Towards Application-specific Evaluation Metrics

Niklas Lavesson* and Paul Davidsson
Blekinge Institute of Technology, Box 520, SE-372 25 Ronneby, Sweden
{Niklas.Lavesson,Paul.Davidsson}@bth.se

Abstract

Classifier evaluation has historically been conducted by estimating predictive accuracy via cross-validation tests or similar methods. More recently, ROC analysis has been shown to be a good alternative. However, the characteristics vary greatly between problem domains and it has been shown that some evaluation metrics are more appropriate than others in certain cases. We argue that different problems have different requirements and should therefore make use of evaluation metrics that correspond to the relevant requirements. For this purpose, we motivate the need for generic multi-criteria evaluation methods, i.e., methods that dictate how to integrate metrics but not which metrics to integrate. We present such a generic evaluation method and discuss how to select metrics on the basis of the application at hand.

1. Introduction

We consider the supervised concept learning problem, i.e., when a learning algorithm, given a set of training instances, should generate a classifier that can be used to predict the class of new instances of the same kind. Recognizing that there is a need for a structured way of evaluating candidates for this problem and that there is no metric so versatile and general that it can be used for a particularly broad range of applications, we have proposed a generic multi-criteria evaluation method that can be customized for the application at hand (Lavesson & Davidsson, 2008). This paper further elaborates on the motivation for, and intended use, of this method but the main purpose here is to suggest how to select which metrics to use when applying the evaluation method for a particular problem.

In Section 2 we motivate the need for generic multi-criteria evaluation methods. We then present related work in Section 3 and describe our generic multi-criteria evaluation method along with relevant definitions in Section 4. In Section 5 we then suggest an approach for selecting metrics. Finally, we summarize and discuss future work in the last section.

2. Motivation for Generic Evaluation Methods

Evaluation is crucial (Witten & Frank, 2005); both with respect to determining the viability of, and for the purpose of selection between, candidates. Historically, the most frequently applied evaluation method has been to estimate the predictive accuracy, which can be defined as the ratio between the number of correctly classified instances and the total number of classified instances from a known data set. Predictive accuracy has traditionally been estimated using statistical methods like cross-validation (Stone, 1974) which work by systematically partitioning the known data into training and testing sets, performing evaluation, and then repeating the process in order to use the maximum amount of data for both training and testing. However, serious concerns have been raised against the validity of such accuracy estimations. For instance, that they assume equal class distribution and misclassification cost which are unlikely to be true in real-world data sets (Provost et al., 1998). Additionally, if the goal is to discover new knowledge to be used in human decision making there are factors to take into account like, e.g., comprehensibility and interestingness (Freitas, 2006). Alternatives to the use of predictive accuracy as the only evaluation metric have been proposed, perhaps most notably ROC analysis (Provost et al., ibid.). However, it is argued theoretically and shown empirically that some evaluation metrics are more appropriate than others for certain problems (Caruana & Niculescu-Mizil, 2006). This argument implies that, analogously to the no-free-lunch theorems (Wolpert, 1995) for supervised learning, which stipulates that no algorithm is superior on all problems, there is no single criterion that is always superior for evaluation. As a consequence, we argue that evaluation criteria and metrics need to be selected on the basis of the requirements of the problem at hand. Depending on the application, the process could also include the evaluation of more than on metric. We believe that there is a fundamental problem with some of the earlier proposed multi-criteria methods in that they are usually based on a static selection of metrics. We argue that there are several benefits from using a generic multi-criteria evaluation approach, i.e., the application of a method that dictates how to integrate metrics but does not specify which metrics. This approach

* corresponding author

lets the user decide which metrics to include on the basis of application requirements. It could be argued that there is no need for a generic method for integrating metrics and that the integration could instead be tailor-made for the application. However, we argue that the benefits from using a generic method is that different instances of such a method (using different metrics and configuration) could be more easily compared and refined across research studies. The use of a generic method takes the focus away from integration issues and puts it on application-critical requirements. The use a generic evaluation method also simplifies the development and use of, what we denote metric-based learning algorithms, i.e., algorithms that take a metric as input and try to generate a classifier with optimal performance with respect to this metric. A metric-based algorithm that is based on a generic evaluation metric can easily be tailored for a particular application just by selecting relevant metrics and configuring metric weights and acceptable ranges.

3. Related Work

Multi-criteria evaluation and analysis has been increasingly applied to topics such as artificial intelligence, machine learning, and decision support systems (Urli & Nadeau, 1999). Notable early attempts include the use of multiple criteria to define fitness functions for evolutionary computing problems, cf., Deb (2001). Performance metrics have been studied extensively by researchers using empirical experiments, cf. Caruana & Niculescu-Mizil (2006) and Huang & Ling (2006), however not with any particular emphasis on multi-criteria evaluation. A seminal machine learning study on multi-criteria evaluation presented a multi-criteria metric based on data envelopment analysis and mathematical programming (Nakhaeizadeh & Schnabl, 1997). Although, the most common application of evaluation metrics is to assess the performance of generated classifiers, some recent studies also show that it is fruitful to use evaluation metrics as a replacement for the inherent metric or bias of popular algorithms. For instance, several studies address the question of how to optimize learning algorithms toward different objectives. We distinguish between three existing approaches, where the first focuses on improving accuracy. For instance, it has been shown that the convergence of a back-propagation-based neural network can be improved by replacing the error function with a log-likelihood cost function (Holt & Semnani, 1990). The second approach is to try to optimize a metric other than accuracy. Notable examples of such studies include the optimization of ROC using decision trees (Ferri et al., 2002) and gradient-descent (Herschtal & Raskutti, 2004), as well as optimization of the f-measure using support vector machines (Musicant et al., 2003). The third approach aims to optimize more than one metric, either by replacing the inherent metric of an existing algorithm with a multi-criteria metric, or by developing a new algorithm that optimizes such a metric. For example, the support vector machines algorithm has been generalized to optimize multi-criteria non-linear performance metrics (Joachims, 2005), and dynamic bias selection has been implemented for prediction rule discovery (Suzuki & Ohno, 1999). Additionally, one study presents an approach called measure-based evaluation (Andersson et al., 1999) and describes how to implement hill-climbing learning algorithms that optimize a multi-criteria metric, called the measure function. Finally, quality attributes and metrics have been applied in empirical machine learning experiments by Lavesson & Davidsson (2006) and Freitas (2006) discusses quality attributes like interestingness and comprehensibility in the knowledge-based engineering context. Elazmeh et al. (2006), Salzberg (1997), and Dietterich (1996) discuss machine learning evaluation procedures but are more focused towards the proper use of statistical tests and experimental procedure while this paper is more concerned with the selection and integration of multiple metrics on the basis of application requirements.

4. A Generic Evaluation Method

In previous work (Lavesson & Davidsson, 2008) we identified a number of attractive properties of possible multi-criteria evaluation methods and presented a generic multi-criteria method designed with these properties in mind. The main purpose of this method, which is called the candidate evaluation function (CEF), is the integration of an arbitrary number of existing metrics in order to get a single scalar result. Additionally, CEF normalizes each included metric, on the basis of its application-dependent acceptable range, in order to get a uniform output domain. It is also possible for the user to specify explicit weights for each metric to ensure that trade-offs important for the application at hand can be properly represented. CEF itself does not dictate which metrics should be used; it merely dictates how metrics are combined. Let c be a candidate classifier and D a data set. We then define m_i as a metric with index i from an index set, I, over the selected set of metrics. CEF is defined as follows:

$$\mathrm{CEF}(c, D, I) = \begin{cases} 0 : \exists i(\overline{m}_i(c, D) < 0) \\ \sum_{i \in I} w_i \overline{m}_i(c, D) \text{ otherwise} \end{cases}$$
$$\text{where} \sum_{i \in I} w_i = 1 \text{ and} \qquad (1)$$
$$\overline{m}_i(c, D) = \begin{cases} 1 : m_i(c, D) > b_i^h \\ \dfrac{m_i - b_i^l}{b_i^h - b_i^l} \text{ otherwise} \end{cases}$$

As can be viewed in Equation 1, it is possible to set an acceptable range for each metric using b_i^l as the lower bound and b_i^h as the higher bound. This makes it possible to completely disqualify classifiers with unacceptable

behavior. We recognize the need to empirically validate CEF but argue that the definition above is sufficient for demonstrating some of the more important concepts of generic evaluation methods.

5. Metric Selection

It is pretty straight-forward to use the method documented in the last section for evaluation. However, one is still faced with the problem of selecting relevant metrics and specifying important trade-offs. The tradition has been to evaluate classifiers on the basis of their predictive accuracy. However, as already noted studies such as Freitas (2006) has pointed out the importance of evaluating other metrics such as interestingness and comprehensibility. Additionally, there are often space and time requirements that prohibit the use of many popular techniques in real-world applications, e.g., where test sets are extremely large or where storage space or computational power is severely limited (Bûcila et al., 2006). Since such applications may also be dependent on high generalization performance, one is faced with the difficult task of meeting several evaluation criteria. We have previously introduced the software engineering concepts of quality attributes (QAs) and quality metrics (QMs) in the context of supervised concept learning (Lavesson & Davidsson, 2006). We showed empirically how to use the higher abstraction layer of QAs to simplify the selection of metrics by distinguishing between which qualities need to be evaluated and finding out how to actually quantify the degree of fulfillment of each quality instead of, e.g., getting stuck in discussions about whether to use method *X* or *Y* for accuracy estimation. We have no intention to diminish the importance of such comparisons. However, we believe that to successfully implement a learning component in a real-world application, more care has to be taken to assess the requirements of the application at hand to decide which specific qualities need to be evaluated. There are very few machine learning studies about quality attributes and related metrics (cf., Alonso et al., (1994) for an example). However, the knowledge engineering community has produced quite a few studies in this area. A notable study (Doyle & Verbruggen, 1992) explains that for all aspects of software to be controlled a diverse range of measurements are required. We therefore suggest the development of a quality attribute taxonomy, under which existing evaluation metrics can be sorted. Such a taxonomy would arguably enable us to more easily map application requirements to the corresponding evaluation metrics. For the purpose of discussion, we borrow some examples of quality attributes from the software engineering field that could be included in the taxonomy: performance, as an attribute, could be further divided into time, space, and accuracy. Other relevant quality attributes that could be included are: comprehensibility, complexity and interestingness. This example is by no means complete, but hopefully it could serve as a basis for further discussions and research. A quality attribute taxonomy enables us to categorize metrics according to which quality is evaluated, or in the case of multi-criteria metrics, which qualities are evaluated. It also makes the process of comparing different metrics more straight-forward. This is partly due to the fact that the quality attribute abstraction lets us employ attribute selection and analysis tools commonly used in areas such as software engineering instead of selecting between metrics immediately. For instance, it is possible to use the Analytic Hierarchy Process (AHP), which provides a comprehensive and rational framework for structuring a problem, for representing and quantifying its elements, for relating those elements to overall goals, and for evaluating alternative solutions (McCaffrey, 2005). It should be noted that we have not validated the use of AHP for the task of selecting quality attributes for a supervised concept learning evaluation problem. However, we merely point out the possibility to use such methods if the higher abstraction of quality attributes is used. The main motivation for using the quality attribute concept in machine learning evaluation can be described as a possible way to enforce a more structured approach to the evaluation process. It is also possible to impose a higher level of detail in the taxonomy by dividing the metrics available for each quality attribute into groups of, e.g., subjective and objective metrics. In order to visualize our suggested approach, we describe an example scenario. Let us assume that we intend to develop a mobile device decision support application. The target platform severely restricts the amount of available systems resources such as memory and processing speed. Additionally, it is crucial that the number of false positives is below a certain threshold and that the generated classifier is human-understandable. It is quite intuitive to map these requirements to a set of quality attributes. We can then apply a suitable method for ranking and quantifying the importance of the different quality attributes. This, in turn, makes it possible for us to select metrics that correspond to the quality attributes as well as selecting acceptable ranges and explicit weights for each metric to balance important trade-offs. Since we have structured the main problem into a number of sub problems, like assessment of application requirements, selection of candidate quality attributes, ranking of quality attributes, mapping from quality attributes to metrics and acceptable ranges, and formulation of a CEF configuration, it is possible to validate each sub problem separately. Researchers could also compare studies on several abstraction levels. For instance, it would be possible to evaluate which quality attribute selection method is most suitable in general, or for a certain type of problem.

6. Summary and Future Work

We recognize that there is a need for a structured way of evaluating classifiers and that there is no metric so versatile and general that it can be used for a particularly

broad range of applications. We therefore suggest the use of a generic multi-criteria evaluation method that can be customized for the application at hand. Additionally, we have suggested an approach for selecting which metrics to evaluate when using the generic evaluation method for a particular problem. Our approach is to use the concept of quality attributes as a means to enforce structure in the evaluation process by lifting the abstraction level from selecting between different metrics to the selection of quality attributes that can be more directly mapped from application-critical requirements. A direction for future work is to develop a generic mapping from application requirements to metrics.

References

Alonso, F., Maté, L., Juristo, N., Muñoz, P. L., Pazos, J. (1994). Applying Metrics to Machine Learning Tools: A Knowledge Engineering Approach. *AI Magazine*, 15(3), 63-75.

Andersson, A., Davidsson, P., Lindén, J. (1999) Measure-based Classifier Performance Evaluation. *Pattern Recognition Letters*. 20(11-13), 1165-1173.

Bûcila, C., Caruana, R., Niculescu-Mizil, A. (2006). Model Compression. *Proceedings of the 12th International Conference on Knowledge Discovery and Data Mining*.

Caruana, R., Niculescu-Mizil, A. (2006). An Empirical Comparison of Supervised Learning Algorithms. *Proceedings of the 23rd International Conference on Machine Learning*.

Deb, K. (2001). Multi-Objective Optimization Using Evolutionary Algorithms. Wiley.

Dietterich, T. G. (1996). *Proper Statistical Tests for Comparing Supervised Classification Learning Algorithms* (Technical Report), Department of Computer Science, Oregon State University, USA.

Doyle, P., Verbruggen, R. (1992). Applying Metrics to Rule-based Systems. *Proceedings of the 4th International Conference on Software Engineering and Knowledge Engineering*.

Elazmeh, W., Japkowicz, N., Matwin, S. (2006). *A Framework for Measuring Classification Difference with Imbalance* (Technical Report WS-06-06). Proceedings of the 2006 AAAI Workshop on Evaluation Methods for Machine Learning. AAAI Press.

Ferri, C., Flach, P., Hernandez-Orallo, J. (2002) Learning Decision Trees using the Area under the ROC Curve. *Proceedings of the 19th International Conference on Machine Learning*.

Freitas, A. (2006). Are We Really Discovering "Interesting" Knowledge from Data? *BCS-SGAI Expert Update*. 9(1), 41-47.

Herschtal, A., Raskutti, B. (2004) Optimising Area under the ROC Curve using Gradient Descent. *Proceedings of the International Conference on Machine Learning*.

Holt, M. J. J., Semnani, S. (1990) Convergence of Back-propagation in Neural Networks using a Log-likelihood Cost Function. *Electronics Letters*. 26(23), 1964-1965.

Huang, J., Ling, C. X. (2006). *Evaluating Model Selection Abilities of Performance Measures* (Technical Report WS-06-06). Proceedings of the 2006 AAAI Workshop on Evaluation Methods for Machine Learning. AAAI Press.

Joachims, T. (2005). A Support Vector Method for Multivariate Performance Measures, *Proceedings of the 22nd International Conference on Machine Learning*.

Lavesson, N., Davidsson, P. (2008). Generic Methods for Multi-criteria Evaluation. *Proceedings of the SIAM International Conference on Data Mining*.

Lavesson, N., Davidsson, P. (2006). Quantifying the Impact of Learning Algorithm Parameter Tuning, *Proceedings of the 21st AAAI National Conference on Artificial Intelligence*.

McCaffrey, J. (2005). Test Run: The Analytic Hierarchy Process. *Microsoft Developer's Network Magazine*.

Musicant, D., Kumar, V., Ozgur, A. (2003) Optimizing f-measure using Support Vector Machines. *Proceedings of the 16th International Florida Artificial Intelligence Research Society Conference*.

Nakhaeizadeh, G., Schnabl, A. (1997). Development of Multi-Criteria Metrics for Evaluation of Data Mining Algorithms. *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*.

Provost, F., Fawcett, T., Kohavi, R. (1998). The Case against Accuracy Estimation for Comparing Induction Algorithms. *Proceedings of the 15th International Conference on Machine Learning*.

Salzberg, S. (1997). On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach, *Data Mining & Knowledge Discovery*. 1(3), 317-327.

Stone, M. (1974) Cross-Validatory Choice and Assessment of Statistical Predictions. *Royal Statistical Society*. B, 36, 111-147.

Suzuki, E., Ohno, T. (1999) Prediction Rule Discovery Based on Dynamic Bias Selection. *Proceedings of the 3rd Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*.

Urli, B., Nadeau, R. (1999). Evolution of Multi-Criteria Analysis: A Scientometric Analysis. *Multi-Criteria Decision Analysis*. 8, 31-43.

Witten, I., H., Eibe, F. (2005) Data mining - Practical Machine Learning Tools and Techniques. Elsevier.