A Class Boundary Selection Criterion for Classification

David Ha * Juliette Maes ** Yuya Tomotoshi * Charles Melle *** Hideyuki Watanabe ****
Shigeru Katagiri * Miho Ohsaki *

1 Introduction

We address the problem of selecting the optimal status for a general form of classifier model, namely the status that leads to the minimum expected classification error called Bayes error rate. The optimum classifier status corresponds to a classification boundary surrounded by uncertain samples, namely samples whose posterior probability is equal for the two most probable classes. However, accurate estimation of posterior probabilities using finite data inevitably requires the specification of a classifier model whose status must be optimally, which in turn requires accurate posterior probability estimation. To break this vicious circle, we propose a two-step procedure, which first identifies training samples surrounding the estimated boundary, and then assesses their uncertainty without the need for a model specification. Experimental results on synthetic data and benchmark real life data show the potential of our approach to select the optimal classifier status.

2 Background and Outline of Proposed Method

In the standard statistical approach to pattern classification, the ultimate goal of classifier training is to achieve the minimum expected classification error through an optimal setting of classifier parameters, whose accurate estimation inevitably needs to observe an infinite amount of training samples. In practice, only a finite amount of training data is available, which raises the issue of over-fitting: achieving perfect classification error on finite training data does not guarantee a low expected classification error.

Various methods have been vigorously investigated to select the optimal model based on the estimation of the expected classification error. Though cross-validation provides a practical solution to avoid overfitting, it reduces the amount of data available for training, and requires a cumbersome search. Leave-one-out is an extreme case of cross-validation which keeps all but for one samples for training, however its high computational time restricts it to small datasets. By contrast to cross-validation, bootstrap methods [1] allow sampling with replacement. Although the resulting replication of the data can artificially avoid reducing the available data for training, it can also reduce the reliability of the estimate. Several approaches perform model selection without the use of a validation set, by making the most of the information available in the training set. For example, information criteria such as Akaike Information Criterion [2] (AIC) and Bayes Information Criterion [3] (BIC) can directly estimate the expected error in the context of sample distribution estimation, however classification tasks focus on class boundary estimation. Another example is Structural Risk Minimization [4] (SRM), which provides upper bounds on the expected error for any classification task. However the estimated upper bound can be fairly loose.

Sampling procedures such as [5] and [6] provide an alternative approach to direct error estimation. They focus instead on selecting samples deemed informative for classification (hopefully samples located near the ideal Bayes boundary) by attributing weights to the training samples. Intuitively, emphasizing correct classification on informative samples constraints the estimated boundary to approach the Bayes boundary, and therefore a classifier status which correctly classifies such samples should be appropriate. One point is, performance can significantly change from one weighting scheme to another [7], and how to define the appropriate weighting is still a research issue.

By contrast to previous approaches, our proposed method does not directly aim at estimating the expected classification error. It instead focuses on the boundary, which should approach the Bayes risk status, and is referred to as Bayes boundary in the paper. Boundary status is assessed based on the uncertainty of its *salient training samples*, though we do not attempt to attribute a measure of uncertainty to samples individually, and rather the set of salient samples as a whole. The main idea is that uncertain training samples around the Bayes boundary can be indifferently labelled as either one of the two classes. The resulting method can be applied to a general form of classifier which we describe in Section 3. Section 4 details the extraction of boundary salient samples and their use to select a classifier status. Finally, Section 5 experimentally analyzes the behavior of our model selection method on synthetic data as well as on real life data. Table 1 summarizes the main notations used in the paper.

3 Discriminative training

We consider a task of classifying a given input sample $\boldsymbol{x}$ extracted from the infinite feature space $\mathcal{X}$ into one of J classes $(C_1, \ldots, C_J)$. For the convenience of later discussions, we assume $\boldsymbol{x}$ to be of fixed dimension, though the framework can also handle variable-length patterns. The classification process $C()$ takes the fol-

* Doshisha University
** Ecole Centrale de Lille
*** Ecole Centrale de Marseille
**** ATR

Table 1: Notations

$\boldsymbol{x}$	pattern sample
J	number of classes
N	number of training samples
$\mathcal{T}$	training set
B^*	Bayes boundary
Λ	classifier model parameters to be trained
$B(\Lambda)$	estimated boundary
$\mathcal{S}(\Lambda)$	extracted boundary salient samples for $B(\Lambda)$
$\hat{\mathcal{S}}$	set containing the same pattern samples as in $\mathcal{S}$ except that class labels are flipped
$p(\boldsymbol{x};\Lambda)$	projection of $\boldsymbol{x}$ on $B(\Lambda)$
$NN(\boldsymbol{x})$	nearest neighbour of $\boldsymbol{x}$ in $\mathcal{T}$
$\mathcal{P}$	set of prototypes for a given classifier
$g_j()$	discriminant function for class C_j
$n_{features}$	number of features for a given dataset

lowing general form:

$$C(\boldsymbol{x}) = C_k \text{ iff } k = \arg\max_j g_j(\boldsymbol{x};\Lambda), \quad (1)$$

where $g_j(\boldsymbol{x};\Lambda)$ is a discriminant function of C_j, which is differentiable in class model parameters Λ to be trained. The value of $g_j(\boldsymbol{x};\Lambda)$ represents the degree to which $\boldsymbol{x}$ belongs to class $C_j, j \in [1, J]$.

Training of Λ is performed to achieve the Bayes boundary. To evaluate a trial classification in the training, we use the following misclassification measure, assuming that a training sample $\boldsymbol{x}$ belongs to class C_y:

$$d_y(\boldsymbol{x};\Lambda) = -g_y(\boldsymbol{x};\Lambda) + \log\left[\frac{1}{J-1}\sum_{j,j\neq y} e^{\psi g_j(\boldsymbol{x};\Lambda)}\right]^{1/\psi}, \quad (2)$$

where $\psi > 0$. In particular, in the two-class case, (2) is simplified as

$$d_y(\boldsymbol{x};\Lambda) = -g_y(\boldsymbol{x};\Lambda) + g_i(\boldsymbol{x};\Lambda), \quad (3)$$

where i refers to the incorrect class index for sample $\boldsymbol{x}$. Estimated boundary $B(\Lambda)$ corresponds to an equal score between the two classes, namely

$$B(\Lambda) = \{\boldsymbol{x} | d_y(\boldsymbol{x};\Lambda) = 0\}. \quad (4)$$

4 Proposed Procedure for Optimal Classifier Status Selection

4.1 Bayes boundary

For the sake of simplicity we consider the two-class case ($J = 2$). However extension to the multi-class case is straightforward because classes can be considered pairwise. If the joint probability underlying the data distribution is known, then we can choose the decision boundary which leads to the lowest expected classification error, or in other words, the Bayes risk. The minimum probability of making a mistake is obtained if input sample $\boldsymbol{x}$ is assigned to the class for which the joint probability $P(C_j, \boldsymbol{x}), j = 1, 2$ is the largest [8]. Therefore the optimal Bayes boundary corresponds to samples which satisfy (see Figure 1)

$$P(C_1, \boldsymbol{x}) = P(C_2, \boldsymbol{x}). \quad (5)$$

The quality condition described by (5) is our starting point to *choose the classifier status*.

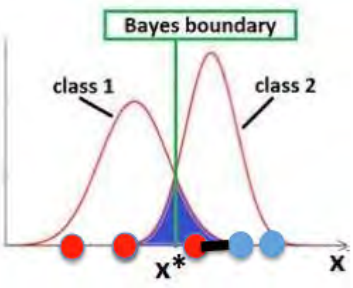

Figure 1: Two-class one dimensional example, and its Bayes boundary x^*. Perfect classification on $\mathcal{T}$ would lead to a boundary located somewhere on the thick black interval ($\neq x^*$).

Although the true joint probabilities are unknown unless infinite training data is available, we can check whether the estimated boundary satisfies the necessary and sufficient condition in (5) for the Bayes boundary. Indeed, (5) means that samples on B^* as a whole are perfectly ambiguous in terms of class label. Therefore assigning the opposite class label for all samples on B^* should not perturb the joint probability distribution. From now on, we will refer to such operation on the estimated boundary salient samples as "class label flip".

We expect the data distribution to be robust to the class label flip on B^*, hence an estimated boundary which barely changes after class label flip would correspond to a classifier status close to the Bayes status. Because B^* provides the highest generalization performance, we can thus select the classifier status which has the highest generalization performance. Two practical issues remain: the extraction of the set $S(\Lambda)$ of estimated boundary salient samples from $\mathcal{T}$ (Section 4.3), and the measure of the amount of perturbation on the data distribution induced by the class label flip (Section 4.4). Before entering these details, we summarize our proposed model selection procedure in Algorithm 1.

4.2 Method Overview

In the following, $g_j()$ refers to a general form of discriminant function, whose parameters are automatically determined by training procedure, but whose hyperparameters are yet to set through model selection. The proposed procedure simply measures the difference $\Delta(\Lambda)$ between the boundary estimated by the classifier

trained on the original training set, and the boundary estimated by the classifier trained on the training set after the class label flip. Such measure of difference is detailed in Section 4.4.

Algorithm 1: Selection of the optimal classifier status Λ^* for $g_j()$

Input: $g_j(), \mathcal{T}$
Output: $\Lambda^* = \arg\min_\Lambda \Delta(\Lambda)$
for *each hyperparameter setting* Λ **do**
  Train $g_j(;\Lambda)$ on $\mathcal{T}$: status "before" (b)
  Extract $\mathcal{S}(\Lambda)$
  Replace $\mathcal{S}(\Lambda)$ by $\hat{\mathcal{S}}(\Lambda)$ in $\mathcal{T}$
  Re-train $g_j(;\Lambda)$: status "after" (a)
  Measure $\Delta(\Lambda)$: changes on the boundary measured between (b) and (a)
end

4.3 A procedure to extract $S(\Lambda)$

The goal in this step is to find in $\mathcal{T}$ samples close to $B(\Lambda)$, since we only know condition in (5) which holds for samples that would lie exactly on B^*. By contrast to existing approaches such as [9], the proposed procedure uses an analytical expression of the estimated boundary for a general form of discriminant function. In this paper, we focus on the case of piecewise linear boundaries, however extension to the non linear case is also under consideration.

The proposed extraction method relies on the results from [11], where the geometric distance from a sample to the estimated boundary can be derived from the misclassification measure for piecewise linear models:

$$D_y(\boldsymbol{x};\Lambda) = \frac{d_y(\boldsymbol{x};\Lambda)}{||\nabla d_y(\boldsymbol{x};\Lambda)||}, \tag{6}$$

where $d_y(\boldsymbol{x};\Lambda)$ is the misclassification measure defined in Section 3.

In this case, a naive approach to select salient samples can consist in selecting training samples which are located within a tolerance region centered on the estimated boundary. However, defining a criterion to set the width of such region is not straightforward. Any width would inevitably retain training samples in some regions that are "not closest" to the estimated boundary, and in other regions reject training samples that are actually "closest" to the estimated boundary. To circumvent this issue, a more local extraction approach is required.

Our extraction approach consists of two steps. First, we project the entire $\mathcal{T}$ on the estimated boundary, using knowledge of the model as well as the geometric distance (for further details, please see Section 5). Second, we use projections as local anchors in the boundary to select surrounding salient samples, for example by applying the 1-nearest neighbor rule. Not only does such procedure require no hyperparameter, but it also provides as many representatives of $B(\Lambda)$ exactly on $B(\Lambda)$, as there are samples in $\mathcal{T}$, which will turn out useful to measure the amount of change on the estimated boundaries respectively before and after we flip the class labels of salient samples. Algorithm 2 summarizes the extraction procedure.

Algorithm 2: Extraction of $S(\Lambda)$

Input: $g_j(;\Lambda), \mathcal{T}$
Output: $S(\Lambda)$
for $\boldsymbol{x}$ *in* $\mathcal{T}$ **do**
  Project $\boldsymbol{x}$ on $B(\Lambda)$: $p(\boldsymbol{x};\Lambda)$
  if $NN(p(\boldsymbol{x};\Lambda)) \notin \mathcal{S}(\Lambda)$ **then**
    $\mathcal{S}(\Lambda) \leftarrow \mathcal{S}(\Lambda), NN(p(\boldsymbol{x};\Lambda))$
  end
end

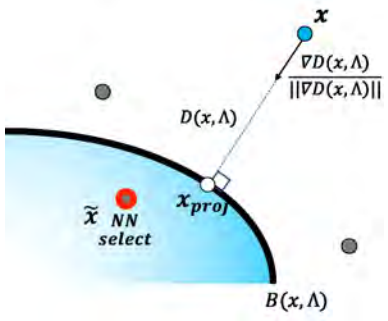

Figure 2: Projection on an arbitrary decision boundary using geometric margin definition, and salient sample selection. After sample $\boldsymbol{x}$ (blue) is projected on the boundary, we use $\boldsymbol{x}_{proj}$ (white) to select the salient sample $\boldsymbol{x}_{NNselect}$ (red) from the training samples.

4.4 A measure for the changes on the boundary

The purpose of this section is to quantify the amount of perturbation on the joint probability distributions $P(\boldsymbol{x}, C_y)(y = 1..J)$ induced by the class label flip performed around $B(\Lambda)$. Although estimating probability distributions can require specification and careful setting of a probabilistic model, we only need to measure a change in the joint probability distribution of the two classes. Such change is conveyed by a change of the estimated boundary between the two classes, hence we can focus on measuring a change in $B(\Lambda)$.

For convenience, we will refer to the state before (resp. after) the class label flip with index "b" (resp."a"). Using the parametric form of $B(\Lambda)$ expressed in (4) to define a difference measure is not straightforward. Instead we can represent the boundary by a set of samples close to the boundary, for example the boundary salient samples extracted in Section 4.3, or the projections on the boundary themselves. Several methods such as [12] provide a general measure

between two set of points. However simpler measures might be more suited to our specific case of a classification boundary, for example

$$\Delta(\Lambda) = |\sum_{\boldsymbol{x}_s \in \mathcal{S}(\Lambda)_b} |D_y(\boldsymbol{x}_s;\Lambda)_b| - \sum_{\boldsymbol{x}_s \in \mathcal{S}(\Lambda)_b} |D_y(\boldsymbol{x}_s;\Lambda)_a||, \tag{7}$$

and

$$\Delta(\Lambda) = \sum_{\boldsymbol{x}_s \in \mathcal{S}(\Lambda)_b} |D_y(\boldsymbol{x}_s;\Lambda)_b - D_y(\boldsymbol{x}_s;\Lambda)_a|, \tag{8}$$

where $\boldsymbol{x}_s$ refers to boundary salient samples before the class label flip. (7) measures a global amount of geometric distance of the salient samples to $B(\Lambda)_b$ and $B(\Lambda)_a$. Indeed, if the classifier model status is not properly set, then the class label flip is expected to globally produce a higher geometric distance of $S(\Lambda)_b$ to the re-estimated boundary $B(\Lambda)_a$. Although similar to (7), (8) defines a more sample-based measure, which might preserve more information about the classification process.

5 Experimental Evaluation

5.1 Experimental conditions

To evaluate our model selection procedure, we used four published two-class datasets presented by the UCI Machine Learning Repository [(*)]: the Breast Cancer, the Cardiotocography, the Ionosphere, and the Spambase datasets. Because we aim at a hyperparameter selection without any validation data, the data was solely split into a training set and a testing set.

Obviously, a flat trend on the classification accuracy curve for the testing set against the classifier model status would not require selection of a particular classifier model status, and conversely a clear maximum on the curve makes considerations and assessment of our model selection framework easier. Hence for each dataset, we tried several ratio between training and testing, and chose the ratio so that a maximum can be clearly seen on the classification accuracy curve for the testing set. Preparation of the datasets is summarized in Table 2.

Because our model selection approach strongly focuses on the decision boundary, we generated a synthetic two-dimensional dataset to visualize the selection of boundary salient samples, and have an intuitive understanding of the relationship between the estimated boundary and the Bayes boundary before and after the class label flip. The synthetic data was generated using Gaussian Mixture Models. The difficulty of the task was controlled through the degree of overlapping between the classes, the complexity of the shape of the Bayes boundary, and the size of the training subset. For similar reasons underlying our data splitting scheme, we tried to generate a relatively difficult task so that the optimal classifier model complexity is neither too low nor too high, and observe a maximum for medium model complexity. The resulting data called GMMu is displayed in Figure 4. The name comes from the way the data was generated and from its U shape.

Table 2: Benchmark datasets

Dataset	Training	Testing	Dimensions
Breast cancer	341	312	9
Cardiotocography	1000	1126	30
GMMu	250	17500	2
Ionosphere	200	151	34
Spambase	500	4101	57

The current experiments focused on prototypes-based classifiers trained by the Kmeans clustering procedure, in which case the geometric margin can be written as:

$$D_y(\boldsymbol{x};\Lambda) = \frac{-||\boldsymbol{p}_y - \boldsymbol{x}||^2 + ||\boldsymbol{p}_i - \boldsymbol{x}||^2}{2||\boldsymbol{p}_y - \boldsymbol{p}_i||}, \tag{9}$$

where for a given sample $\boldsymbol{x}$, $\boldsymbol{p}_i$ and $\boldsymbol{p}_y$ refer to the closest prototypes to $\boldsymbol{x}$ both in the wrong class and in the correct class.

5.2 Evaluation of the projection for prototype-based classifiers

To ensure exact calculation of the distance to the boundary, it is necessary to choose those prototypes carefully. Thus $\boldsymbol{p}_i$ and $\boldsymbol{p}_y$ must also be located directly beside the boundary and be adjacent one to the other, defining one segment of the piecewise linear boundary.

A natural idea to obtain projection for each sample $\boldsymbol{x}$ is to use the knowledge of $D_y(\boldsymbol{x};\Lambda)$ to compute the gradient projection

$$\boldsymbol{x}_{proj} = \boldsymbol{x} - D_y(\boldsymbol{x};\Lambda)\frac{\nabla D_y(\boldsymbol{x};\Lambda)}{||\nabla D_y(\boldsymbol{x};\Lambda)||}, \tag{10}$$

For linear models, this projection procedure gives reliable results. The projection however requires more careful devising for piecewise linear models applied to datasets with high dimensional features spaces. Indeed, since the boundary is defined for $k$means as a selection of bounded hyperplanes among Voronoi-type cells delimitation, there is a risk that a projection falls outside the bounds separating prototypes of different classes. In that case, the sample must be projected on the corner of two or more hyperplanes, and this projection depends on more than the two previously-defined prototypes. Problematic samples can be easily spotted by calculating misclassification measure after gradient projection: since defective projections are not projected on the boundary but land in different cells, their misclassification measure is not 0 (*cf.* Figure 3). This issue can then be corrected by solving of the op-

[(*)] http://archive.ics.uci.edu/ml/index.php

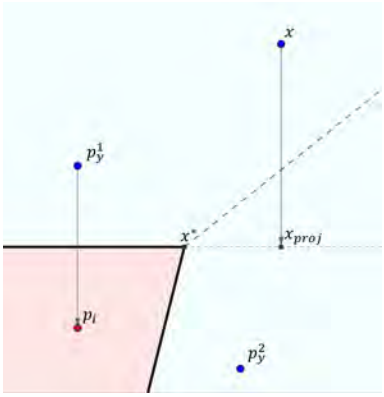

Figure 3: Projection on a piecewise linear decision boundary using geometric margin definition for a sample which projection falls outside the bounded separation between $\boldsymbol{p}_y$ and $\boldsymbol{p}_i$.

timization problem:

$$\begin{aligned} \min_{\mathbf{z}} \quad & ||\boldsymbol{x} - \mathbf{z}||, \\ s.t. \quad & \mathbf{z} \in \mathcal{L}, \end{aligned} \tag{11}$$

where $\mathcal{L}$ is the intersection of all hyperplanes intersecting on the corner we are projecting onto. The number of intersecting hyperplanes in $\mathcal{L}$ is dominated by the number $n_{features}$ of features for the chosen dataset and does not depend on the chosen K-means-clustering-based model as long as $n_{features} < K$. For models where $K < n_{features}$, space tesselation is less complicated and the number of intersecting hyperplanes in $\mathcal{L}$ can be dominated by K.

In our case, to alleviate computational costs, we chose only to keep K-means prototypes instead of calculating the cells and hyperplanes delimitations. Thus the number of intersecting hyperplanes in any location of the $n_{features}$-dimensional feature space is unknown. To solve the projection for a given sample $\boldsymbol{x}$ and a given set of prototypes $\mathcal{P}_K$ for the K-means-clustering-based model, in $n_{features}$-dimensional feature space, we thus proceed iteratively by solving the optimization problem (11) for $\boldsymbol{x}$ with a growing number of constraints (hyperplanes in $\mathcal{L}$) until the solution $\mathbf{z}^*$ verifies

$$D_y(\mathbf{z}^*; \Lambda) = 0. \tag{12}$$

Considering a hyperplane on the boundary is defined as the bisector between prototypes of both the wrong class and the correct class, it essentially means selecting a growing number of prototypes around sample $\boldsymbol{x}$ in set $\mathcal{P}_K$. The first two prototypes are chosen as defined for the gradient projection; for ensuing iterations, we select the prototype in $\mathcal{P}_K$ closest to the projection that verifies both adjacency to the boundary and to the previously chosen prototypes.

The method is summarized by Algorithm 3.

Efficiency of the projection is measured by the number of projected samples obtaining a misclassification measure of 0 (see Figure 5). In that regard, results presented in Figure 6 justify the use of an iterative algorithm, as the number of correctly projected samples increases with n_{iter}.

Algorithm 3: Sample projection on $B(\Lambda)$

Input: $\mathcal{T}, \mathcal{P}_K$
Output: $p(\boldsymbol{x}; \Lambda)$
foreach $\boldsymbol{x} \in \mathcal{T}$ **do**
  Select $\boldsymbol{p}_i$ and $\boldsymbol{p}_y$ next to the boundary
  Project $\boldsymbol{x}$
end
$n_{iter} \leftarrow 1$
while $\exists \boldsymbol{x}, D_y(p(\boldsymbol{x}; \Lambda)) \neq 0$ *or* $n_{iter} < \min(n_{features}, K)$ **do**
  $n_{iter} \leftarrow n_{iter} + 1$
  Select samples $\boldsymbol{x}$ that are not accurately projected
  foreach $\boldsymbol{x}$ **do**
    Select $\boldsymbol{p}_y^{n_{iter}}$ closest to $p(\boldsymbol{x}; \Lambda)$ next to the boundary
    Project $\boldsymbol{x}$ using the $n_{iter} + 1$ prototypes
  end
end

Bidimensional datasets can also be visualized to compare the boundary obtained through projection with the classifier estimated boundary, as presented in Figure 4. The more correctly projected samples we have, the better our knowledge of the estimated boundary becomes.

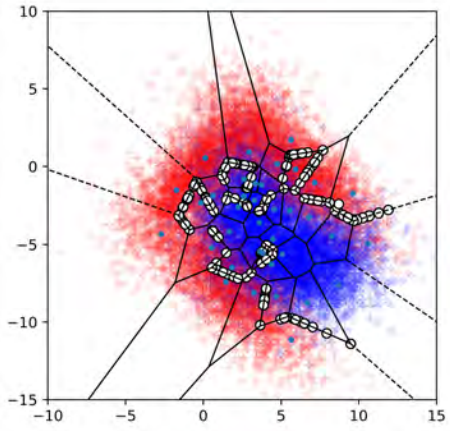

Figure 4: $K-$means clustering applied to two-class bidimensional distribution. Space is fragmented in Voronoi-type areas around $K-$means centroids. The decision boundary, on which training samples are projected (white circles), is defined as the line separating centroids of different classes.

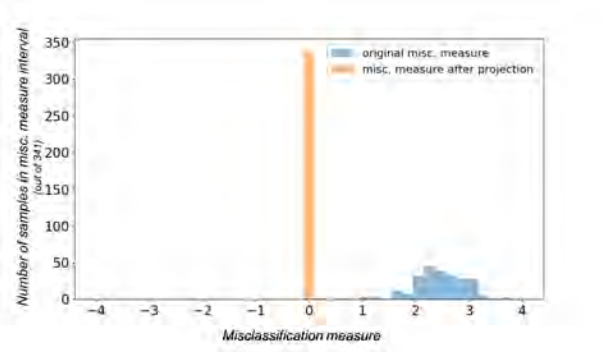

Figure 5: Distribution of the misclassification measure for the samples (blue) and their projections (orange) on the breast cancer dataset.

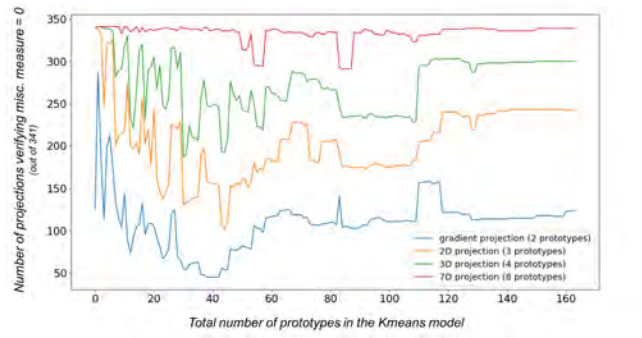

Figure 6: Number of projections obtaining a misclassification measure of 0 for n_{iter} = 1,2,3 and 7 on the breast cancer dataset depending on the chosen number of prototypes for $K-$means. For example 2D refers to a projection using $n_{iter} = 1$, which would lead to perfect projection if the data was two dimensional.

5.3 Evaluation of model selection for prototype-based classifiers

Figures 7 through 11 show the predictive performance of our model selection procedure on the $K-$means clustering, for which the hyperparameter to set is the number of prototypes per class K. To simplify the search, we set an equal K for all classes. For each dataset, the upper graph shows the classification accuracy on the testing, while the lower graph measures the perturbation measured on $B(\Lambda)$ when applying Algorithm 1 using (7), and the horizontal axis corresponds to K. Ideally the lower and upper graphs would show opposite trends. In particular a global minimum on the lower graph would correspond to a global maximum on the upper graph which we would select for optimal generalization performance. Note that at the time the results below were obtained using the projection using the gradient only in the boundary salient sample extraction step. Hence there is still possibility to improve the accuracy of the results using the improved projection procedure described in the previous section;

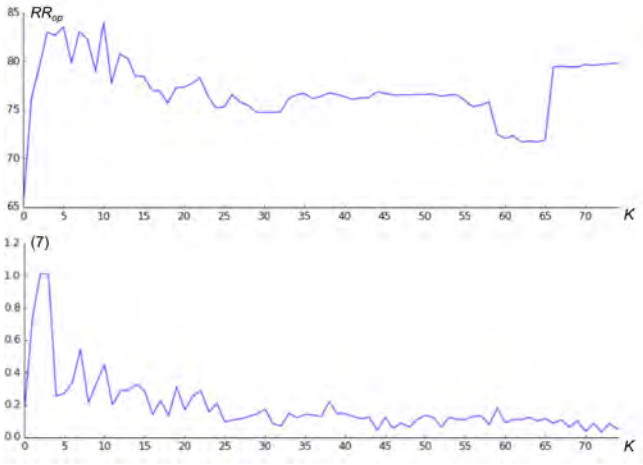

Figure 7: GMMu

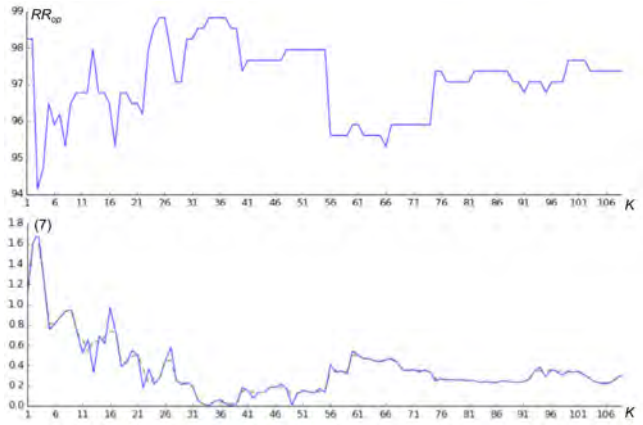

Figure 8: Breast cancer; RR_{op} : recognition rate on testing data

The results on the breast cancer dataset, the spambase, and the cardiotocography datasets are particularly encouraging, as (7) shows an opposite trend (monotonicity, local peaks) to the recognition rate on testing data. This points to the potential of the class label flip to measure the closeness of the estimated boundary to the ideal Bayes boundary.

The trends on ionosphere are also consistent, however compared to the recognition accuracy curve, (7) shows quite mild variations. This can be explained in part by the small number of testing samples available on ionosphere compared to the other datasets. As a result, changes on the estimated boundary affect the testing performance in a sharper way than can be observed based on the training set. By contrast, the procedure does not provide any predicting trend on the synthetic data GMMu. Although the pronounced "U" shape and the high overlapping between the two classes

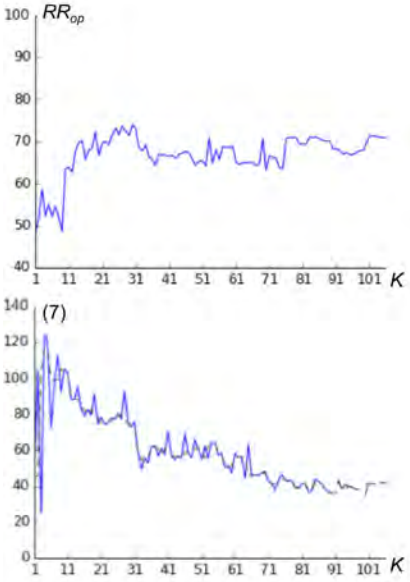

Figure 9: Cardiotography

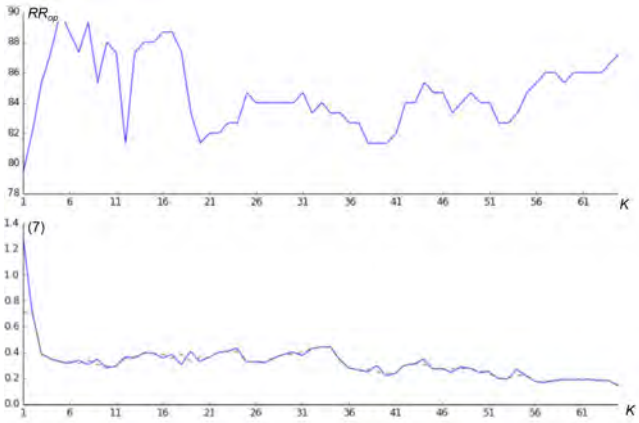

Figure 10: Ionosphere

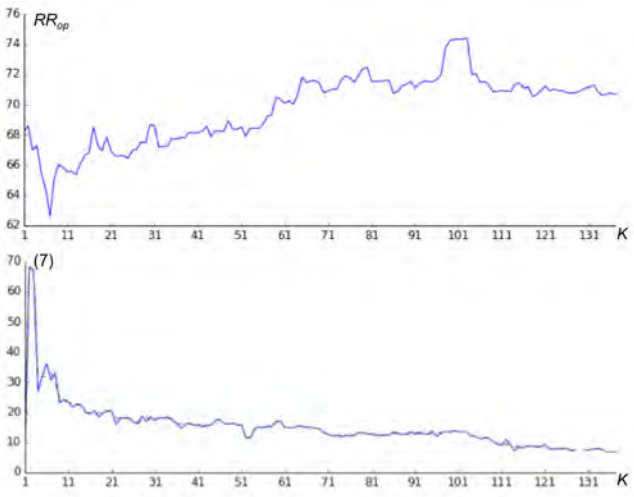

Figure 11: Spambase

might account for the difficulty of the task, further investigation about this case is necessary, for example by controlling the shape and overlapping of the synthetic data. Also the symmetric U shape embedding the two distributions might cause undesired robustness to the class label flip for low k.

Even though the results above are favorable overall, they depend on the metric used to quantify the amount of perturbation. To get further insight on the influence of the metric, we also reproduced the experiments using (8). The observed trend was quite disappointing compared to the results provided by (7). Such difference might be explained by the nature of our selection method, which tends to consider the entire distribution (and thus the entire estimated boundary) as a whole rather than focus individually on training samples. Hence global measures of perturbation are more natural in the current procedure.

The current perturbation measure depends on the number of selected salient samples, which varies from one candidate boundary to another. Therefore normalizing by the number of salient samples might improve the fairness of the comparison:

$$\Delta(\Lambda) = |\frac{1}{S_b}\sum_{s=1}^{S_b}|D_y(\boldsymbol{x}_s;\Lambda_b)| - \frac{1}{S_b}\sum_{s=1}^{S_b}|D_y(\boldsymbol{x}_s;\Lambda_a)||, \tag{13}$$

Alternatively, for each classifier status Λ, the set of projected samples on $B(\Lambda_b)$ can be used instead of S_b. Not only do all sets of projections contain the same number N of samples, but they also represent $B(\Lambda_b)$ more accurately.

6 Conclusion

In this paper we presented a novel way of choosing among different sets of trained classifier model parameters, the one that achieves higher generalization performance. Although the proposed procedure is applicable to a general form of classifier model, in a first stage we detailed the case of prototype-based classifiers. Experimental evaluation on real life data shows the potential for using the concept of class label flip to select the classifier model. However dependance of the framework on the perturbation metric, as well as uneven performance on our synthetic data requires further investigation. Hence future work will focus on how to develop the concept of class label flip, as well as how to reliably measure its effect on the joint probability distributions.

Acknowledgement

This work was supported in part by JSPS Grants-in-Aid for Scientific Research No. 26280063 and MEXT Supported Program " Driver-in-the-Loop" .

References

[1] B. Efron, R. Tibshirani, An Introduction to the Bootstrap, Chapman and Hall, 1993.

[2] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control 19*, pp. 716-722, 1974.

[3] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, 6(2), pp. 461-464, 1678.

[4] V. Vapnik, "The Structural Risk Minimization principle". In V. Vapnik (2nd ed). Statistical Learning Theory , pp. 219-268, 1998.

[5] D. Lewis, "Heterogeneous uncertainty sampling," *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 148-156, 1994.

[6] Y. Freund, R. Shapire, "Experiments with a new boosting algorithm," *Proceedings on the International Conference on Machine Learning*, pp. 148-156, 1996.

[7] M. Saar-Tsechansky, F. Provost, "Active Sampling for Class Probability Estimation and Ranking," *Machine Learning 54*, pp. 153-178, 2004.

[8] C. Bishop, "Introduction". In C. Bishop. Pattern Recognition and Machine Learning, pp. 40, 2006.

[9] C. Lee, D.A. Landgrebe, "Decision boundary feature extraction for nonparametric classification," *IEEE Transactions on Systems, Man, and Cybernetics 23(2)*, pp. 433-444, 1993.

[10] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery 2*, pp. 121-167, 1998.

[11] H. Watanabe, T. Ohashi, S. Katagiri, M. Ohsaki, S. Matsuda, H. Kashioka, "Robust and efficient pattern classification using large geometric margin minimum classification error training," *Journal of Signal Processing Systems 74(3)*, pp. 297-310, 2014.

[12] T. Eiter, H. Mannila, "Distance measures for point sets and their computation," *Acta informatica 34*, pp. 109-133, 1997.