|  |  |  |
| --- | --- | --- |
| ITU Logo | INTERNATIONAL TELECOMMUNICATION UNION**TELECOMMUNICATIONSTANDARDIZATION SECTOR**STUDY PERIOD 2017-2020 | FG-AI4H-D-039 |
| **ITU-T Focus Group on AI for Health** |
| **Original: English** |
| **WG(s):** | N/A | Shanghai, 02-05 April 2019 |
| **DOCUMENT** |
| **Source:** | Fraunhofer HHI |
| **Title:** | Aspects of Evaluation Procedures for Machine Learning Algorithms |
| **Purpose:** | Discussion |
| **Contact:** | Nils StrodthoffFraunhofer HHI Germany | Tel: +49 30 31002-104Fax: +49 30 31002-558Email: nils.strodthoff@hhi.fraunhofer.de  |
| **Contact:** | Arun ShroffXtend.AI USA | Tel: +1-908-208-1100Fax: Email: arun@xtend.ai  |
| **Contact:** | Wojciech Samek Fraunhofer HHIGermany | Tel: +49 30 31002 417Email: wojciech.samek@hhi.fraunhofer.de  |

|  |  |
| --- | --- |
| **Abstract:** | We discuss aspects of evaluation procedures for Machine Learning algorithms focussing on (a) the selection of representative subsets (b) ways of addressing covariant shift and (c) performance metrics, score estimation and comparison classification algorithms and provide references to existing literature. |

1 Introduction

This document serves two purposes: On the one hand to review the vast amount of Machine Learning literature on this evaluation aspects and on the other hand to provide practical recommendations for practical applications in the Focus Group. In particular, we discuss three aspects:

1. Methods for selecting a representative subset from a larger dataset

2. Methods for detecting and correcting covariate shift

3. Evaluation metrics, score estimation and statistical tests

We will try to review the existing literature and provide a series of practical recommendations for each of the three aspects. In the following we focus mostly on classification tasks as these represent the predominant task among the identified use cases.

2 Methods for create a representative subset from a given dataset

This aspect was the focus of input document FG-AI4H-B-027. Here one has to distinguish two cases depending on whether a baseline (classification) model is available or not. In the latter case the subset has to be created using just the dataset itself. These methods could be applied both for extracting a reduced training set or a test set from a given dataset or simply to characterize a dataset via prototypical examples

2.1 No baseline model available

Here again one has to distinguish selection strategies depending on whether the just the input data (without label), just the label or both should be used. Even though input data will always be available, it might be difficult to define a distance measure on it (e.g. in the self-assessment use case), which is a prerequisite for many selection algorithms.

2.1.1 Input data only

At this point it is worth noting that all selection algorithms could operate on the raw input or on intermediate features obtained from some pretrained model that is not domain specific (e.g. intermediate features of a convolutional network pretrained on Imagenet). If it is possible to define a distance measure on the input data, a classic technique is for example the use of k-medoid clustering [Kaufman1987], for which several iterative algorithms exist. K-medoid clustering can be seen as analogue of k-means clustering with the additional constraint that the presented prototypes have to be contained in the dataset.

Another class of algorithms that address this problem address the problem in a set-cover setting: In this case the algorithm aims to find a subset of the original dataset such that a utility function reaches a desirable fraction of the value attained on the whole dataset. The definition of the utility function depends on the application domain; there exist for example distinguished solutions for image summarization [Simon2007] and document summarization [Lin2011]. Recently, it has been demonstrated, how set cover algorithms can be implemented in a highly efficient and distributed manner [Badanidiyuru2014, Mirzasoleiman2015].

2.1.2. Class label only

In the case where only the label information can be used for selecting the subset, the most appropriate solution is to use stratified sampling, i.e. to sample in such a way that the relative frequencies of the respective classes in the subsampled dataset approximately reflect those in the original dataset. Stratified sampling has been shown for example to reduce the variance of crossvalidation predictions [Kohave1995]. While stratified sampling is straightforward to implement for single-label classification, it has received much less attention for multi-label classification, but even here practical solutions exist [Sechidis2011, Szymański2017]. Finally, in many applications the number of different class labels is large, which renders traditional stratified sampling impractical. However, often class labels can be organized hierarchically, which enables the use of stratified multistage sampling methods such as [Jain2006].

2.1.3 Both input data and class label

In case both input data and label can be used for subset creation, cover-digraph or prototype selection for interpretable classification algorithms are applicable. The former includes approaches such as [Priebe2003] and can be seen as extension of set-cover as discussed in 2.2.1 to the classification setting. Prototype selection methods such as [Bien2011] aim to select prototypes that maximize the coverage within the class, but minimize the coverage across classes. There are also extensions of the latter approach that use not only prototypical samples but also outliers to characterize different classes in the dataset, see [Kim2016], which can even be applied in the absence of class labels.

2.2 Baseline model available

In case a baseline model is available, the most straightforward approach is to use intermediate features of this pretrained model for further processing according to 2.1.1 or 2.1.3. Other possibilities that have also been discussed in input document FG-AI4H-B-027, include the use of Fisher Kernels [Khanna2018], which aim to approximate the empirical test data distribution using samples from the training set, or using influence functions [Koh2017]. Both latter solutions suffer from a high computational complexity.

2.3 Recommendations

The most important recommendation for practical applications is that there is no strategy that applies to all cases, i.e. the optimal strategy is task-specific. Using a baseline classification model in the subset creation process is not necessarily the best strategy unless it is easy to build or standard in the respective field as it requires domain expertise to build it and a poor baseline classification model might hamper the subset creation process rather than help it. The central choice is whether to use input data/label or both for subset selection, where in all cases practical algorithms for subset sampling exist. However, the implications of the different approaches for subset creation for downstream tasks such as covariate shift detection have to our knowledge not been investigated systematically.

3. Detecting and correcting for covariate shift

Covariate shift designates the situation where the distribution of the inputs used as predictors (covariates) changes between training and test stages [Sugiyama2012, Moreno-Torres2012]. It has to be distinguished from dataset shift, where the joint distribution of inputs and the output also changes, see [Quionero-Candela2009, Moreno-Torres2012].

3.1 Detecting covariate shift

There is a simple algorithm for detecting covariate shift that has become common folklore. It involves training a classifier to predict if a given data point originated from training set or test set (discarding the original labels) and to use the correlation coefficient evaluated on a hold-out set to judge the performance of the classifier and hence on the separability of training and test set. A well-performing classifier suggests the need for methods to alleviate covariate shift as discussed below.

3.2 Correcting for covariate shift

The main approach for alleviating the effects of covariate shift is to use importance-weighting. The idea is to weigh the training distribution to make it more similar to the test distribution. The weight factors can be determined using the Kullback-Leibler Importance Estimation Procedure (KLIEP) method [Sugiyama2008], which is a stable algorithm that directly estimates the ratio between training and test distributions and avoids density estimation. The importance-weighting algorithm is described in [Sugiyama2005] and was even shown to work for relatively small sample sizes. There also exist extensions to correcting for covariance shift in crossvalidation estimates, see [Sugiyama2007].

3.3 Recommendations

Releasing unlabelled test set samples would allow people to use covariate shift adaption techniques to adapt for covariate shift between their own private training set and the undisclosed test set using established techniques. This would help to reduce potential bias in the evaluation procedure.

4. Performance metrics, score estimation and comparison between algorithms

An extensive review on this topic is beyond the scope of this document, instead we refer to excellent reviews in the literature [Santafe2015, Japkowic2011]. Both texts discuss the overall evaluation process with focus on practical recommendations, in particular focussing on the following three aspects:

* the choice of the performance metric
* the way the score is estimated on a given dataset
* ways of comparing the performance of two classification algorithms

4.1 Performance metrics

Performance metrics can be roughly divided into three classes, see [Santafe2015] for details:

* non-balanced scores (such as accuracy), which do not aim to account for a trade-off between positive and negative class scores
* balanced scores (such as area under receiver-operator curves (AUC ROC) or Fβ scores), that do account for a trade-off between positive and negative class scores.
* calibration scores (such as log loss), which also take into account the prediction certainty of the model

One has to be aware of the particular strengths and weaknesses of the different classifiers, e.g. using accuracy as a performance metric might be an appropriate choice for datasets that are well-balanced with respect to the distribution of the class labels. AUC is another very popular metric, which is less sensitive to class imbalance. However, it treats the cost of misclassification differently for each classification algorithm, see [Santafe2015] for details. Therefore lesser known metrics that avoid these difficulties such as H-measures [Hand2014] should be taken into consideration.

4.2 Score estimation

The most predominantly used score estimation techniques are evaluation on a hold-out set or various variants of crossvalidation or bootstrap algorithms. The evaluation on a hold-out set is most relevant in this setting since the other approaches require retraining the algorithm on different subsets of the original dataset.

As emphasised in [Santafe2015] and even more in [Isaksson2008], which is a dedicated review focusing on the evaluation of ML algorithms on small datasets, reliable score estimation and hence performance evaluation is very difficult for small sample sizes. In the small sample regime, two fundamentally different sources of variance were identified, see [Brain1999, Isaksson2008, Santafe2015], where the first relates to the way the small dataset was obtained from the underlying probability distribution and the second relates to the way the existing samples are used to estimate the score. The unfortunate situation in the case of small sample sizes is the fact that typically both variances are large. [Isaksson2008] advocates the use of Bayesian confidence intervals, to provide bounds on the error in the score estimation. A different approach is taken by permutation tests [Good2000], which aim to quantify if the dataset is large enough to capture the complexity of the given classification problem. It is important to be aware of the particular intricacies of small sample sizes to avoid drawing wrong conclusions on the algorithms to be evaluated.

4.3 Comparing classification performance

The most predominantly used techniques for comparing algorithmic performance for a given performance metric and score estimation method is by using either statistical tests or by providing confidence intervals for predictions. Performance comparison using statistical tests such as McNemar’s test using only a single dataset is generally considered unsafe as the obtained score values are not independent. The problem with the extension to several evaluation datasets is the fact that the score measured on several datasets is hardly commensurable. A good default approach when dealing with performance comparison of two classification algorithms on several datasets is the Wilcoxon signed-rank test or non-parametric omnibus test such as the Friedmann test for the comparison of several classification algorithms. See [Santafe2015] for an extensive comparison.

4.4 Recommendations

Reiterating the recommendations from the previous sections, there is not a single best evaluation strategy for all cases. The choice of the evaluation metric is of crucial importance for a reliable evaluation. Its selection should depend on the characteristics of the underlying dataset such as label imbalance etc., miss-classification costs if available rather than defaulting to widely used metrics such as accuracy or AUC. Special care has to be devoted to the effects of small sample sizes on the score evaluation process. When comparing different algorithms based on statistical tests, particular care has to be taken with respect to the setting i.e. evaluation of two/multiple algorithms on a single/multiple datasets, the use of appropriate statistical tests for the chosen setting and a proper understand of which information the test can actually provide. Concerning all three aspects the reader is referred to excellent reviews on these topics [Santafe2015, Japkowic2011].

5. References

[Badanidiyuru2014] A. Badanidiyuru, B. Mirzasoleiman, A. Karbasi, and A. Krause. Streaming submodular maximization: Massive data summarization on the fly. In KDD. ACM, 2014.

[Brain1999] D. Brain and G.I. Webb. On the effect of data set size on bias and variance in classification learning. In: Proceedings of the 4th Australian knowledge acquisition workshop, 1999, 117–128.

[Good2000] P.I. Permutation test: a practical guide to resampling methods for testing hypotheses. Springer, 2000.

[Hand2014] D.J. Hand and C. Anagnostopoulos. A better Beta for the H measure of classification performance. Pattern Recogn Lett 40, 2014, 41–46.

[Isaksson2008] A. Isaksson, M. Wallman, H. Göransson, and M.G. Gustafsson. Cross-validation and bootstrapping are unreliable in small sample classification. Pattern Recognition Letters, 29(14), 2008, 1960-1965.

[Jan2006] A.K. Jain and R.E. Hausman. Stratified Multistage Sampling. In Encyclopedia of Statistical Sciences (eds S. Kotz, C. B. Read, N. Balakrishnan, B. Vidakovic and N. L. Johnson), 2006.

[Japkowicz2011] N. Japkowicz and M. Shah. Evaluating learning algorithms: a classification perspective. Cambridge University Press, 2011.

[Kaufman1987] L. Kaufman and P. Rousseeuw. Clustering by means of medoids. North-Holland, 1987.

[Khanna2018] R. Khanna, B. Kim, J. Ghosh, and O. Koyejo. Interpreting Black Box Predictions using Fisher Kernels. arXiv preprint arXiv:1810.10118, 2018.

[Kim2016] B. Kim, R. Khanna and O. Koyejo. Examples are not enough, learn to criticize! Criticism for Interpretability. In Advances in Neural Information Processing Systems 29, 2016.

[Kohavi1995] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2 (IJCAI'95), Vol. 2. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, 1137-1143.

[Lin2011] N. Lin and J. Bilmes. A class of submodular functions for document summarization. In ACL, 2011.

[Mirzasoleiman2015] B. Mirzasoleiman, A. Karbasi, A. Badanidiyuru, and A. Krause. Distributed submodular cover: Succinctly summarizing massive data. In NIPS, 2015.

[Moreno-Torres2012] J.G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N.V. Chawla, F. Herrera, A unifying view on dataset shift in classification, Pattern Recognition, Volume 45, Issue 1, 2012, Pages 521-530.

[Priebe2003] C.E. Priebe, D.J. Marchette, J.G. DeVinney, and D.A. Socolinsky. Classification using class cover catch digraphs. Journal of classification, 2003.

[Quionero-Candela2009] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N.D. Lawrence. Dataset shift in machine learning. The MIT Press, 2009.

[Santafe2015] G. Santafe, I. Inza, and J.A. Lozano. Dealing with the evaluation of supervised classification algorithms. In Artif Intell Rev 44: 467, 2015.

[Sechidis2011] K. Sechidis, G. Tsoumakas, I. Vlahavas. On the Stratification of Multi-label Data. In D. Gunopulos, T. Hofmann, D. Malerba, M. Vazirgiannis (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2011. Lecture Notes in Computer Science, vol 6913. Springer, Berlin, Heidelberg, 2011.

[Simon2007] M. Simon, N. Snavely, and S.M. Seitz. Scene summarization for online image collections. In ICCV, 2007.

[Sugiyama2005] M. Sugiyama and K.R. Müller. Input-dependent estimation of generalization error under covariate shift. Statistics & Decisions, 23(4/2005), 2005, 249-279.

[Sugiyama2007] M. Sugiyama, M. Krauledat and K.R. Müller. Covariate shift adaptation by importance weighted cross validation. Journal of Machine Learning Research, 8(May), 2007, 985-1005.

[Sugiyama2008] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation for covariate shift adaptation. Annals of the Institute of Statistical Mathematics, 60(4), 2008, 699-746.

[Sugiyama2012] M. Sugiyama and M. Kawanabe. Machine learning in non-stationary environments: Introduction to covariate shift adaptation. MIT press, 2012.

[Szymański2017] P. Szymański, and T. Kajdanowicz. A Network Perspective on Stratification of Multi-Label Data. First International Workshop on Learning with Imbalanced Domains: Theory and Applications. 2017.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_