An Efficient Parameter-Free Method for Large Scale Offline Learning

Keywords: large scale learning, offline learning, naive Bayes, discretization, variable selection, model averaging

Marc Boullé MARC.BOULLE@ORANGE-FTGROUP.COM
France Telecom R&D, 2 avenue Pierre Marzin, 22300 Lannion

Abstract

With the rapid growth of computer storage capacities, available data and demand for scoring models both follow an increasing trend, sharper than that of the processing power. However, the main limitation to a wide spread of data mining solutions is the non-increasing availability of skilled data analysts, which play a key role in data preparation and model selection.

In this paper we present a parameter-free scalable classification method, which is a step towards fully automatic data mining. The method is based on Bayes optimal univariate conditional density estimators, naive Bayes classification enhanced with a Bayesian variable selection scheme, and averaging of models using a logarithmic smoothing of the posterior distribution. We focus on the complexity of the algorithms and show how they can cope with datasets that are far larger than the available central memory. We finally report results on the Large Scale Learning challenge, where our method obtains state of the art performance within practicable computation time.

1. Introduction

Data mining is "the non-trivial process of identifying valid, novel , potentially useful, and ultimately understandable patterns in data" (Fayyad et al., 1996). Several industrial partners have proposed to formalize this process using a methodological guide named CRISP-DM, for CRoss Industry Standard Process for Data Mining (Chapman et al., 2000). The CRISP-DM model provides an overview of the life cycle of a data mining project, which consists in the following phases: business understanding, data understanding, data preparation, modeling, evaluation and deployment. Practical data mining projects involve a large variety of constraints, like scalable training algorithms, understandable models, easy and fast deployment. In a large telecommunication companies like France Telecom, data mining applies to many domains: marketing, text mining, web mining, traffic classification, sociology, ergonomics. The available data is heterogeneous, with numerical and categorical variables, target variables with multiple classes, missing values, noisy unbalanced distributions, and with numbers of instances or variables which can vary over several order of magnitude. The most limiting factor which slows down the spread of data mining solutions is the data preparation phase, which consumes 80% of the process (Pyle, 1999; Mamdouh, 2006) and requires skilled data analysts. In this paper, we present a method[1] which aims at automatizing the data preparation and modeling phases of a data mining project, and which performs well on a large variety of problems.

The paper is organized as follows. Section 2 summarizes our method and Section 3 focuses on its computational complexity. Section 4 reports results obtained on the Large Scale Learning challenge. Finally, Section 5 gives a summary and discusses future work.

2. A Parameter-Free Classifier

Our method, introduced in (Boullé, 2007), 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.

[1] The method is available as a shareware, downloadable at http://perso.rd.francetelecom.fr/boulle/

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2.1. Optimal discretization

The naive Bayes classifier has proved to be very effective on many real data applications (Langley et al., 1992; Hand & Yu, 2001). It is based on the assumption that the variables are independent within each output class, and solely relies on the estimation of univariate conditional probabilities. The evaluation of these probabilities for numerical variables has already been discussed in the literature (Dougherty et al., 1995; Liu et al., 2002). Experiments demonstrate that even a simple equal width discretization brings superior performance compared to the assumption using a Gaussian distribution. In the MODL approach (Boullé, 2006), the discretization is turned into a model selection problem and solved in a Bayesian way. First, a space of discretization models is defined. The parameters of a specific discretization are the number of intervals, the bounds of the intervals and the output frequencies in each interval. Then, a prior distribution is proposed on this model space. This prior exploits the hierarchy of the parameters: the number of intervals is first chosen, then the bounds of the intervals and finally the output frequencies. The choice is uniform at each stage of the hierarchy. Finally, the multinomial distributions of the output values in each interval are assumed to be independent from each other. A Bayesian approach is applied to select the best discretization model, which is found by maximizing the probability $p(Model|Data)$ of the model given the data. Owing to the definition of the model space and its prior distribution, the Bayes formula is applicable to derive an exact analytical criterion to evaluate the posterior probability of a discretization model. Efficient search heuristics allow to find the most probable discretization given the data sample. Extensive comparative experiments report high performance.

The case of categorical variables is treated with the same approach in (Boullé, 2005), using a family of conditional density estimators which partition the input values into groups of values.

2.2. Bayesian Approach for Variable Selection

The naive independence assumption can harm the performance when violated. In order to better deal with highly correlated variables, the selective naive Bayes approach (Langley & Sage, 1994) exploits a wrapper approach (Kohavi & John, 1997) 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 (Boullé, 2007), 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 label. 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.

2.3. Compression-Based Model averaging

Model averaging has been successfully exploited in Bagging (Breiman, 1996) using multiple classifiers trained from re-sampled datasets. In this approach, the averaged classifier uses a voting rule to classify new instances. Unlike this approach, where each classifier has the same weight, the Bayesian Model Averaging (BMA) approach (Hoeting et al., 1999) weights the classifiers according to their posterior probability. In the case of the selective naive Bayes classifier, an inspection of the optimized models reveals that their posterior distribution is so sharply peaked that averaging them according to the BMA approach almost reduces to the maximum a posteriori (MAP) model. In this situation, averaging is useless. In order to find a trade-off between equal weights as in bagging and extremely unbalanced weights as in the BMA approach, a logarithmic smoothing of the posterior distribution, called compression-based model averaging (CMA), is introduced in (Boullé, 2007). Extensive experiments demonstrate that the resulting compression-based model averaging scheme clearly outperforms the Bayesian model averaging scheme.

3. Complexity Analysis

In this section, we first recall the algorithmic complexity of the algorithms detailed in (Boullé, 2006; Boullé, 2007) in the case where all the data fit in central memory, then introduce the extension of the algorithms when data exceed the central memory.

The algorithm consists in three phase: data prepro-

cessing using discretization or value grouping, variable selection and model averaging. The preprocessing phase is super-linear in time and requires $O(KN \log N)$ time, where K is the number of variables and N the number of instances. In the variable selection algorithm, the method alternates fast forward and backward variable selection steps based on randomized reorderings of the variables, and repeats the process several times in order to better explore the search space and reduce the variance caused by the dependence over the order of the variables. The number of repeats is fixed to $\log N + \log K$, so that the overall time complexity of this phase is $O(KN(\log K + \log N))$, which is comparable to that of the preprocessing phase. The model averaging algorithm consists in collecting all the models evaluated in the variable selection phase and averaging then according to a logarithmic smoothing of their posterior probability, with no overhead on the time complexity. Overall, the train algorithm has a $O(KN(\log K + \log N))$ time complexity and $O(KN)$ space complexity.

When Data Exceeds Central Memory. With the $O(KN)$ space complexity, large datasets cannot fit into central memory and training time becomes impracticable as soon as memory pages have to be swapped between disk and central memory[2]. To avoid this strong limitation, we enhance our algorithms with a specially designed chunking strategy. First of all, let us consider the access time from central memory (t_1) and from sequential (t_2) or random (t_3) disk access. In modern personal computers (year 2008), t_1 is in the order of 10 nanoseconds. Sequential disk access are so fast (based on 100 Mb/s transfer rates) that t_2 is in the same order as t_1: CPU is sometimes a limiting factor, when parsing operations are involved. Random disk access t_3 is in the order of 10 milliseconds, one million times slower than t_1 or t_2. Therefore, the only way to manage very large amounts of memory space is to exploit disk in a sequential manner.

Let $S{=}(KN)$ be the size of the dataset and M the size of the central memory. For the preprocessing phase of our method, each variable is analyzed once after being loaded into central memory. We partition the set of input variables into C chunks of K_C variables, such that $K_C N < M$ and $C > S/M$. The preprocessing phase loops on the C subsets, and at each step of the loop, read the dataset, parse and load the chunk variables only, preprocess them to build conditional probability tables and unload the chunk. In the variable selection phase, the algorithm first replaces each input value by its index in the preprocessed conditional probability table, and creates as many preprocessed chunk files as necessary. The variable selection algorithm then loops on the preprocessed chunks in random order: one single chunk is loaded in memory at a time.

[2]On 32 bits CPU, central memory is physically limited to 4 Go, or even to 2 Go on Windows PCs

Table 1. Challenge datasets and validation results of our method for the aoPRC criterion.

Dataset	Training	Dimensions	aoPRC
alpha	500000	500	0.2536
beta	500000	500	0.4616
gamma	500000	500	0.0112
delta	500000	500	0.0818
epsilon	500000	2000	0.0663
zeta	500000	2000	0.034
fd	5469800	900	0.2314
ocr	3500000	1156	0.1564
dna	50000000	200	0.8612
webspam	350000	variable	0.0033

4. Results on the Large Scale Learning Challenge

The Pascal Large Scale Learning Challenge[3] is designed to enable a direct comparison of learning methods given limited resource. The datasets, summarized in Table 1 represent a variety of situation, from artificial datasets (alpha to zeta), face detection (fd), character recognition (ocr), dna split point prediction and webspam detection. They contain up to millions of instances, thousands of variables and tens of gigabytes of disk space.

Although it is not designed to compete with online learning methods from a training time point of view, the challenge datasets are a convenient way to evaluate the scalability of our offline method, when dataset size is larger than the central memory by one order of magnitude (on computers with 32 bit CPU). We made one fully automatic submission, using the raw representation of the dataset for all the datasets. For the two image-based datasets (fd and ocr), we also studied the impact of initial representation using centered reduced rows, and finally chose the raw representation for ocr and centered-reduced representation for fd.

Table 1 reports our accuracy results in the challenge for the area over the precision recall curve (aoPRC) criterion. Except for the alpha dataset, these results always rank among the first competitors, which is a remarkable performance for a fully automatic method,

[3]Web site: http://largescale.first.fraunhofer.de/about/

which exploits the limited naive Bayes assumption. It is noteworthy that when the datasets size if far beyond the size of the central memory, the overhead in wall-clock time (whole load time plus training time) is by only a factor two.

As anticipated, our offline training time is far longer than that of the online methods of the other competitors, by about two orders of magnitude. However, when the overall process is accounted for, with data preparation time and data loading time, our training time become competitive, with about one hour of training time per analyzed gigabyte. Overall, our method is highly scalable and obtains competitive performance fully automatically, without tuning any parameter.

5. Conclusion

We have presented a parameter-free classification method that exploits the naive Bayes assumption. It estimates the univariate conditional probabilities using the MODL method, with Bayes optimal discretization and value groupings for numerical and categorical variables. It searches for a subset of variables consistent with the naive Bayes assumption, using an evaluation based on a Bayesian model selection approach and efficient add-drop greedy heuristics. Finally, it combines all the evaluated models using a compression-based averaging schema.

Our classifier is aimed at automatically producing competitive predictions in a large variety of data mining contexts. Our results in the Large Scale Learning Challenge demonstrates that our method is highly scalable and automatically builds state-of-the art classifiers. When the dataset size is larger than the available central memory by one order of magnitude, our method exploits an efficient chunking strategy, with a time overhead of only a factor two. In future work, we plan to further improve the method and extend it to classification with large number of class values and to regression.

Acknowledgments

I am grateful to Bruno Guerraz, who got the initial Windows code to work under Linux. I would also like to thank the organizers of the Large Scale Learning Challenge for their valuable initiative.

References

Boullé, M. (2005). A Bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research*, *6*, 1431–1452.

Boullé, M. (2006). MODL: a Bayes optimal discretization method for continuous attributes. *Machine Learning*, *65*, 131–165.

Boullé, M. (2007). Compression-based averaging of selective naive Bayes classifiers. *Journal of Machine Learning Research*, *8*, 1659–1685.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0 : step-by-step data mining guide*.

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

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). Knowledge discovery and data mining: towards a unifying framework. *KDD* (pp. 82–88).

Hand, D., & Yu, K. (2001). Idiot bayes ? not so stupid after all? *International Statistical Review*, *69*, 385–399.

Hoeting, J., Madigan, D., Raftery, A., & Volinsky, C. (1999). Bayesian model averaging: A tutorial. *Statistical Science*, *14*, 382–417.

Kohavi, R., & John, G. (1997). Wrappers for feature selection. *Artificial Intelligence*, *97*, 273–324.

Langley, P., Iba, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. *10th national conference on Artificial Intelligence* (pp. 223–228). AAAI Press.

Langley, P., & Sage, S. (1994). Induction of selective Bayesian classifiers. *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence* (pp. 399–406). Morgan Kaufmann.

Liu, H., Hussain, F., Tan, C., & Dash, M. (2002). Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, *4*, 393–423.

Mamdouh, R. (2006). *Data preparation for data mining using SAS*. Morgan Kaufmann Publishers.

Pyle, D. (1999). *Data preparation for data mining*. Morgan Kaufmann Publishers, Inc. San Francisco, USA.