained on the test set, which shows the robustness of the approach.

We provide below the list of variables selected by the SNB for each of the three challenge tasks, by decreasing order of *Level* (see Equation 1 in Section 2.1).

Table 1: Challenge final results

	Visits		Twitter		Facebook	
1st	Flavio	0.989	Petrichor	0.651	Flavio	1.378
2nd	Petrichor	0.991	Flavio	0.666	Petrichor	1.381
3rd	Marc	1.053	Marc	0.675	Marc	1.426
4th	Joao Palotti	1.068	Shestakoff	0.815	Joao Palotti	1.589
5th	Shestakoff	1.106	Joao Palotti	0.816	Shestakoff	1.744
6th	Sergey Kovalchuk	1.416	Sandeep	0.886	Sandeep	2.230
7th	Sandeep	1.735	Sergey Kovalchuk	1.001	Sergey Kovalchuk	2.542

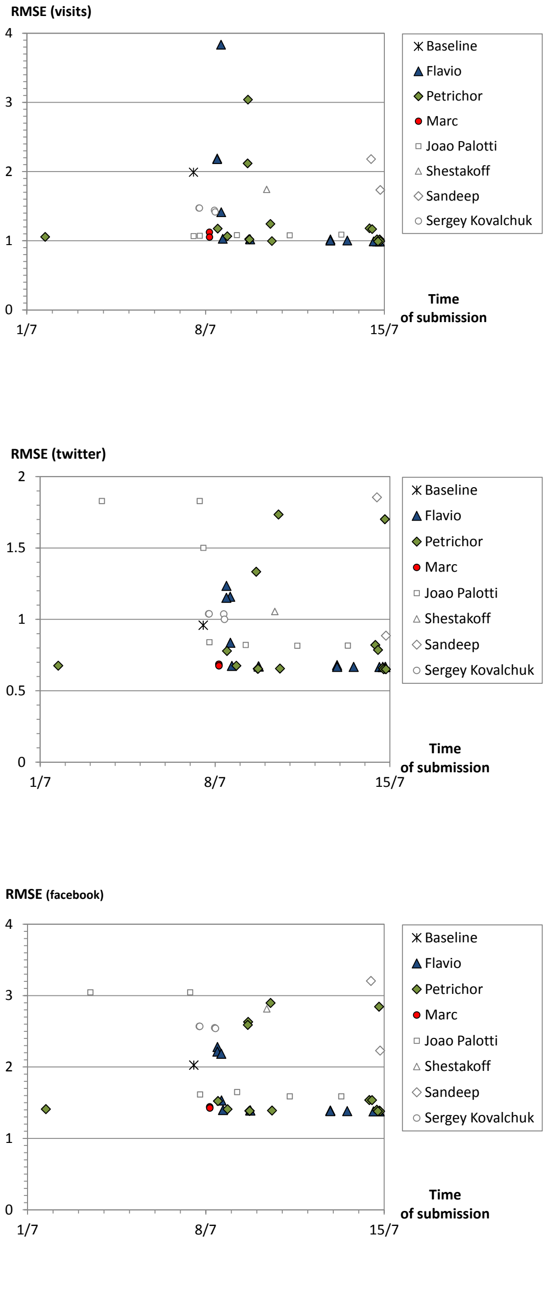

Figure 2: Challenge leaderboard: performance vs submission time

Variables used in the SNB for the visits task:

- 0.02387: Sum(series1h.visits)
- 0.01211: Sum(series1h.visitsRatio) where minute $>$ 32.5
- 0.01041: host_id
- 0.00028: Min(series1h.twitter) where minute $\leq$ 32.5
- 0.00026: posted_hour
- 0.00006: Max(series1h.time) where time $>$ 37.5
- 0.00005: Sum(series1h.twitterRatio) where visits in]0.5, 3.5]

Variables used in the SNB for the Twitter task:

- 0.05324: Sum(series1h.twitter)
- 0.02686: host_id
- 0.00199: Max(series1h.minute) where visitsRatio $\leq$ 0.067
- 0.00053: Max(series1h.time) where time $>$ 37.5
- 0.00050: Min(series1h.facebook) where timeRatio $>$ 0.078
- 0.00005: Min(series1h.facebook) where visits in]3.5, 8.5]

Variables used in the SNB for the Facebook task:

- 0.04482: Sum(series1h.facebook) where facebookRatio $>$ 5.9 10^{-6}
- 0.01314: host_id
- 0.00585: Max(series1h.visitsRatio) where minute $\leq$ 32.5
- 0.00005: Max(series1h.minute) where time in]0.5, 21.5]

Whereas 1000 variables are automatically constructed and preprocessed using bivariate discretization, the SNB regressor selects a small subset of variables, which improves both the intepretability and the deployment time. As expected the sum of the target observations in the first hour (e.g. variable named *Sum(series1h.visits)* in the first task) is the best predictor for each task. Using this best variable alone, we obtained the following scores on the train dataset: 1.185 for the visits, 0.785 for the number of Twitter messages and 1.543 for the number of of Facebook messages. This univariate predictor would have been ranked 4th in the challenge for the Twitter and Facebook tasks. Another important variable for each task is the host. According to the task, this variable (having 100 values) is partitioned into around 25 categories of hosts. These two simple variables are the most important for the three prediction tasks, with levels far beyond that of the other variables. Other more complex variables bring small further improvements, such as *Sum(series1h.visitsRatio) where minute > 32.5*. This automatically constructed variable can easily be interpreted as the proportion of visits in the second half of the first hour after the page is publicated.

4. CONCLUSION

In this paper, we have described our solution for the ECML 2014 Predictive Web Analytics challenge. We have shown that using the Selective Naive Bayes rank regressor together with automatic variable construction allows to quickly and efficiently obtain a competitive solution. Our solution is almost parameter-free. It requires to describe the input data using a multi-table representation and to choose a number of variables to construct. The rest of the process is fully automatic, which is a key feature for a data mining method and allows to face the increasing number of data mining projects. Furthermore, the evaluation of our solution shows good robustness and interpretability. The test performance is very close to the train performance and the trained predictors exploit a very small subset of variables. These automatically constructed variables rely on SQL-like formulas, which provide an easy interpretation. In future work, we plan to extend the variable construction framework by providing additional construction rules with potential specialization per application domain.

5. REFERENCES

[1] C. Castillo and J. Schwartz, “ECML/PKDD predictive challenge dataset,” 2014, provided by Chartbeat, Inc. https://sites.google.com/site/predictivechallenge2014/.

[2] M. Boullé, “Towards automatic feature construction for supervised classication,” in *ECML/PKDD 2014*, 2014, accepted for publication.

[3] C. Hue and M. Boullé, “A new probabilistic approach in rank regression with optimal bayesian partitioning,” *Journal of Machine Learning Research*, pp. 2727–2754, 2007.

[4] M. Boullé, “Compression-based averaging of selective naive Bayes classifiers,” *Journal of Machine Learning Research*, vol. 8, pp. 1659–1685, 2007.

[5] J. Rissanen, “Modeling by shortest data description,” *Automatica*, vol. 14, pp. 465–471, 1978.

[6] P. Langley, W. Iba, and K. Thompson, “An analysis of Bayesian classifiers,” in *10th National Conference on Artificial Intelligence.* AAAI Press, 1992, pp. 223–228.

[7] D. Hand and K. Yu, “Idiot’s bayes ? not so stupid after all?” *International Statistical Review*, vol. 69, no. 3, pp. 385–399, 2001.

[8] P. Langley and S. Sage, “Induction of selective Bayesian classifiers,” in *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence.* Morgan Kaufmann, 1994, pp. 399–406.

[9] R. Kohavi and G. John, “Wrappers for feature selection,” *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[10] D. Pyle, *Data preparation for data mining.* Morgan Kaufmann Publishers, Inc. San Francisco, USA, 1999.

[11] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, “CRISP-DM 1.0 : step-by-step data mining guide,” The CRISP-DM consortium, Tech. Rep., 2000.

[12] H. Liu and H. Motoda, *Feature Extraction, Construction and Selection: A Data Mining Perspective.* Kluwer Academic Publishers, 1998.

[13] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, Eds., *Feature Extraction: Foundations And Applications.* Springer, 2006.

[14] A. J. Knobbe, H. Blockeel, A. Siebes, and D. Van Der Wallen, “Multi-Relational Data Mining,” in *Proceedings of Benelearn ’99*, 1999.

[15] S. Kramer, P. A. Flach, and N. Lavrač, “Propositionalization approaches to relational data mining,” in *Relational data mining*, S. Džeroski and N. Lavrač, Eds. Springer-Verlag, 2001, ch. 11, pp. 262–286.

[16] M.-A. Krogel and S. Wrobel, “Transformation-based learning using multirelational aggregation,” in *ILP.* Springer, 2001, pp. 142–155.

[17] H. Blockeel, L. De Raedt, and J. Ramon, “Top-Down Induction of Clustering Trees,” in *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp. 55–63.