# Improving Cross-Validation Based Classifier Selection using Meta-Learning

Jesse H. Krijthe
*Delft University of Technology*
*jkrijthe@gmail.com*

Tin Kam Ho
*Bell Laboratories, Alcatel-Lucent*
tin.ho@alcatel-lucent.com

Marco Loog
*Delft University of Technology*
*mloog@tudelft.nl*

## Abstract

*In this paper we compare classifier selection using cross-validation with meta-learning, using as meta-features both the cross-validation errors and other measures characterizing the data. Through simulation experiments we demonstrate situations where meta-learning offers better classifier selections than ordinary cross-validation. The results provide some evidence to support meta-learning not just as a more time efficient classifier selection technique than cross-validation, but potentially as more accurate. It also provides support for the usefulness of data complexity estimates as meta-features for classifier selection.*

## 1 Introduction

In selecting a classifier for a classification problem, a standard procedure is to determine the cross-validation errors of a wide range of classifiers, and to choose one with the lowest cross-validation error [9]. However, problems with this procedure have been observed before. For small samples, cross-validation has been shown to be unreliable [1, 4]. Yet with large training samples, this approach to selection can become very expensive. Research in meta-learning has given us measures that are less computationally intensive but can still aid the classifier choice [2]. There meta-learning is considered to be a time efficient alternative to the presumably more accurate cross-validation.

In this paper, we demonstrate that meta-learning can provide not only efficiency, but also potential improvement to the *accuracy* of classifier selection over cross-validation. We show this by treating the classifier selection problem itself as a classification problem, where the cross-validation errors are used as features describing the datasets, and the goal is to predict which classifier works better for each dataset. Viewing cross-validation from this perspective, we can compare it to other meta-learning procedures, and compare the roles of these features and decision rules to their counterparts in ordinary classification problems. We also show that additional meta-features describing properties of the dataset can improve selection accuracy, providing additional evidence for the potential utility of so-called data complexity measures [3, 6, 7] in guiding classifier selection.

The rest of the paper is structured as follows. Section 2 presents cross-validation classifier selection in the form of a meta-classifier. Section 3 discusses a situation where this meta-classifier fails and how to improve upon it. Section 4 demonstrates the usefulness of adding new features to this meta-problem. We end with a discussion of our results and a conclusion.

## 2 Classifier Selection Problem

The premise of effective classifier selection depends on two assumptions. First, the classifier domains of competence [7], meaning the types of problems a classifier provides the best performance on, do not overlap completely. Hence a careful selection can potentially optimize performance. Secondly, there exist measures characterizing each dataset that we can rely on to determine when one classifier outperforms another. These measures can be considered as a parametrization of the space of all classification problems, $S$. An example of such a parametrization is cross-validation errors of several classifiers, which characterize each dataset by the difficulties it causes to the concerned classifiers. Another example from the meta-learning literature is the 'data complexity' measures [3, 6, 7] describing geometrical and topological properties of a dataset.

We believe *good* parameterizations of $S$ need to have some additional properties. (1) Preferably the domains of competence of the classifiers are compact in the chosen parameterization. This means the datasets a classifier works well on are close to one another in terms of the chosen parameters, and are locally continuous, hence some separation of the domains is possible. (2) They can be efficiently computed, which has been the focus of most research in meta-learning and an important prerequisite for their practical appeal. (3) Finally, it would be helpful if the parameters are interpretable. This property helps diagnose problems when performance is worse than required, as well as gives insight into the properties of different algorithms. Combined with compactness in (1), this allows for an understanding of what specific type of problems a classifier works well for.

Note that the assumptions mentioned earlier are the same as for any classification problem. In fact we can treat the classifier selection problem as another classification problem, or a "meta-problem". This meta-problem has as its meta-features the measures characterizing the dataset and the meta-classes are the classifiers that perform best on each dataset.

Seen through this framework, classifier selection using cross-validation is a meta-classifier that makes an additional assumption: not only is it possible to distinguish between the domains of competence, but we can do this by a simple static decision rule. The selection rule is in fact an untrained meta-classifier: we *always* select the classifier with the lowest cross-validation error, regardless of our experience on other datasets.

In section 3 we will demonstrate that there exist universes of problems where the space formed by the cross-validation errors does not support the superiority of this decision rule. Instead, using a different meta-classifier allows for an improvement in accuracy over cross-validation based selection. In section 4 we study how adding extra meta-features to the meta-problem may allow us to further improve accuracy, demonstrating cross-validation errors do not necessarily carry all information useful in selecting a classifier.

## 3 Cross-Validation in Meta-Learning

Cross-validation based selection employs a static decision rule: the classifier with the lowest cross-validation error gets selected regardless of experience obtained on selecting classifiers on previous problems. In case of a choice between two classifiers, this rule can be visualized by the diagonal in the 2D cross-validation error space (see Figure 1). To illustrate how a situation may arise where this rule is too rigid and we can improve upon it, we sample classification problems from a chosen universe that is generated systematically. The goal is to visualize the potential issue of a static selection procedure using a simplified but intuitive example. We believe similar results also occur for more complex universes.

The example universe of problems we consider consists of instances of two problems: a problem with strongly overlapping, two dimensional Gaussian classes (call it $\mathbf{G}$) and a more complex problem with 'banana-shaped' classes with Gaussian noise and small overlap (call it $\mathbf{B}$) (see Figure 1). We introduce some differences to the instances of $\mathbf{G}$ by selecting the separation of the class means from a uniform distribution. For instances in $\mathbf{B}$, we change the variance of the Gaussian noise. In this way we create a family of 500 instances for each problem: $\mathbf{G} = \{G_1, G_2, ..., G_{500}\}$, and $\mathbf{B} = \{B_1, B_2, ..., B_{500}\}$. From each of these problems $B_i$ or $G_i (i = 1, ..., 500)$, we extract a small sample of size $N_{train}$ (varying uniformly randomly between 20 and 100) for training, call this $G_i^{train}$ and $B_i^{train}$, and use the rest ($N_{test} = 20000 - N_{train}$) for testing, $G_i^{test}$ and $B_i^{test}$. The small training set size is chosen to cause cross-validation error to have high variance. In real world problems, with complex boundaries, this effect will occur for larger sample sizes. The large test set size is used to get an accurate estimate of the real error of the classifier for the problem instance.

The goal is to choose between two classifiers: a nearest mean classifier (NM) and a 1-nearest neighbor classifier (1-NN). Casting this as a meta-learning problem, we create the meta-features and meta-classes as follows. For each $G_i$, we obtain a 10 times repeated,10-fold cross validation error for the NM classifier using $G_i^{train}$. That is, for each of ten passes, NM is trained using $N_{train} * \frac{9}{10}$ samples, and tested using the remaining $N_{train} * \frac{1}{10}$ samples. The total number of errors over the ten passes, divided by $N_{train}$ is the error for one repeat. Repeating this 10 times, the average of the 10 repeats is used as meta-feature 1 for $G_i$. Doing the same for the 1-NN classifier, we obtain the value of meta-feature 2 for $G_i$, and similarly for each $B_i$.

The meta-class for each of $G_i$ or $B_i$ is the classifier that performs the best for the testing data $G_i^{test}$ or $B_i^{test}$. This is determined by, say, using all of $G_i^{train}$ for training the NM classifier, estimating the error rate of the resultant classifier on $G_i^{test}$, and doing the same for the 1-NN classifier. If the NM gives a lower error, $G_i$ is labeled with the true meta-class NM, and vice versa. Similarly we obtain the true meta-class for $B_i$.

For the meta-learning problem, a random half of the 500 instances from $\mathbf{G}$ or $\mathbf{B}$ are used for meta-training (denoted by $\mathbf{G}^{TRAIN}$ and $\mathbf{B}^{TRAIN}$) and the rest are reserved for meta-testing ($\mathbf{G}^{TEST}$ and $\mathbf{B}^{TEST}$). Recall that the cross validation rule corresponds to a static classifier, and hence requires no training. We compare

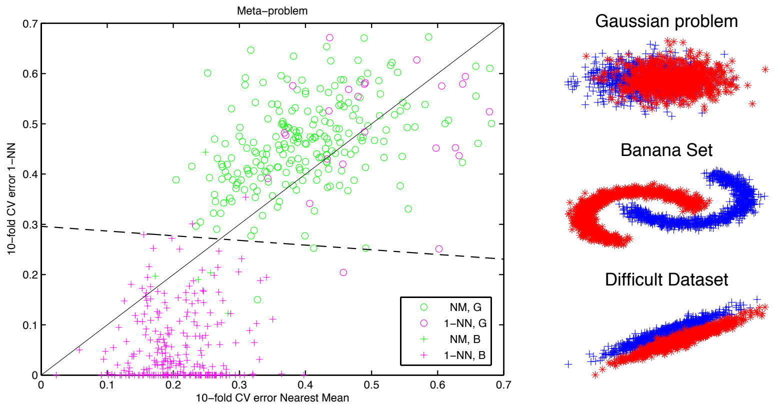

**Figure 1. On the left: an example of a universe of problems where static cross-validation selection does not perform well. On the right: the problems 'Gaussian', 'Banana Set' as discussed in Section 3, and the 'Difficult Dataset', a 2D version of the problems discussed in Section 4.**

this to a trainable classifier that is a support vector machine, trained with the meta-features for $\mathbf{G}^{TRAIN}$ and $\mathbf{B}^{TRAIN}$. The comparison is shown in Figure 1.

In Figure 1, the points plotted represent problem instances from $\mathbf{G}^{TEST}$ and $\mathbf{B}^{TEST}$. Their coordinates are given by the two meta features, and they are marked with symbols denoting their source family ($\circ$ for the Gaussian problems, or $+$ for the Banana problems), and colors denoting their true meta-class (green for NM, and pink for 1-NN).

We can observe that 1-NN almost always gives better performance for the cluster of Banana problems (almost all points in the cluster of $+$ are marked pink). The other cluster formed by the Gaussian problems primarily favors the NM classifier (most $\circ$'s are marked green). Note that in this particular universe, the cross-validation rule does not give the best separation: the decision boundary crosses through the cluster of Gaussian problems ($\circ$'s), forcing one to choose the 1-NN classifier for the points below the diagonal, while performance in this cluster is in fact almost always better with the NM classifier. The selection error, defined as the proportion of datasets in $\mathbf{B}^{TEST} \cup \mathbf{G}^{TEST}$ assigned by the meta-rule to a classifier different from their true meta-class, is $0.16$ for cross-validation. The support vector machine reduces the selection error to $0.06$, offering a more accurate alternative.

## 4 Adding Meta-Features

The example in the previous section has shown that learning the selection rule can improve performance in some cases where the cross-validation errors have high variance. In more elaborate universes of problems, the meta-classes may overlap heavily, making it difficult to select a classifier based solely on the cross-validation errors. Other measures characterizing the dataset may reduce this overlap by introducing extra features that can aid the selection task.

Consider a universe of 100 dimensional linearly separable two-class problems where most of the class variance is along the class boundary (the 'Difficult Dataset' from the PRTools library (prtools.org)), see Figure 1 (right). We introduce some variability to the universe by randomly selecting the class-priors using a uniform distribution between 0 and 1. Similar to Section 3, we estimate cross-validation errors with different sample sizes ($N_{train}$ is again between 20 and 100). The classifiers we considered are now the nearest mean classifier and the Fisher classifier. Cross-validation selection gives a selection error of $0.265$. Using a nearest neighbor classifier as a meta-learner, we improve the error to $0.234$. Adding to the meta-features an extremely simple characterization of the sampling density of the dataset, the number of objects divided by the number of dimensions, causes a further reduction in error to $0.204$.

## 5 Discussion

We consider classifier selection as a classification problem in its own right. Recall that the reason for doing classifier selection in the first place is an assumption that different problems may need different classifiers. If we really believe this to be true, we should assume the same for the meta-problem, thereby allowing different meta-classifiers. Yet cross-validation based selection is a static, universal rule. We have shown that it could indeed be suboptimal for this meta-problem, and could be inferior to another rule obtained by meta-learning.

The reason meta-learning can outperform cross-validation is that it leverages experience from other previously seen instances from the same source problem, whereas the cross-validation rule remains static. During our experiments, we have often come across situations where one classifier outperforms all others on most instances of the same problem family. A meta-learner might learn that this classifier should be preferred for all problems from this family. This could be a better choice than deciding, for each instance, solely based on cross-validation error, as that may be unreliable when small samples are used to train a complex classifier, which is exactly the case where cross-validation selection frequently fails. An interesting observation that follows from this is that the belief by a practitioner working on a specific application domain that a certain classifier should always be preferred is not necessarily irrational and may be a more accurate rule than using cross-validation.

The problem universe where extra measures characterizing the data improve selection accuracy shows that these measures can not only serve as computationally efficient proxies to cross-validation errors, but actually improve performance. This provides support for the goal of approaches like data complexity measures. To make this approach more useful, however, further research should focus on finding measures that take into account the properties a good parameterization should have. Especially interpretability will likely help adoption of these extra measures, because this not only tells the user when each classifier has good performance, but also helps explain why this is the case.

We have chosen to use extremely small training sets and large test sets to highlight the issue of variance in cross-validation error estimation. In real world data, this could happen when complex class boundaries must be inferred from relatively limited training samples. Another situation where we expect cross-validation selection to become unreliable is when choosing between a large number of similar classifiers. As this number increases, at least one of the cross-validation errors will likely be low by chance. Meta-learning allows us to correct for this case. Even for a task where large samples are eventually available, the procedure could be used to select a classifier early on, and before the large samples arrive. As more data becomes available we will be able to evaluate whether this choice is correct.

The key question that remains is whether one can get this improved selection strategy to work not just in some artificial universe but in a domain of real-world datasets as well. This can only be established through an empirical study. Unfortunately, the lack of availability of large numbers of standardized real-world datasets is a recurring problem in meta-learning research. Some projects [5, 8] supporting the publication of datasets might help alleviate this problem, but currently offer still too few real-world datasets. For specific restricted application domains, the possibility of finding a reasonably large set of datasets is more plausible.

## 6 Conclusion

The purpose of this paper has been to show that the effectiveness of the cross-validation selection could be impacted by factors such as the variance of the cross-validation error estimates, not unlike those factors affecting the accuracy of an ordinary classification problem. As we have demonstrated, in some simple universes, cross-validation selection is not the best strategy. In these cases, treating the cross-validation errors as meta-features in a meta-classification problem offers a chance to increase selection accuracy using other decision rules. Adding other features characterizing the dataset may also increase selection accuracy, which supports the idea of data complexity measures and similar approaches to meta-learning.

## References

[1] U. Braga-Neto and E. R. Dougherty. Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20(3):374–380, 2004.

[2] P. Brazdil, J. Gama, and B. Henery. Characterizing the applicability of classification algorithms using meta-level learning. In F. Bergadano and L. De Raedt, editors, *Machine Learning: ECML-94*, volume 784 of *Lecture Notes in Computer Science*, pages 83–102. Springer Berlin / Heidelberg, 1994.

[3] T. K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):289–300, 2002.

[4] A. Isaksson, M. Wallman, H. Göransson, and M. G. Gustafsson. Cross-validation and bootstrapping are unreliable in small sample classification. *Pattern Recogn. Lett.*, 29:1960–1965, October 2008.

[5] P. Liang and J. Abernethy. mlcomp.org, 2009.

[6] N. Macià, T. K. Ho, A. Orriols-Puig, and E. Bernadó-Mansilla. The lansdcape contest at icpr'10. In S. A. Devrim Ünay and Z. Çataltepe, editors, *Contests in ICPR 2010*, volume 6388 of *Lecture Notes in Computer Science*, pages 29–5, Berlin, Heidelberg, August 2010. Springer.

[7] E. Mansilla and T. K. Ho. On classifier domains of competence. In *Proceedings of the 17th ICPR*, volume 1, pages 136 – 139 Vol.1, aug. 2004.

[8] C. S. Ong, M. L. Braun, S. Sonnenburg, S. Henschel, and P. O. Hoyer. mldata.org, 2009.

[9] C. Schaffer. Selecting a classification method by cross-validation. *Machine Learning*, 13:135–143, 1993.