Predicting Dangerous Seismic Events in Coal Mines under Distribution Drift

Marc Boullé
Orange Labs,
2 avenue Pierre Marzin, 22300 Lannion, France
http://www.marc-boulle.fr/
Email: marc.boulle@orange.com

***Abstract*—We describe our submission to the AAIA'16 Data Mining Competition, where the objective is to devise a reliable prediction model for detecting periods of increased seismic activity in coal mines. Our solution exploits a selective naive Bayes classifier, with optimal preprocessing, variable selection and model averaging, together with an automatic variable construction method that builds many variables from time series records. One challenging part of the competition is that the input variables are not independent and identically distributed (i.i.d.) between the train and test datasets, since the train data and test data rely on different coal mines and different times periods. We apply a drift-aware methodology to alleviate this problem, that enabled to get a final score of 0.9246 (team marcb), less than 0.015 from the challenge winner.**

I. Introduction

The AAIA'16 Data Mining Competition[1] is related to a problem of prediction of dangerous seismic events in coal mines. Coal mines are equipped with seismic sensors that register bumps and energy. Sensor readings are available as times series of 24 records, with hourly statistic summaries (such as number of bumps, sum, mean or max of energy...). Train data consists of 79,893 samples from a first period, whereas the test data contains 3,860 samples coming from a second period. In the test data, the time periods do not overlap and are in random order, which is not the case in the train data. In this competition, active participants are rewarded by up to four additional train datasets, provided that they make enough submissions. Altogether, the most active participants can obtain up to a total of 133,151 train samples. The objective is to predict if the total seismic energy perceived with 8 hours after the period covered by a data sample exceeds a warning threshold, and the evaluation criterion is the Area Under the ROC Curve (AUC).

In this paper, we present our submission to the challenge. It exploits a Selective Naive Bayes classifier together with an automatic variable construction method (Section II). A good classifier trained on the train data obtains a disastrous leaderboard score. This is not caused by over-fitting, but by a severe distribution drift between train and test data. We suggest in Section III a methodology to alleviate this problem, and apply it in Section IV to elaborate our submissions to the challenge. Finally, Section V summarizes the paper.

[1]https://knowledgepit.fedcsis.org/contest/view.php?id=112

II. Supervised Classification Framework

We summarize the Selective Naive Bayes (SNB) classifier[2] introduced in [1]. It extends the Naive Bayes classifier owing to an optimal estimation of the class conditional probabilities, a Bayesian variable selection and a Compression-based Model Averaging. We also describe the automatic variable construction framework presented in [2], used to get a tabular representation from times series.

A. Optimal preprocessing

Numerical variables are preprocessed using supervised discretization [3] to evaluate the class conditional probabilities. In the MODL approach [4], the discretization is turned into a model selection problem and solved in a Bayesian way. Using a hierarchical prior distribution on the discretization parameters, the Bayes formula is applicable to derive an exact analytical criterion to evaluate the posterior probability of a discretization model. A 0-1 normalized version of this criterion provides a univariate informativeness evaluation of each input variable. Similarly, categorical variables are preprocessed using supervised value grouping [5].

B. Bayesian Approach for Variable Selection

The naive independence assumption can harm the performance when violated. In [1], the Selective Naive Bayes (SNB) classifier [6] is trained using a Bayesian model selection approach to select the best subset of variables [7]. Efficient search heuristics with super-linear computation time are proposed, on the basis of greedy forward addition and backward elimination of variables.

C. Compression-Based Model Averaging

Instead of taking the best subset of variables, the method introduced in [1] averages all the classifiers resulting from different subsets of variable, using a logarithmic smoothing of the posterior distribution of the trained classifiers. The weighting scheme on the models reduces to a weighting scheme on the variables, and finally results in a single Naive Bayes classifier with weights per variable.

[2]Available as a shareware at http://www.khiops.com

D. Automatic Variable Construction for Multi-Table

Variable construction [8] has been less studied than variable selection in the literature. It is all the more necessary in the case of relational data to obtain a flat input data table with tabular representation. It implies a large amount of work for the data analyst and heavily relies on domain knowledge to construct new potentially informative variables. Learning from relational data has recently received an increasing attention in the literature, since the introduction of Multi-Relational Data Mining (MRDM) in [9], [10]. In this paper, we exploit the automatic variable construction framework presented in [2]. It relies on a formal description of the data structure, with a root table and several secondary tables in 0 to 1 or 0 to n relationship and a set of construction rules (*Count*, *CountDistinct*, *Mode*, *Min*, *Max*, *Mean*, *Median*, *StdDev*, *Sum*, *Selection*). The space of variables that can be constructed is virtually infinite, which raises both combinatorial and over-fitting problems. These problems are solved by introducing a prior distribution over all the constructed variables, as well as an effective algorithm to draw samples of constructed variables from this distribution.

III. A Methodology to Reduce the Drift Problem

Statistical learning relies on identically and independently distributed (i.i.d.) data. Given this assumption, models trained from a train dataset can be deployed on a test dataset, with some guarantees of performance. This i.i.d. assumption does not hold in many real world cases, for example in case of time series data, in the marketing field where a model (churn, fraud, cross-selling...) is trained on a past period and deployed on a future period, ergonomics where a model is trained from a panel of few volunteers... In these cases of *drift* between the train and deployment datasets, as the data are not i.i.d, obtaining good classification performance on the train data does not guarantee good performance on the test data.

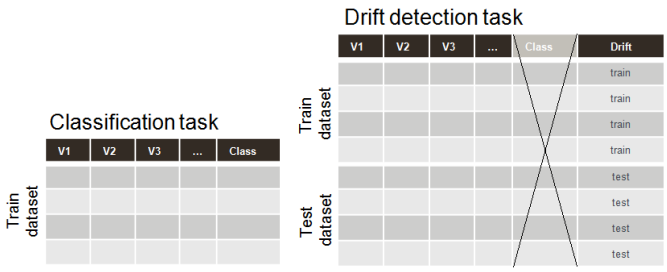

Fig. 1. Classification and drift detection tasks

In [11], [12], we have investigated this issue and proposed a methodology to reduce the drift problem. Let us assume that we have a classification task, a train dataset with class labels and a test dataset that potentially comes from a different distribution. The objective is to train a classifier and to predict the test class labels as accurately as possible whatever be the drift. Let us then consider two tasks: classification and detection of the drift. The drift detection task can be turned into a classification task as in [13], by merging the train and test datasets and using the dataset label ('train' or 'test') as the target variable (see Figure 1).

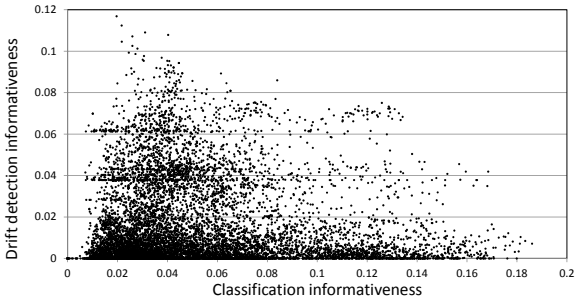

Fig. 2. Informativeness of 10,000 input variables

The initial representation is then evaluated using the pre-processing method summarized in Section II-A, both for the classification and the drift detection tasks. Intuitively, if we are able to select an input representation with good classification performance on the train data but poor drift detection, we expect that our classifier will be less sensitive to drift and its performance drop on the test dataset will be reduced.

The objective is then to explore varying input representations and select the one with the best classification performance together with the poorest drift detection. To illustrate this, we represent in Figure 2 the informativeness of 10,000 input variables for the classification and drift detection tasks (borrowed from the AAIA'2015 challenge [11]). The results show that there are variables with large drift informativeness and small classification informativeness (top-left of the figure), or on the contrary variables with small drift informativeness and large classification informativeness (bottom-right). The interesting variables are those on the right and close to the X axis, with small drift informativeness. Using these information, we can select interesting variables, either automatically (as in the AAIA'2015 challenge [11]) or manually with a focus on interpretability (as in the IJCRS'2015 challenge [12]). With interesting variables only, the classification performance may slightly decrease in the train dataset (because only part of the available variables are exploited), but the performance is likely to be more resilient to drift, with a better performance on the test dataset.

IV. Challenge submissions

A. Applying the Framework for the Challenge

Coal mines are represented using a multi-table schema:

- root table that contains the identifier of the main working site (coal mine) and 12 other characteristics related to the whole period of 24 hours,
- secondary table (0-n) for the time series of 24 hourly summarized seismic sensor readings,
- secondary table (0-1) that contains some meta-data per working site.

Using the data structure presented in Figure 3 and the construction rules introduced in Section II-D, one can for

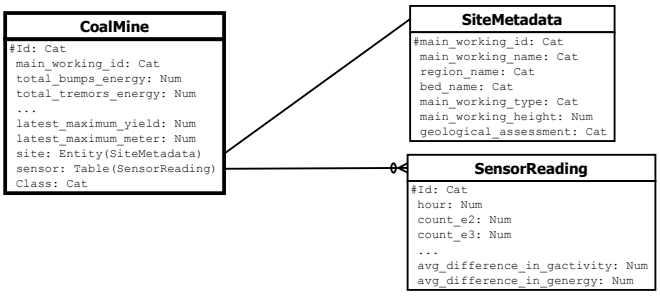

Fig. 3. Multi-table representation for the data of the AAIA'16 challenge

example construct the following variables ("name": comment) to enrich the description of a *CoalMine*:

- "Mean(sensor.sum_e2)": mean of the sensor sum_e2 readings,
- "Count(sensor) where sum_e2 $>$ 0.5": number of sensor readings where the sum_e2 value is greater than 0.5,
- "Max(sensor.sum_e2) where highest_bump_energy $>$ 50": max of the sum_e2 value from sensor readings where highest_bump_energy is greater than 50.

The number of variables to construct is the only user parameter. An input flat data table representation is then obtained from the set of all automatically constructed variables. All these variables are then preprocessed using the optimal discretization method (cf. Section II-A) to assess their informativeness and evaluate their class conditional probabilities, before training the SNB classifier.

For each experiment, 1000 variables are built using the automatic variable construction framework summarized in Section II-D.

B. Preliminary experiments

We first perform some explanatory analysis to better understand the data, without any submission on the leaderboard.

1) Evaluation of the expected performance: Using a 70%-30% train-test split of the train dataset, we obtain a train AUC of 0.99 and a test AUC of 0.97. The performance are both accurate and reliable. However, prediction of increased seismic activity should not be so easy, and we suspect that the good performance might be caused by some bias in the dataset.

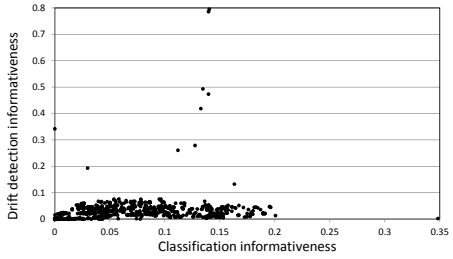

Fig. 4. Informativeness of 1000 input variables for the AAIA'2016 challenge

2) Evaluation of the drift: To evaluate whether the train and test dataset are i.i.d, we apply the drift detection methodology described in Section III. This drift detection task achieves an almost perfect performance with an AUC of 0.995, meaning that the train and test data can be well separated. The most informative variables for the drift detection task are the identifier of the main working site, as well as all the meta-data variables per working site. These drift informative variables are far above most other input variables in Figure 4. As the data are not i.i.d, obtaining good classification performance on the train data does not guarantee good performance on the test data.

3) Distribution of coal mines in train and test datasets: To further investigate on the observed drift, we collect the identifiers of the coal mines in the train and test datasets. Overall, 24 coal mines are used: 7 in the initial train dataset, 16 with all the additional train datasets and 21 in the test dataset. The distribution of the coal mines is heavily unbalanced in the train dataset, whereas is is more balanced in the test dataset.

4) Distribution of the target labels: The target class is heavily unbalanced, with 1171 *warning* (around 1.5%) and the rest as *normal*. Furthermore, 2 among the 7 initial train coal mines are never labeled as *warning*.

According to the challenge organizers, the time periods in the test data do not overlap and are in random order. We then assume that in the train data, the time periods overlap and are in sequential order. This overlapping causes an additional problem of non i.i.d data, with the train data being oversampled compared to the test data.

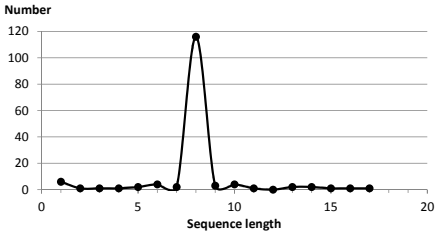

Fig. 5. Number of sequences of train warnings per sequence size

To evaluate the over-sampling factor we collect the sequences of consecutive *warning* in the train dataset. The results, displayed in Figure 5, show that most sequences are of length 8.

C. First submission

The preliminary explanatory analysis summarized in Section IV-B shows that there is a severe drift between the train and test datasets, caused by different mines, different time periods and different sampling rates. To reduce this drift problem, we apply the following protocol:

- remove any variable that identifies the coal mines (*main_working_id* plus the additional meta-data variables per working site),
- re-sample the train mines by keeping at most 5% of instance per mine, so as to get a more balanced distribution as in the test dataset (21 test mines, thus $1/21 \approx 5\%$),
- sub-sample the remaining train instances by a factor of $1/8 \approx 12\%$), so as to get approximatively the same sampling factor as in the test dataset.

This first trial, submitted one day after entering the challenge, obtains a leaderboard AUC of 0.9239, which is very competitive ($\approx 1\%$ from the leader at the submission time).

D. Second and final submission

Although the first obtained results are quite good, they exploit only a small subset of the train instances (around 3% after applying the sampling strategies). To better exploit all the available train data, we repeat the train protocol described previously 100 times (based on different random samples) and average the predictions. This second trial (our final one) obtains almost the same leaderboard AUC (0.9243), but we expect the averaging strategy may lead to more reliable predictions (the leaderboard AUC is evaluated on only 25% of the test data).

Furthermore, this averaging over 100 train samples provides additional insights w.r.t. the variance of the results, which amounts to around 1%. We expect that the variance of leaderboard AUC is still higher and that the best participant submissions (over potentially hundred of submissions) are likely to over-estimate the true test AUC. Thus getting a leaderboard AUC within the variance of the leader leaves few room for further improvement.

E. Additional experiments

First, as a sanity check, we submitted the first prediction obtained using all the available train data (all variables and instances: see Section IV-B). As expected, the drift effect is disastrous, and our 0.97 train AUC dropped down to a 0.60 leaderboard AUC.

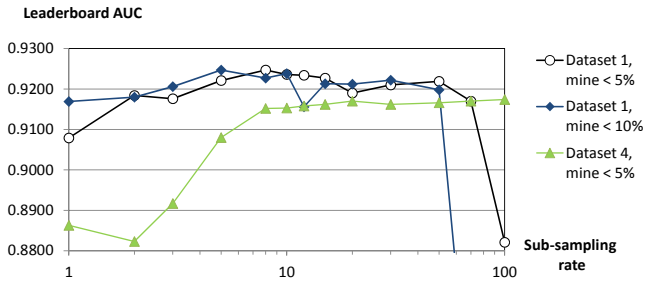

Fig. 6. Leaderboard AUC per sampling rate for three scenarios

We also performed sensitivity analysis, with varying the train mine re-sampling rate, sub-sampling rate, averaging strategy and number of constructed variables. We also experimented with using the additional train datasets, with some manual or automatic feature selection as well as finer anti-drift approaches (see [12]). For example, Figure 6 displays the leaderboard AUC using 1 or all 4 additional train datasets, with train mine resampling rate of at most 5% or 10%, and sub-sampling rate varying from 1% to 100%. This shows that the initial train mine re-sampling rate (5%) looks stable in a wider range (than 10%), that the initial sub-sampling rate (12%) is in a plateau of good performance (between 8% and 20% looks fine). Finally, using all the 4 additional train datasets, the performance was slightly less accurate, but as this is within the expected variance, this is not significant.

Overall, the obtained leaderboard results showed that the preliminary chosen protocol parameters (see Sections IV-B, IV-D) were quite stable, and the results dropped down only for significant changes in the re-sampling and sub-sampling rates (worse performance for at least half or twice the initial value of the parameters). The additional data did not provide any further improvement, but there was not much room for such improvement. In the end, we choose to keep our second submission, that went from a 0.9243 leaderboard AUC to a 0.9246 final AUC.

V. Conclusion

In the AAIA'16 Data Mining Competition, the train and test data are not i.i.d, which causes a dramatic drop of the test performance, even for accurate and reliable trained classifiers. After preliminary explanatory analysis, we identified several causes of drift between the train and test data: different distributions of coal mines, different sampling rate and different period. To be more robust to drift, we proposed a methodology based on removing variables too sensitive to drift, re-sampling to get a more balanced distribution of the train mines and sub-sampling to achieve approximately the same sampling rate. Applying this methodology, 100 classifiers were trained, each exploiting sub-samples of only 3% of the trained instances, and the averaged predictions obtained a 0.9246 final AUC.

References

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

[2] ——, "Towards automatic feature construction for supervised classification," in *ECML/PKDD 2014*. Springer-Verlag, 2014, pp. 181–196.

[3] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proceedings of the 12th International Conference on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 1995, pp. 194–202.

[4] M. Boullé, "MODL: a Bayes optimal discretization method for continuous attributes," *Machine Learning*, vol. 65, no. 1, pp. 131–165, 2006.

[5] ——, "A Bayes optimal approach for partitioning the values of categorical attributes," *Journal of Machine Learning Research*, vol. 6, pp. 1431–1452, 2005.

[6] 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.

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

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

[9] A. J. Knobbe, H. Blockeel, A. Siebes, and D. Van Der Wallen, "Multi-Relational Data Mining," in *Proceedings of Benelearn '99*, 1999.

[10] 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.

[11] M. Boullé, "Tagging fireworkers activities from body sensors under distribution drift," in *Federated Conference on Computer Science and Information Systems*, 2015. doi: 10.15439/2015F423 pp. 389–396.

[12] ——, "Prediction of methane outbreak in coal mines from historical sensor data under distribution drift," in *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing - 15th International Conference, RSFDGrC*, 2015. doi: 10.1007/978-3-319-25783-9 pp. 439–451.

[13] A. Bondu and M. Boullé, "A supervised approach for change detection in data streams," in *Proceedings of International Joint Conference on Neural Networks*, 2011, pp. 519–526.