Generic vs. Domain-oriented Evaluation Methods for Fielded Applications

Chunsheng Yang Chunsheng.Yang@nrc.gc.ca
National Research Council Canada, Ottawa, Ontario K1W 1C7, Canada
Yubin Yang yangyubin@nju.edu.cn
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China

Abstract

In the past decade, machine learning algorithms have been widely applied to various real-world applications such as diagnostics, prognostics and bioinformatics. In developing high-performance classifiers for such problems, evaluation of classifiers remains an important challenge facing researchers. This paper addresses this issue from the viewpoint of fielded classifier evaluation. Generic methods, such as classic accuracy, a paired t-test, ROC and AUC, cannot fully meet its needs. The paper first reviews generic methods. It summarizes the criticisms from the machine learning community and analyzes the limitations or deficiencies with respect to fielded applications. It also surveys some emerging domain-oriented approaches which take the specificities of domain problems into consideration. We argue that classifiers have to be evaluated carefully not only using generic methods but also domain-oriented ones in order to promote the acceptance of classifiers in real-world applications.

1. Introduction

Classifier evaluation, or the evaluation of machine learning algorithms, is an important exercise, particularly, in developing classifiers for real-world applications. Evaluation not only helps practitioners compute the performance of a classifier but also helps end-users assess the usefulness of a classifier for a particular domain problem. The ideal way to evaluate classifiers is to perform field trials in real-world environments. This is, however, not realistic because of cost. The traditional way to address this problem is to conduct statistical-based evaluation using generic methods such as accuracy calculation, paired t-test, cross-validation, and bootstrapping. Over the last decade, some newly developed methods have gradually replaced the traditional methods. They include ROC (Receiver Operating Characteristics) analysis [3] and ROC-based approaches such as ROCCH (ROC Convex Hull) [4, 5], AUC (Area Under the ROC Curve) [6,7]. The DEA (Data Envelopment Analysis) [9, 10] approach is also gaining popularity in the multi-class problem. These approaches improve the evaluation of classifiers for general purposes.

However, they fail to address specificities of fielded applications, and thus cannot satisfy the needs of classifier evaluation for real-world applications. Different domain problems demand a tailored approach to evaluate the performance of a classifier. For example, from the perspective of prognostic[1] classifier evaluation, the approach has to take the time to failure and failure coverage into consideration [11]. Unfortunately, none of the existing generic methods do. Even through our experience lies in prognostic applications, we believe that the arguments from other fielded applications such as bioinformatics are similar. Bioinformatics may have different requirements or specificities for classifier evaluation. In order to improve over generic methods for evaluating classifiers researchers have been focusing on developing domain-oriented approaches, to address issues and overcome limitations with generic methods [11, 12]

This paper briefly reviews the generic methods and their deficiencies, and surveys some domain-oriented approaches developed for real-world applications. These domain-oriented approaches not only help developers to evaluate performance of classifiers, but also help operators to identify the usefulness, or business value, which will be achieved if a classifier is deployed. They promote machine leaning techniques to be more widely applied to real-world applications. From our own experience, we argue that classifier evaluation strategy

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

[1] Data-driven prognostics [15] is an emerging application of machine learning or data mining to real-world problems. In data-driven prognostics, the main task is to develop the predictive models from large-sized database by using techniques from machine learning. The prognostic model (or called prognostic classifier) is able to predict the likelihood of a failure with a precise time-to-failure prediction in the prognostic systems.

has to take the specificities of fielded applications into consideration. We contend that a classifier has to be carefully and effectively evaluated not only using generic methods, but also domain-oriented approaches.

The following Section reviews issues with generic methods from the viewpoint of fielded applications, for simplicity, we focus on prognostic applications; Section 3 surveys domain-oriented approaches that have been developed recently; the final Section presents our view on classifier evaluation strategy from the viewpoint of fielded applications.

2. Issues with Generic Methods

Over the past decades, in machine learning research, generic methods, either traditional metrics such as accuracy or error-rate, or recent approaches such as ROC and AUC, have been widely used by a majority of researchers. There is no doubt that these methods play an important role in evaluating the performance of a classifier. However, some researchers [1, 2] have recently started to question the effectiveness of these generic methods. The issues with generic methods in classifier evaluation [1] can be summarized from three perspectives: evaluation metrics, sampling approaches, and interpretation of evaluation results.

Evaluation metrics have many shortcomings themselves. For example, the widely-used accuracy does not account the cost of misclassification. This is a serious problem because some errors cost more than others in different problems. Even though ROC or AUC can help overcome this shortcoming, they are sometimes carelessly used in evaluation. As the author in [1] points out, the results obtained on different datasets (or domains) are averaged for each classifier. This may not work sometimes because the values on different datasets may have different meanings. For example, when a new algorithm is published, the results are most likely compared to old ones by averaging the metrics obtained on multiple datasets such as ones from the UCI Repository.

On the side of sampling approaches, a generic method, such as statistical-based approach, requests iid sampling from a normal distribution. This is very dangerous in practice because some data may be dependent on others. For example, data in time-series depend on each other. Random sampling will separate dependent data into different groups. In practical problems, the distribution is not always a normal distribution.

On the side of interpretation of evaluation results, generic metrics cannot tell operators meaningful information on a classifier. In other words, interpretation of evaluating results is hard to be understood, even misleading. For example, AUC is developed for addressing the problem of ROC by normalizing the value between 0 and 1. Theoretically, the higher the AUC value, the better the performance of a classifier. Therefore, a classifier with 0.8 of AUC value should be better than one which has 0.75 of AUC value. However, such interpretation may be meaningless or useless for an end user from fielded applications. From the viewpoint of business value that a classifier produces if it is deployed, the interpretation may be totally different.

From the perspective of fielded applications, the most important concern is that existing generic methods do not take the specificities or settings of applications into consideration. Taking prognostics as an example, those methods failed to capture two important aspects for prognostic applications. The first aspect is the time to failure (or called remaining useful life of a component). A classifier that predicts a failure too early leads to non-optimal component use. On the other hand, if the failure prediction is too close to the actual failure then it becomes difficult to schedule an action for the maintenance. The second aspect relates to coverage of potential failures. Ideally, a prognostic classifier generates at least one alert for all failures instead of many alerts for just a few failures. That is, the failure coverage is very important when we evaluate performance of classifiers.

Certainly, specific consideration for other applications such as bioinformatics will be different. Therefore, domain-oriented approaches are critical for classifier evaluation.

3. Domain-oriented Approaches

Due to the problems of generic methods in fielded applications, researchers from various areas started to look into domain-oriented approaches for evaluating performance of classifiers. Domain-oriented approaches take specificities of fielded applications into consideration. For example, in the area of text mining, Precision, Recall and the F-measure were often used; in the area of diagnostics, cost curve [8] is widely used to evaluate performance of classifiers. Cost curve transforms a point in ROC which represents a classifier into a cost line by normalizing the error rate and failure rate into cost. Cost curve virtualizes performance of a classifier. Such cost information is much clearer than a coordinate point in ROC. People have used it to evaluate diagnostic classifiers in maintenance domain. As discussed in the later section, cost curve has been used to estimate a range of cost saving for prognostic classifiers as well.

The most successful development of domain-oriented evaluation approaches lies in the area of prognostics. To address the issues described above, two main approaches, score-based and cost-based approaches [11, 12, 13, 14], have been developed and are often used in evaluating prognostic classifiers. The following is the brief review of these two approaches.

Score-based approach: This method takes specificities of prognostics into the evaluation metric [13, 14]. Basically, it defines a reward function for positive prediction (or so-called alerts) from each prognostic classifier. The reward function determined a score based on the timeline of each positive prediction. In prognostics, it is desirable that a positive prediction should be generated in a target time window, which is determined with the requirements of applications. Therefore, a positive prediction within the target window will be rewarded as a positive value; otherwise, it will be punished by assigning a negative value. On the other hand, score-based approach also incorporates the problem coverage into its score formula. In the score-based metric, the accumulated score for all positive predictions will be multiplied by a factor which is the number of predicted failure over total number of failures in the testing dataset. This method has been successfully applied to evaluate classifiers for component prognostics in a complex system such as train wheel prognostics and aircraft engine prognostics.

Cost-based approach: Although the score-based approach proposed above takes the time to failure prediction and problem detection coverage into account for evaluating prognostic classifiers, the scores computed do not inform the end user on the expected cost savings of the classifiers. In real-world applications, the best way to promote machine learning algorithms is to estimate cost savings that will be achieved if a classifier is deployed. To this end, the authors in [11, 12] developed a cost-based method for prognostic classifier evaluation. The goal of this method is to estimate the cost savings for a deployed classifier in a fielded application.

Estimating cost saving is a challenging task. It fully depends on the real cost information from applications. In particular, the cost may be changed from time to time and deployment environments may be changed as well. Therefore, two different metrics for estimating the cost savings are proposed: one for accurate cost information [11], another one for uncertain or missed cost information [12].

When we are able to obtain the accurate cost information from the end user, we can use cost-saving metric in [11] to estimate the business value. By using this metric, four kinds of cost information are requested: the cost of a false alert (an inspection without component replacement), a pro rata cost for early replacement, the cost for fixing a faulty component, and the cost of an undetected failure (i.e., a functional failure during operation without any prior prediction from the prognostic model). The first three costs are generally easy to obtain while the last one is difficult to approximate accurately. This is because failures during operations may incur various other costs that are themselves difficult to estimate.

However, the real cost will change from time to time or is missed in many applications. For example, the cost for fixing a failure changes from time to time. Also, the requested cost information above is not always easy to obtain. This may occur because no one has all this information readily at hand. It may also be due to the fact that people are often reluctant to count the costs associated with safety issues. Even in such tough case, we are still able to estimate a range of the cost savings by using the approach proposed in [12]. This method is called reverse-engineering cost. The idea is to apply the cost curve to visualize the cost saving range for uncertain cost information. For example, when the cost of missed failure ranges from X to Y, and other cost information is exact, the cost curve will show a potential range of cost savings to the end users. Such estimation for cost savings is still useful for evaluating prognostic classifiers.

4. Remarks

Even through generic methods have limitations for classifier evaluation, particularly, for fielded classifier evaluation, they are improved or incorporated to domain-oriented approaches. The most domain-oriented approaches are evolved from generic methods. We believe that generic methods, as statistical-based approaches, are still and will be used for classifier evaluation in the machine learning community. These generic methods will be improved through different processes. For instances, N. Japkowicz suggested two useful processes to improve generic methods: better education and better division between exploratory and evaluation [1].

We also believe that more and more feasible and effective domain-oriented approaches will be developed to meet the needs of real-world applications. This is a right way to promote machine learning algorithms in solving real-world problems. Domain-oriented approaches overcome the limitations of generic methods and incorporate specificities of a domain problem into the evaluation metrics; therefore, they are widely used in real-world applications. At the same time, they are also welcome by end users. However, we can't expect to apply a domain-oriented approach to different domains. This should never happen. Otherwise, we will go back to way, a way generic method goes.

We strongly argue that a classifier has to be evaluated not only using generic methods but also domain-oriented approaches. Generic method could help developers identify the performance of a classifier from the statistical viewpoint at the beginning stage of classifier development. Domain-oriented approaches will be used to evaluate the usefulness and business value more clearly in a simple way, which in turn will promote machine learning techniques in real-world applications.

It has been a long time that we have dreamed to have a simple way to validate the classifiers not only for researchers but also for end users. We believe the cost-based approach [11, 12] will make our dream to be true. This should become an example for other fielded applications.

Acknowledgments

Special thanks go to Sylvain Letourneau and Chris Drummond, for their support, valuable discussions. We also would like to thank the organizers of the ICML 09 workshop on Evaluation Methods for Machine Learning for providing a forum to discuss this important issue. We thank the reviewers for their valuable comments and suggestions.

References

[1] N. Japkowicz, "Classifier Evaluation: A Need for Better Education and Restructuring", ICML 2008 Workshop on Evaluation Methods for Machine Learning, Helsinki, Finland, 2008

[2] J. Demsar, "On the Appropriateness of Statistical Tests in Machine Learning", ICML 2008 Workshop on Evaluation Methods for Machine Learning, Helsinki, Finland, 2008

[3] J. Egan, "Signal Detection Theory and ROC Analysis", New York Academic Press, 1975

[4] Foster Provost and Tom Fawcett, "Analysis and Visualization of Classifier Performance: Combination under Imprecise Class and Cost Distributions", Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD1997), 1997

[5] Foster Provost and Tom Fawcett, "Robust Classification for Imprecise Environment", Machine Learning, 42, pp. 203-231

[6] A.P. Bradley, "The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms", Pattern Recognition, Vol. 30, pp. 1145-1159, 1997

[7] Jin Huang, and Charles X. Ling, "Using AUC and Accuracy in Evaluating Learning Algorithms", IEEE Transactions on Knowledge and Data Engineering, Vol. 17, NO. 3, March 2005, pp. 299-310

[8] C. Drummond, and R. Holte, "Explicitly Representing Expected Cost: An Alternative to ROC Representation", Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2000), New York, 2000, 198-207

[9] R.D. Banker, A. Chanes, and W.W. Cooper, "Some Models for Estimating Technical and Scale Inefficiencies in Data Envelopment Analysis", Management Science, Vol. 30 No. 9, pp. 1078-1092, 1984

[10] Z. Zheng, B. Padmanabhan and H. Zheng, "A DEA Approach for Model Combination", Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2004), Seattle, Washington, USA, 2004, 755-758

[11] C. Yang and S. Létourneau, "Model Evaluation for Prognostics: Estimating Cost Saving for the End Users", The Proceedings of the 6th International Conference on Machine Learning and Applications (ICMLA 2007), Cincinnati, OH, USA, December, 2007

[12] C. Drummond and C. Yang, "Reverse-Engineering Costs: How much will a Prognostic Algorithm save?", Proceedings of the 1st International Conference on Prognostics and Health Management. October 2008, Denver, USA

[13] C. Yang and S. Létourneau, "Learning to Predict Train Wheel Failures", in Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2005), Chicago, USA, August, 2005, 516-525

[14] S. Létourneau, F. Famili, and S. Matwin, "Data Mining for Prediction of Aircraft Component Replacement", IEEE Intelligent Systems Journal, Special Issue on Data Mining. December 1999. 59-6

[15] M. Schwabacher and K. Goebel, "A Survey of Artificial Intelligence for Prognostics", The 2007 AAAI Fall Symposium, Arlington, Virginal, USA, Nov., 2007