Journal of Machine Learning Research 18 (2017) 1-35 Submitted 4/16; Revised 11/16; Published 2/17

Empirical Evaluation of Resampling Procedures for Optimising SVM Hyperparameters

Jacques Wainer WAINER@IC.UNICAMP.BR
Computing Institute
University of Campinas
Campinas, SP, 13083-852, Brazil

Gavin Cawley G.CAWLEY@UEA.AC.UK
School of Computing Sciences
University of East Anglia
Norwich, NR4 7TJ, U.K.

Editor: Russ Greiner

Abstract

Tuning the regularisation and kernel hyperparameters is a vital step in optimising the generalisation performance of kernel methods, such as the support vector machine (SVM). This is most often performed by minimising a resampling/cross-validation based model selection criterion, however there seems little practical guidance on the most suitable form of resampling. This paper presents the results of an extensive empirical evaluation of resampling procedures for SVM hyperparameter selection, designed to address this gap in the machine learning literature. We tested 15 different resampling procedures on 121 binary classification data sets in order to select the best SVM hyperparameters. We used three very different statistical procedures to analyse the results: the standard multi-classifier/multi-data set procedure proposed by Demšar, the confidence intervals on the excess loss of each procedure in relation to 5-fold cross validation, and the Bayes factor analysis proposed by Barber. We conclude that a 2-fold procedure is appropriate to select the hyperparameters of an SVM for data sets for 1000 or more datapoints, while a 3-fold procedure is appropriate for smaller data sets.

Keywords: Hyperparameters; SVM; resampling; cross-validation; k-fold; bootstrap

1. Introduction

The support vector machine (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995) is a powerful machine learning algorithm for statistical pattern recognition tasks, with strong theoretical foundations (Vapnik, 1998) and excellent performance in a range of real-world applications (e.g. Joachims, 1998; Furey et al., 2000; Fernández-Delgado et al., 2014). Perhaps the most common variant of the SVM uses the radial basis function (RBF) kernel, as recommended as a default approach in a popular guide to SVM (Hsu et al., 2010). The application of the RBF SVM to a classification problem requires the selection of appropriate values for two hyperparameters: a regularisation parameter, C, and a parameter governing the sensitivity of the kernel, γ. Given values for these two hyperparameters and the training data, an SVM solver, such as libSVM (Chang and Lin, 2011), can find the unique solution of the constrained quadratic optimization problem defining the SVM formulation and return a

classifier. We assume that the reader is familiar with the theory of SVMs and in particular of the SVM with the RBF (also known as Gaussian) kernel.

Unfortunately, the situation is less straightforward for model selection; there is no similarly principled means of optimising the hyperparameters. The simplest approach is to divide the data set into training and testing sets, and for each C and γ from a suitable set, select the pair that result in the SVM that when trained on the training set has lowest error rate over the corresponding test set. More commonly, resampling approaches, such as *cross-validation*, use multiple test/training sets in order to form a better model selection criterion from the available data.

This paper presents an empirical investigation of the effects of different resampling approaches to hyperparameter tuning on the generalisation performance of the final classifier. The investigation is focussed primarily on the SVM with an RBF kernel, but the main conclusions are repeated and validated for the linear and polynomial kernel SVM, as discussed in Section 5.

1.1 Resampling Approaches to Performance Evaluation

Performance evaluation is a key component of model selection procedures typically used in practical applications of support vector machines. Assume we have a sample of data, $\mathcal{G} = \{z_i = (\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ is a vector of attributes describing the i^{th} example and y_i is the corresponding class label; for binary classification tasks, $y_i \in \{-1, +1\}$. Resampling procedures provide a performance estimate based on repeatedly dividing $\mathcal{G}$ to form a *training set* and a *test set* (sometimes known as a *validation set*). More formally, in the i^{th} iteration of the resampling procedure, TR_i represents the training set and TE_i is the test set, such that

$$TR_i \cap TE_i = \emptyset \quad \text{and} \quad TR_i \cup TE_i \subseteq \mathcal{G}.$$

Let $\epsilon(\mathcal{B} \mid \mathcal{A}, C, \gamma)$ represent the error rate of an SVM trained on the training sample $\mathcal{A}$, using hyperparameter values C and γ, evaluated on the test set $\mathcal{B}$. The performance estimate provided by resampling methods is then typically the mean of the error rates obtained on the test set in each fold, i.e.

$$\text{Error}(C, \gamma) = \frac{1}{N} \sum_{i=1}^{N} \epsilon(TE_i \mid TR_i, C, \gamma),$$

where N is the number of iterations, or *folds*, of the resampling procedure. Different resampling *procedures*, such as *k-fold cross-validation*, *bootstrap* and *leave-one-out cross-validation* differ only in the way in which the data are partitioned to form TR_i and TE_i in each fold. Some common resampling procedures include:

- *k-fold cross-validation*: Partition $\mathcal{G}$ to form k disjoint sets F_j of approximately similar size, such that $\bigcup_j F_j = \mathcal{G}$. Then in each of the k iterations, a different set is used for testing and the others for training, i.e. $TE_i = F_i$ and $TR_i = \bigcup_{j \neq i} F_j$. In *stratified* cross-validation, $\mathcal{G}$ is partitioned such that each subset has a similar proportion of patterns belonging to each class. In repeated k-fold cross-validation, this procedure is performed repeatedly, with a different initial partitioning in each iteration.

- *Leave one out cross-validation*: This is the most extreme form of k-fold cross-validation, in which each set F_i consists of a single training pattern, i.e. $TE_i = \{z_i\}$ and $TR_i = \mathcal{G} \setminus \{z_i\}$.

- *Hold-out*: A single training set, TR_1, is defined, with size $p \times n$, and $TE_1 = \mathcal{G} \setminus TR_1$, where $p \in [0,\ 1]$. In stratified hold-out, the partitioning is performed such that TE_1 and TR_1 have a similar proportion of patterns of each class. In repeated hold-out resampling, this procedure is performed repeatedly with different random partitions of $\mathcal{G}$.

- *Bootstrap:* In each iteration, TR_i is obtained by sampling n items, with replacement from $\mathcal{G}$, and $TE_i = \mathcal{G} \setminus TR_i$.

- *Subsampling*: A hold-out resampling procedure where $TR_i \cup TE_i \subset \mathcal{G}$, that is, where only a subset of the available data set is used in each iteration. This is useful where a very large amount of data is provided.

Unfortunately, the names of these procedures are not well standardised. Appendix A discusses alternative names used for the concepts and procedures discussed in this paper.

Finally, resampling should be contrasted with *resubstitution*, a performance estimation method that uses the same set for both training the SVM and measuring its error rate. There are variations on the resubstitution procedure, where the data used to measure the error rate is the same data used in training, but they are given different weights (Braga-Neto and Dougherty, 2004).

1.2 Model Selection

The process of *model selection*, in the case of kernel learning methods, refers to the tuning of the kernel and regularisation hyperparameters in order to maximise generalisation performance. The generalisation error of a classifier can be expressed as an expectation over random samples, $\boldsymbol{z}$, drawn from the distribution $\mathcal{D}$ from which the training set was obtained,

$$\breve{\epsilon}(C, \gamma) = \mathbb{E}_{z \sim \mathcal{D}} \left[\epsilon(z \mid \mathcal{G}, C, \gamma) \right]$$

Ideally we would like to choose the hyperparameters C and γ so that $\breve{\epsilon}$ is minimized, that is:

$$C^*, \gamma^* = \underset{C,\gamma}{\operatorname{argmin}}\ \breve{\epsilon}(C, \gamma) \tag{1}$$

Unfortunately, the distribution giving rise to the data is generally unknown, and so we are unable to evaluate or directly optimise $\breve{\epsilon}$. The solution is to optimise instead an *estimate*, $\tilde{\epsilon}$, of the true generalisation error, $\breve{\epsilon}$. By far the most common approach is to optimise a resampling-based estimate. The estimate, $\tilde{\epsilon}$, of $\breve{\epsilon}$ for a particular resampling procedure rs is defined as:

$$\tilde{\epsilon}_{rs}(C, \gamma) = \frac{1}{N} \sum_{i=1}^{N} \epsilon(TE_i \mid TR_i, C, \gamma).$$

Given that a particular resampling procedure (rs) was selected, the choice of the SVM hyperparameters is governed by:

$$C^*_{rs}, \gamma^*_{rs} = \underset{C,\gamma}{\operatorname{argmin}}\ \tilde{\epsilon}_{rs}(C, \gamma).$$

It would be computationally infeasible to evaluate every possible combination of the hyperparameters, C and γ, so in general the search evaluates combinations from a finite set $\mathcal{S}$.

$$C^*_{rs}, \gamma^*_{rs} = \underset{C,\gamma\in\mathcal{S}}{\operatorname{argmin}}\ \tilde{\epsilon}_{rs}(C, \gamma). \tag{2}$$

Different model selection procedures adopt different methods to generate the set $\mathcal{S}$; they can be specified a-priori, as in the case of grid-search or random search, or successive elements of $\mathcal{S}$ can be generated according to the results obtained from evaluating existing elements, as in the case of Nelder-Mead simplex (Nelder and Mead, 1965), gradient descent (Chapelle et al., 2002) or other non-convex methods (Friedrichs and Igel, 2005; De Souza et al., 2006).

The choice of resampling procedure depends on two possibly conflicting criteria: firstly we would like to maximise generalisation performance, and secondly reduce computational expense. The error of a resampling estimate of generalisation consists of two components, *bias* and *variance*. The bias component represents the degree to which the estimate differs *on average* from the true value, over a large number of datasets of the same size as $\mathcal{G}$ sampled from the same underlying distribution, $\mathcal{D}$. The generalisation performance of classifiers tends to improve as the size of the training set increases. Resampling estimates therefore tend to have a pessimistic bias, systematically underestimating the generalisation performance of a classifier trained on $\mathcal{G}$, as in each fold a classifier is trained on only a subset of $\mathcal{G}$. The optimal hyperparameters for an SVM, particularly the regularisation parameter (C), can also demonstrate some degree of dependence on the size of the training set, which also leads to a bias in the hyperparameter estimates from resampling based model selection procedures. The variance component reflects the difference between the estimated and true values due to the particular sample of data on which the estimate was computed (and also due the random partitioning of the sample). As the model selection procedure directly minimises the resampling estimate, the presence of a non-negligible variance component introduces a risk of over-fitting in model selection (Cawley and Talbot, 2010), which results in suboptimal hyperparameter selection. Let us define the *true* excess loss (el) of a resampling procedure rs, as

$$el(rs) = \mathbb{E}_{\boldsymbol{z}\sim\mathcal{D}}\left[\epsilon(\boldsymbol{z} \mid \mathcal{G}, C^*_{rs}, \gamma^*_{rs}) - \epsilon(\boldsymbol{z} \mid \mathcal{G}, C^*, \gamma^*)\right],$$

where C^* and γ^* are the choices of optimal hyperparameters defined in Equation 1. The first important consideration in the choice of procedure is then to minimise the excess loss.

Unfortunately, the true excess loss is unknowable: Firstly we do not know the true values of γ^* and C^*, i.e. the values minimizing the generalisation error (1), indeed if these values were known, model selection using resampling methods (2) would be entirely redundant. Secondly, we do not know the true distribution of the data, $\mathcal{D}$, so we cannot perform the expectation involved in evaluating the generalisation error (1). The best we can do is to estimate the generalisation error using a finite test sample and compare the performance obtained using model selection based on a given resampling method against model selection

using some sensible baseline method, such as five-fold cross-validation. We therefore define the excess loss relative to five-fold cross-validation as the observed difference in the error rate on a fixed test set for classifiers trained with hyper-parameters adjusted so as to minimise a given resampling based performance estimate and so as to minimise the five-fold cross-validation estimate. This gives a relative indication of the improvement in generalisation performance due to the resampling method used to tune the hyper-parameters.

The second consideration in the choice of resampling procedure is the computational cost. The different resampling procedures have very different computational costs for the hyperparameter search. For example, for each possible pair of values C and γ, 10-fold cross-validation will require the fitting of 10 separate SVM classifiers, each with a training set of $0.9 \times |\mathcal{G}|$. For 5-fold cross validation, for each pair of hyperparameters, there will be 5 classifiers trained, each with a training set of size $0.8 \times |\mathcal{G}|$ — not only fewer support vector machines need be constructed, but each is fitted to a smaller training sample, with lower computational expense. If one is using a batch SVM solver, such as SVMlight (Joachims, 1999) or libSVM (Chang and Lin, 2011), one can assume that the learning time is at least quadratic in relation to the training size (Bottou and Lin, 2007).

A trade-off between these two criteria must be reached in selecting a suitable resampling estimator. We would like to minimise the number of folds to reduce computational expense, however the variance of resampling estimators is generally reduced by increasing the number of folds, resulting in improved hyperparameter estimates. Again, we would like to reduce the size of the training set to reduce training time, but this also tends to increase the variance of the estimator. If the training set is made smaller, the uncertainty in estimating the model parameters will be greater, and hence the test set error more variable. However this leaves more data available for the test set, which tends to reduce the variance of the estimator. In reaching a compromise, we must ensure that the training and test sets are both sufficiently large, and a sufficient number of folds are used, for the estimator to have a suitably low variance, whilst at the same time limiting computational expense so that the procedure remains practical. The goal of this research is to understand the balance of these two conflicting considerations.

We must point out that this paper evaluates different *fixed* resampling procedures, as opposed to adaptive resampling (Kuhn, 2014; Krueger et al., 2015) to select the SVM hyperparameters. Fixed resampling will use the same resampling procedure for each of the possible hyperparameter combinations being tested, by far the most common procedure. However there has been some recent work on adaptive resampling where, for example, the full resampling procedure is not performed for some of the hyperparameter combination if the results for tests sets so far indicate that one can be sure that the results are suboptimal (Kuhn, 2014), or where a small subsampling is used first for all hyperparameter combinations and as one becomes increasingly sure that some of the combinations have better results than others, subsampling with increasing larger training sets are tested (Krueger et al., 2015).

1.3 Related Literature

The problems of the computational cost of hyperparameter selection for the SVM, and the generalisation performance of the resulting classifier, have been discussed in the literature in

many forms. There are in general three alternatives to improve hyperparameter selection, which will be discussed separately:

- Use a different metric derived from the training set, typically a lower bound on generalisation performance, such as Xi-alpha, span and radius/margin bounds to select the hyperparameters (e.g. Vapnik and Chapelle, 2000; Keerthi, 2002; Joachims, 2000; Wahba, 1999).
- Use different search/optimisation procedures such as random search, Nelder-Mead simplex, non-convex optimization procedures, to select the combinations of C and γ to be evaluated during model selection (e.g. Bergstra and Bengio, 2012; Huang et al., 2007; Friedrichs and Igel, 2005; De Souza et al., 2006).
- Use different resampling procedures to estimate $\tilde{\epsilon}$ (e.g. Anguita et al., 2005, 2012).

The first approach, given above, optimises an alternative metric, rather than the error rate, ϵ, i.e.:

$$C^*, \gamma^* = \underset{C,\gamma \in \mathcal{S}}{\operatorname{argmin}} \, \phi(\mathcal{G}, C, \gamma)$$

The metric, ϕ, is sometimes called an *internal metric* or a model selection criterion and they are computed from the training set alone. Internal metrics, proposed in the literature include: the span bound (Vapnik and Chapelle, 2000); the radius/margin bound (Keerthi, 2002); the Xi-Alpha bound (Joachims, 2000); GACV (Wahba, 1999); and maximal discrepancy (Anguita et al., 2005). Duan et al. (2003) compare 5-fold cross-validation with some internal metrics, such as Xi-alpha and GACV as methods to select SVM hyperparameters on five data sets and find that the 5-fold has lower excess loss. Anguita et al. (2005) compare many cross-validation procedures and some internal metrics (maximal discrepancy and compression bound) for hyperparameter selection on 13 data sets, and find that cross-validation based procedures have lower excess loss than the internal metrics.

The second approach to improving the hyperparameter selection procedure usually fixes a particular model selection criterion, say 10-fold cross-validation, and proposes different means by which the C, γ are selected from the $\mathcal{S}$ set, and more generally, how the $\mathcal{S}$ set is dynamically computed given the error for the previously selected pairs C, γ. We will call this approach, the *hyperparameter search procedure*. Most search procedures are based on the fact that the error response surface, that is the error for each value of C and γ, is generally non-convex, and thus methods based on gradient descent can only be *guaranteed* to find a local minima. The standard, or most common search procedure is a simple *grid search*, where the set $\mathcal{S}$ is predefined, usually a geometrically spaced grid in both C and γ, i.e. the $\mathcal{S}$ points are taken from a uniform 2-dimensional grid in the $\log C \times \log \gamma$ space. This is the search procedure used in this research. Bergstra and Bengio (2012) propose a random search in the space $\log C \times \log \gamma$. Huang et al. (2007) propose selecting from fixed points in the $\log C \times \log \gamma$ space following the principles of uniform design (Fang et al., 2000). Keerthi and Lin (2003) discuss the asymptotic behaviour of the error surface for an SVM in the $\log C \times \log \gamma$ space and proposes a method by which the C is optimized in a 1D grid search for the *linear* SVM problem, which yields a $\hat{C}$ value, and the C and γ for the RBF SVM is selected using a 1D grid search on the line that satisfies $\log \gamma = \log C - \log \hat{C}$. Davenport et al. (2010) propose that one should see the non-convexity of

the error surface for different C and γ as a noisy convex surface, and proposes a filtered coordinate descent search where the "true value" of the error at a particular point C, γ is a Gaussian filtered value of the error rate in a neighbourhood of C, γ. Keerthi et al. (2007) define an approximation to the gradient of the error surface and proposes that a gradient descent search should be performed to select the optimal C and γ. Finally, many researchers have proposed the use of non-convex optimization procedures to select the optimal hyperparameters, including evolutionary algorithms (Friedrichs and Igel, 2005), and particle swarm optimisation (De Souza et al., 2006; Li and Tan, 2010)

The third approach, selection of different resampling procedures, is the one explored in this paper. We know of few papers that discuss relative merits of resampling procedures and the selection of hyperparameters: The closest to this research is Anguita et al. (2005). Anguita et al. (2005) compare 9-fold cross-validation (kf9), 10-fold cv (kf10), 10 repetitions of the bootstrap procedure (10xboot), 100 repetitions of the bootstrap (100xboot), leave-one-out (loo), and 70/30 hold-out on 13 data sets (along with two internal metrics) for hyperparameter selection. They use a different experimental procedure — a fixed test set (in our notation a fixed $\mathcal{F}$ data set) and thus they can estimate $\epsilon(\mathcal{F} \mid \mathcal{G}, C^*, \gamma^*)$ and the excess loss using the fixed test set. They average the relative excess loss across the data sets (for each procedure). They report that the k-fold procedures have lower relative excess loss, followed by the 100xboot, the 10xboot, the loo and the 70/30 in that order. They did not include any discussion on whether the differences can be considered negligible. Another result reported in the paper is that 100xboot has the lower estimate error for $\epsilon(\mathcal{F} \mid \mathcal{G}, C^*, \gamma^*)$, followed by loo, 10xboot, 70/30, and last kf.

2. Methods and data

This paper describes an experimental evaluation of different resampling-based model selection criteria, using 121 different data sets (described in detail in Section 2.2). The error rate of the SVM trained with the choice of hyperparameters selected using the different resampling procedures (discussed in Section 2.1) are evaluated using a 2-fold cross-validation. That is, each data set is divided into two halves, the different resampling procedures are used to select the hyperparameters using the first half, the SVM is trained in this first half and its error rate evaluated for the second half. The procedure is repeated using the second half as training set and the first half as test set. The estimate of the error rate for the resampling procedure (or more precisely the error rate of the SVM with hyperparameters selected by the resampling procedure) is the average of the two measured error rates. If i denotes a data set, i_a and i_b two halves of the data set, then the estimated error rate for a resampling procedure rs is

$$eer(rs, i) = \frac{\epsilon(i_b \mid i_a, C^*_{rs,a}, \gamma^*_{rs,a}) + \epsilon(i_a \mid i_b, C^*_{rs,b}, \gamma^*_{rs,b})}{2} \tag{3}$$

where $C^*_{rs,a}$ and $\gamma^*_{rs,a}$ are computed as described in Equation 2, where the i_a half of the data set i corresponds to $\mathcal{G}$ (we discuss the candidate set $\mathcal{S}$ below).

For the 112 smallest data sets, we used three different statistical methods to compare the estimated error rate for each resampling procedure with that of a baseline procedure, in this case the 5-fold cross-validation (details of the comparison methods in Section 3). We

used the 9 remaining large data sets, for which the experiments above would require too much computational time, to verify the conclusions derived from the experiments on the smaller data sets.

The time required to perform hyperparameter selection for each procedure, for each data set, was also recorded and the ratio with 5-fold cross-validation calculated. The time ratio was also averaged over all procedures to compute an "expected time ratio" of the resampling procedures in relation to that of 5-fold cross-validation.

For all procedures and data sets, the hyperparameter search procedure used an 11×10 grid search (the $\mathcal{S}$ set) following the ranges and steps popularized by `libsvm` (Hsu et al., 2010) i.e. $C = \{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$, and $\gamma = \{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$.

2.1 Resampling Procedures

The following resampling procedures are investigated:

1. 2-fold cross-validation (kf2)
2. 3-fold cross-validation (kf3)
3. 5-fold cross-validation (kf5)
4. 10-fold cross-validation (kf10)
5. 2 times repeated 5-fold (2xkf5)
6. 2 times repeated 10-fold (2xkf10)
7. 5, 10, and 20 times repeated bootstrap (5xboot, 10xboot, 20xboot)
8. 80/20 hold-out (80/20) — a training set of size approximately 80% of the original data, and test set of 20%, with similar proportion of classes
9. resubstitution (resub), training and testing in the whole data set
10. inverted 5-fold (invkf5): learning on a single fold, and testing on the remaining.
11. 20/20 hold out (20/20) — training and test sets of 20%
12. 5 times repeated 20/20 hold out (5x20/20)
13. 20/10 holdout (20/10)
14. 10/10 hold out (10/10)
15. 5 times repeated 10/10 hold out (5x10/10)

We describe procedures 3 to 8 as *large-training set* resampling procedures. If n is the size of the $\mathcal{G}$ set ($n = |\mathcal{G}|$) then the large-training procedures will perform training on sets ranging from $0.8 \times n$ for the kf5, 2xkf5, 80/20, to n for the bootstrap[1]. The 3-fold and 2-fold (item

1. Technically the size of the training set for the bootstrap depends on the details of the implementation. The set TR_i has size n, but some of the data are necessarily repeated. If one uses a naive implementation, the training set has size n, but if one makes use of weights for each data point to account for the repetition of the data, then the training size for the bootstrap is on average $0.632n$.

1 and 2 in the list) will be called *medium-training set* procedures, since they will perform training on $0.67n$ to $0.5n$ data points. The other procedures are called *small-training set* procedures, since the training size ranges from $0.1n$ (10/10 and others) to $0.2n$ (20/20 and others). Somewhat counter-intuitively, the *small-training set* procedures are more useful for *large* data sets, When it is the case that a small fraction of G can represent the underlying data-generating distribution, such resampling procedures can obtain good accuracy at a tiny fraction of the computational cost of resampling procedures that build larger training sets.

2.2 Data sets

The 121 data sets used in this study were collected from the UCI repository (Lichman, 2013), processed and converted by the authors of Fernández-Delgado et al. (2014) into a unified format. The data is derived from the 165 available at UCI repository in March 2013. From this set, they discarded 56, mainly because they were too large (number of data and/or number of features), or because they were not in a "common UCI format". Four further data sets were added to those from the UCI repository, and finally some data sets that had two or more definitions of "classes" were converted into different problems. For example the data set *cardiotocography* defined two classification problems, one with 3 classes and the other with 10 classes — each became a different data set. The names of the data sets are a simplification of the original UCI name. For each data set, categorical features were converted to numerical data, and each feature was *standardised* to have a mean of zero and unit standard deviation. No further pre-processing or feature selection was performed by Fernández-Delgado et al. (2014). We used the data generated by the authors of Fernández-Delgado et al. (2014), downloaded in November 2014. In 19 data sets the data were divided into a training and test set, but the test set was not standardized (in the data available in Nov 2014). In these cases we standardized the test set (independently of the training set) and joined the two into a single data set. Some of the 121 original data sets are multi-class. Since the SVM is essentially a binary classification procedure, we converted the multi-class problems into binary problems by ordering each of the original classes according to their names and alternately assigned the original classes to the new positive and negative classes. Finally, we used all the data sets with less than 10,000 data for the experiments, and used the 9 data sets with more than 10,000 data to verify the conclusions from the experiments.

The characteristics of all data sets used are reported in Appendix B. The tables report the size of each half of the data set. The average data set size is 2250 patterns (median 341.5, max 65030); the average number of features is 30 (median 17, max 263). The proportion of samples belonging to the positive class is displayed in the histogram in Figure 6 in Appendix B. The proportions are approximately normally distributed with mean 0.60 and standard deviation of 0.17.

3. Metrics and statistical procedures

We perform three different statistical analyses of the results. The first is the standard multi-classifier/multiple-data set comparison procedure proposed by Demšar (2006), where the multiple resampling procedures play the part of the multiple classifiers. The second

analysis computes confidence intervals for the excess loss of different resampling procedures in relation to kf5, and verify whether this interval is within a range of equivalence. If the confidence interval on the excess loss is smaller than a threshold the error rates of the two different resampling procedures are considered to be "equivalent". This second analysis addresses some of the general criticism of procedures based on the "null hypothesis significance test" (NHST) framework and is in consonance of what is usually referred to in the medical literature as "practical significance" (as opposed to mere "statistical significance"). The final form of analysis is the Bayes factor test proposed by (Barber, 2012, chapter 12).

3.1 Demšar procedure

Demšar (2006) proposed a method to determine whether a statistically significant difference exists in the performance of multiple classifiers on many data sets. In our case, the different resampling procedures used in hyperparameter optimisation play the role of the different classifiers. Demšar suggests using the Friedman test as an omnibus nonparametric paired test (the pairing is each data set). This test computes the p-value based on the null hypothesis that all classifiers are "equivalent" in terms of their true rankings. If the p-value is low enough one would reject the claim that the classifiers are "equivalent" and should proceed to determine *which* differences are "statistically significant" or not. In the case that all classifiers are being compared on an equal basis, Demšar (2006) proposes the Nemeyi post-hoc test. In our case, we are not interested in comparing all pairs of procedures, but only in comparing each resampling procedure with the baseline provided by kf5. In this case, Demšar (2006) proposes pairwise Wilcoxon signed-rank tests (the paired version of the non-parametric Wilcoxon test), but with the correction to the resulting p-values due to the multiple comparison. He proposes either the Bonferroni-Dunn procedure, or one of the step-up/down procedures of Holm, Hochberg, or Hommel.

3.2 Confidence interval on the excess loss

The Demšar's procedure falls under the null hypothesis significance testing framework, that is, one assumes that there are no differences among the resampling procedures and declare that there is a "statistically significant" difference if the probability of a difference in mean rankings at least as large as that actually observed is below a pre-determined threshold (usually 0.05). However, even if a "statistically significant" difference exists, the effect size may be sufficiently small that it is of no relevance to practical applications. For example, it may be that the difference between one procedure and another is a decrease of (say) 0.0001 in the error rate. This difference could be "statistically significant" but would be unlikely to be "practically significant" (Kirk, 1996). One possible way of determining if there is a practically significant variation between a resampling procedure and kf5 (for instance) for selecting SVM hyperparameters would be to determine a confidence interval for the excess loss and show that (with 95% confidence) the excess loss exceeds a *threshold of equivalence*.

The important aspect of the confidence interval procedure is the definition of a "equivalence" threshold — above what level of excess loss should a difference be considered a "practically relevant" change in the error rate? In this paper we will propose a method to determine this threshold of equivalence. Besides the resampling procedures described in Section 2.1, we also evaluated another procedure which was a repetition of the kf5 procedure

but with different folds; we call it the kf5bis procedure. The difference between the kf5 and kf5bis procedure was the random generator seed, and thus the "luck/bad luck" the experimenter has in creating the folds. Thus, in some sense the excess loss of the kf5bis procedure is a limit of equivalence, not necessarily because it is small, but because it is an excess loss one cannot further reduce, since it represents the effect of "luck" in the resampling procedure for selecting the hyperparameter. Thus, in this paper we use the mean excess loss of the kf5bis procedure as the threshold of equivalence. Table 2 in Section 4.2 reports among the other resampling procedures, the mean excess loss of the kf5bis procedure as -0.0031. Thus in this paper we will consider excess loss within the range $[-0.0031, 0.0031]$ as irrelevant. Simplifying, the NHST approach would compute the confidence interval for the excess loss and declare that the excess loss is statistically significant if the confidence interval does not cross the zero. A "practical significance" approach computes the same confidence interval for the excess loss, but declares that the excess loss is "irrelevant" if the interval is fully contained in the $[-0.0031, 0.0031]$ range.

3.3 Bayes factor

The third method to compare error rates across different data sets is the Bayesian analysis proposed by Barber (2012, chapter 12). The excess loss analysis above is based solely on the value of the error rate. The Bayesian method is based on both the magnitude of the error rate and the number of samples used in evaluating that error rate. For example, one will be more willing to assume that a classifier a which made 50 errors over 500 samples is equivalent to a classifier b that made 55 errors in 500, than if the first made 700 errors in 7000 samples, while the second made 770. Although the change in error rate is the same in both cases (0.10 versus 0.11), nevertheless, because of the larger test set, one is less sure that the two classifiers are equivalent in the second scenario. The Bayesian analysis measures the ratio between the probability that the classifiers are the equivalent versus the probability that they are not equivalent (given the data), and this *Bayes factor* should be much lower in the second scenario.

Given two classifiers evaluated on the same data set (or in our case the classifiers based on the different choices of hyperparameters derived using different resampling procedures), the method computes "How much evidence is there that the two samples of correct and incorrect predictions in the test set comes from independent multinomial distributions?" which is a possible rephrasing of the question "How much evidence is there supporting the contention that the two classifiers are performing differently?" Let us assume that classifier a when applied to the test set results in $e_a = \langle c_a, i_a \rangle$ where c_a is the number of correct predictions, and i_a the number of incorrect predictions, and similarly for classifier b. $P(H_{\text{same}} \mid e_a, e_b)$ is the posterior probability that the pairs of correct and incorrect results e_a and e_b come from the same (unknown) binomial distribution, which would indicate that both classifiers are equivalent. $P(H_{\text{indep}} \mid e_a, e_b)$ is the posterior probability that they came from independent distributions and therefore that the two classifiers are not equivalent (more precisely, it will be very unlikely that the two independent distributions are the same). The Bayes factor (BF) is the ratio of these two probabilities:

$$BF = \frac{P(H_{\text{same}} \mid e_a, e_b)}{P(H_{\text{indep}} \mid e_a, e_b)} \tag{4}$$

The larger the Bayes factor, the higher is the evidence towards the hypothesis that the two classifiers are equivalent. If $Z(x)$ is the beta function of a pair, and $u = \langle 1, 1 \rangle$, then the BF is calculated as

$$BF = \frac{Z(u)Z(u + e_a + e_b)}{Z(u + e_a)Z(u + e_b)} \tag{5}$$

Appendix C contains the derivation of this formula. In our case, we will consider 5-fold cross-validation as the baseline, and we will compare all other procedures to it, and thus we will calculate for each resampling procedure its Bayes factor in relation to 5-fold cross-validation.

We will report the $2 \log_e BF$ as defined in Equation 4. The reason to use the log is that the BF is a multiplicative factor, and for the mean and confidence interval calculations we need an additive factor. The use of the constant 2 is to follow the table of Kass and Raftery (1995) regarding the interpretation of strength of the evidence in favour of one or the other hypothesis, which is based on $2 \log_e$.

3.4 Computational Expense

Considering the time consumed for hyperparameter selection via the different resampling procedures, we ran all of the procedures for a single data set in sequence on a single core (of a multiple core machine). Different data sets were distributed to different cores of the same machine. We collected the total time to perform the resampling procedure to select the hyperparameters — the time to learn the final classifier with the optimal hyperparameters and to apply it to the other half of the data set was not included in the time measure. Again we use 5-fold cross-validation as baseline, and report the ratio of the execution time of each procedure and the 5-fold cross-validation execution time. The statistical calculations are performed with the log of the time ratio, which was then converted back to report the mean and confidence interval.

3.5 Statistical procedure

Besides the statistical tests performed by the Demšar procedure, we are interested in estimation of the mean excess loss, the log BF, and the ratio of execution time for each resampling procedure, in relation to the 5-fold. To evaluate the confidence interval of the mean of each of these measures, we use a bootstrap procedure with the "BCA" (bias- corrected and accelerated) evaluation method for the confidence interval (Efron, 1987). We use a 95% confidence level and $5,000$ replications of the bootstrap procedure.

3.6 Reproducibility

The 121 data sets, the program used to run the hyperparameter search, the raw results for each resampling procedure and data set, the program to analyse the results and generate the figures in this paper are available at `https://dx.doi.org/10.6084/m9.figshare.1359901`.

4. Results

In this section we report the results obtained from all experiments using the 112 smallest data sets, for which the experiments were computationally feasible.

4.1 Demšar procedure

We performed the Friedman test (as implemented in the libraries of the R programming language) with the following results: Friedman $\chi^2 = 382.8071$, d.f. $= 17$, p-value $<$ 2.2e-16. Thus, we reject the hypothesis that all resampling procedures are equivalent. We then performed the Wilcoxon signed-rank test for each procedure against kf5. The resulting p-values are given in Table 1. The first column is the average rank of each resampling procedure. The table includes kf5 as the first entry. The results indicate that there is no clear winner among the different resampling procedures. Notice that all average ranks for the medium and large training set procedures are similar (around 7) with the exception of the resubstitution (average rank of 15.2). Note that the average rank for the kf5 procedure itself is 7.2; thus procedures with average rank less than 7.2 are "better" then kf5, while a larger average rank indicates "worse" resampling procedures.

The second column of Table 1 displays the original p-value (from the Wilcoxon test) without any correction for multiple comparisons. The third column shows the result of Holm's correction, followed by the results of the Hochberg, Hommel, and Bonferroni-Dunn corrections, where p-values below 0.05 (which indicates a 95% confidence) are in bold. Again the results of these tests suggest that resubstitution, and the *small training set* procedures (except 5x20/20) are inferior to kf5, and the other methods are statistically equivalent, or more precisely, we cannot show that the other methods are statistically dissimilar. Table 1 shows that besides resub, all other large and medium training procedure are not statistically significantly different from kf5.

4.2 Excess loss

Table 2 reports the mean and 95% confidence interval of the excess losses for each of the resampling procedures. Figure 1 repeats the excess loss data of Table 2, removing the resubstitution data. The result of the mean excess loss for the kf5bis is -0.0031. As discussed, the absolute value of that excess loss is our threshold of equivalence; excess losses with absolute value lower than 0.0031 are considered in this paper as irrelevant.

4.3 Bayes factor

Table 3 reports the results of the Bayes factor calculations. Most of the results for the log BF are within the range from 5.5 to 6.1 in favour of the hypothesis that the error rate of each of the resampling procedures are the same as kf5. Following the scale of interpretation given by Kass and Raftery (1995), if the $2\log_e$ of the Bayes factor is in the range from 2 to 6, the evidence should be considered "positive", and from 6 to 10 "strong". Thus most of the results are in the direction of a positive evidence in favour that the resampling procedures results are the same as the 5-fold results.

A more specific threshold is derived from the kf5 and kf5bis results (first two lines in Table 3). kf5 is the result of calculating the Bayes factor for the kf5 procedure itself, so we

procedure	mean rank	original	Holm	Hochberg	Hommel	Bonferroni
kf5	7.2					
kf5bis	6.7	0.08	0.51	0.51	0.40	1.00
kf2	6.8	0.66	1.00	0.78	0.78	1.00
kf3	6.3	0.04	0.39	0.36	0.30	0.74
2xkf5	6.2	0.23	0.99	0.71	0.68	1.00
kf10	6.5	0.04	0.39	0.36	0.30	0.76
5xboot	7.2	0.78	1.00	0.78	0.78	1.00
10xboot	5.8	0.20	0.99	0.71	0.60	1.00
20xboot	5.8	0.06	0.41	0.41	0.35	0.99
80/20	9.3	0.01	0.14	0.14	0.14	0.24
resub	15.2	0.00	**0.00**	**0.00**	**0.00**	**0.00**
invkf5	7.8	0.24	0.99	0.71	0.71	1.00
20/80	8.8	0.00	**0.02**	**0.02**	**0.02**	**0.02**
20/20	10.6	0.00	**0.00**	**0.00**	**0.00**	**0.00**
5x20/20	10.3	0.01	0.14	0.14	0.12	0.21
20/10	10.3	0.00	**0.00**	**0.00**	**0.00**	**0.00**
10/10	11.1	0.00	**0.00**	**0.00**	**0.00**	**0.00**
5x10/10	10.3	0.00	**0.00**	**0.00**	**0.00**	**0.00**

Table 1: The result of the Demšar procedure comparing the resampling procedures. First column names the procedure, second display the average rank. The following columns display the p-value of the Wilcoxon signed rank test when compared with kf5: the original p-value, and then after the Holm, Hochberg, Hommel, and Bonferroni corrections

are sure that both procedures have exactly the same expected generalization error, and yet for this case, the log Bayes factor is on average 5.89. Thus, 5.89 is the mean theoretical maximum of the log Bayes factor for the data sets used in the experiment. Thus, the results in Table 3 are very close to the mean maximum. The kf5bis provides a limit to what is a "relevant" certainty. By design, the kf5bis results should be irrelevantly different than their kf5 counterpart — it could be that for one data set, the kf5bis result is sufficiently different to the kf5 result, but we are averaging over 112 data sets. So, we would like to make the claim that Bayes factor above 5.70, which is the average of the kf5bis, show that the differences are irrelevant to this problem. Figure 2 repeats the Bayes factor data Table 3 for data points around the 5.70 threshold above which one should consider that there is enough evidence that the resampling procedure is equivalent to the 5-fold.

4.4 Computational Expense

Table 4 reports the mean and confidence interval of the ratio of the time required for each resampling procedure and that for kf5. We would like to point out some anomalies or

procedure	mean	(95%CI)
kf5bis	−0.0031	−0.0073 , 0.0002
kf2	−0.0012	−0.0056 , 0.0024
kf3	−0.0031	−0.0094 , 0.0002
2xkf5	−0.0014	−0.0043 , 0.0019
kf10	−0.0020	−0.0054 , 0.0023
5xboot	0.0014	−0.0021 , 0.0053
10xboot	−0.0022	−0.0065 , 0.0009
20xboot	−0.0018	−0.0047 , 0.0016
80/20	**0.0053**	−0.0003 , 0.0115
resub	**0.1354**	0.1105 , 0.1632
invkf5	0.0009	−0.0038 , 0.0044
20/80	**0.0074**	0.0009 , 0.0133
20/20	**0.0191**	0.0102 , 0.0323
5x20/20	**0.0052**	−0.0012 , 0.0162
20/10	**0.0217**	0.0118 , 0.0357
10/10	**0.0255**	0.0154 , 0.0372
5x10/10	**0.0160**	0.0083 , 0.0274

Table 2: Excess loss in relation to kf5 for all the resampling procedures. The second column is the mean excess loss (in relation to kf5) of each procedure; the third and fourth columns are the 95% confidence interval for the mean. In bold, the mean excess losses that are of practical significance.

unexpected results regarding the computational expense. The first one is that resubstitution is slower than kf5. A second one is that the 5-times repeated 20/20 and 10/10 are only three times as expensive as the corresponding non-repeated procedure. We do not know how to explain these results, but it is possible that they are also due to interference between different runs; as discussed in Section 3.4 the many cores of the machines were executing the experiments in different data sets at the same time.

5. Discussion

The results for the medium-training procedures (kf2 and kf3) are a welcome surprise. The Demšar analysis shows that they are not significantly different to those obtained using kf5. Furthermore, the excess loss confidence interval shows that their excess loss is mostly within the irrelevance threshold, and the Bayes factor indicates that there is good evidence that their performances are equivalent to that of kf5. However, these two procedures select the SVM hyperparameters in 33% and 55% of the time of the kf5 procedure — a useful computational saving.

Most of the large-training procedures have a negative mean excess loss, that is, they select hyperparameters that result in slightly lower error rates for the future data. The two

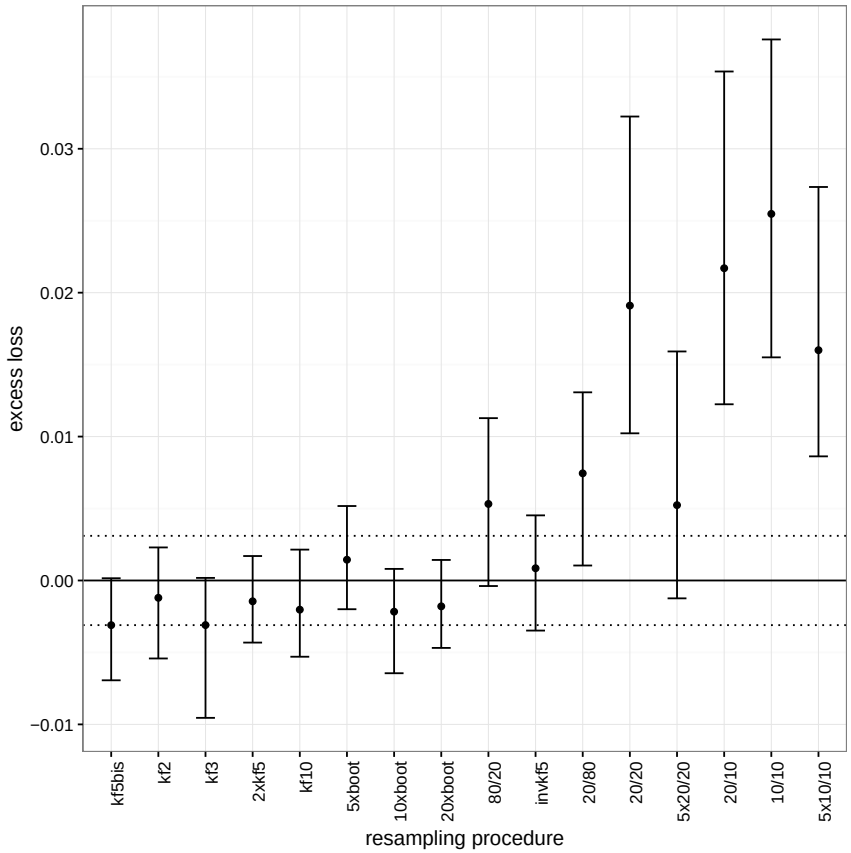

Figure 1: Excess loss in relation to kf5 for the resampling procedures (resubstitution not included). The dotted line indicates the limit for what is considered an irrelevant change in the loss.

exceptions are the 80/20 and the resub procedures discussed below. But for all procedures that result in a better selection of the hyperparameters the mean excess loss it is still within our range of equivalence to kf5. That is, even using computationally more expensive procedures such as 20- and 10-times bootstrap, 2 times repeated 5-fold, on average it is not likely that there will be relevant changes in the final error rate of the classifier. This result is consistent with the Demšar analysis, which show no statistically significant differences between these procedures and kf5. The two large-training resampling procedures that have positive excess loss are the 20% hold-out (80/20) and the resubstitution estimator. The result for resubstitution is widely known—the use of the same data for training and testing causes severe overfitting—but we have shown that the overfitting is also severe in regards to hyperparameter selection.

The small-training procedures all incur positive excess losses, to varying extents, and thus are in general "worse" than kf5, but the inverse-kf5 and the 20/80 holdout have the lowest mean excess loss. The excess loss for the 20/20 procedures (sample 20% for training and 20% for testing), and for the 20/10 and 10/10 procedures are very significant. We

procedure	mean	(95%CI)
kf5	5.89	5.52 , 6.33
kf5bis	5.70	5.33 , 6.11
kf2	5.61	5.24 , 6.05
kf3	5.73	5.35 , 6.16
2xkf5	5.74	5.37 , 6.19
kf10	5.67	5.30 , 6.11
5xboot	5.67	5.29 , 6.11
10xboot	5.73	5.35 , 6.16
20xboot	5.72	5.34 , 6.15
80/20	5.32	4.88 , 5.75
resub	−150.53	−274.41 , −81.73
invkf5	5.48	5.05 , 5.94
20/80	5.19	4.69 , 5.70
20/20	3.14	−1.49 , 4.45
5x20/20	4.93	3.85 , 5.52
20/10	−2.85	−30.38 , 3.26
10/10	2.97	1.26 , 4.05
5x10/10	4.51	3.45 , 5.16

Table 3: $2\log_e$ of the Bayes factor of the hypothesis that the resampling procedure results are the same as the kf5 results versus the hypothesis that they are independent. The numbers in the table are the mean difference of the log of the Bayes factor between the procedures indicated in the left column and the kf5. The last two columns are the 95% confidence interval for the mean.

conclude from these observations, that there seems to be no real gain in using resampling procedures more costly than 5-fold cross-validation. On the other hand, the kf2 and kf3 procedures seem to achieve a similar result to kf5, with lower computational costs.

An important issue is whether the results for the equivalence, for practical purposes, of the kf2, kf3 and kf5 procedures carries forward to larger and higher dimensional data. We tested the kf2 and kf3 on the 9 largest data sets. The results are in Table 5. The table shows the excess loss in relation to kf5, the log BF of both procedures and the kf5 log BF (which indicates the maximum value) and the time ratios in relation to kf5. With the exception of the nursery data set, all excess losses are within our limit or irrelevance, and the BF are close to the maximum, a very strong evidence that these procedures perform equally well for larger size data sets. Figures 3 and 4 show the values of the excess loss as a function of the data set size and the number of dimensions of the data. For data sets with sizes beyond the examples tested, one can look at the trend of excess loss as a function of the data set size in Figures 3 and 4: there is a compelling trend of diminishing excess loss as the data set size increase. Therefore we are very confident that one can safely use kf2 or kf3 to tune

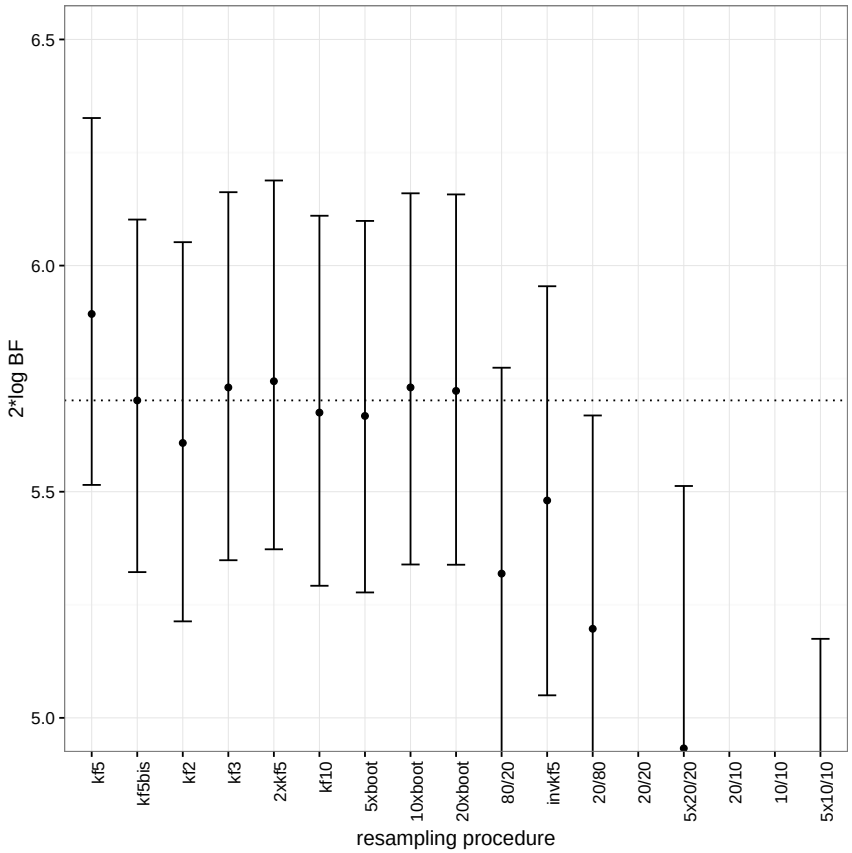

Figure 2: Bayes factor of the ratio of the probabilities that the resampling procedure and the 5-fold are equivalent. Data around the 5.7 threshold, above which one should consider that there is enough evidence to claim the equivalence.

SVM hyperparameters in large data sets. Note that as the size of the dataset increases, the bias and variance of most resampling procedures tend to decrease, so this is entirely in accord with a-priori intuition.

The evidence of a decreasing trend for larger dimensionality is less compelling for the kf2 than it is for the kf3. For one data set with larger dimensionality, the excess loss of kf2 is outside our range of irrelevance, but that is not true for the kf3. It should be noted though that the RBF kernel is often unsuitable for high-dimensional datasets and a linear kernel is often preferable in such cases (Hsu et al., 2010).

Another important issue is whether the practical equivalence of the kf2 and kf3 to kf5 is also valid for an SVM with other kernels. We evaluated the excess loss of kf2 and kf3 on the 112 smallest datasets on SVM with polynomial and linear kernels. The results are displayed in Table 6. The data are also displayed in Figure 5. The results are not too different than the excess loss for the RBF SVM, with the exception of a somewhat higher excess loss for the kf2 for linear SVM. So we believe that the conclusions of practical equivalence of the kf2

procedure	mean	(95%CI)
kf2	0.35	0.31 , 0.39
kf3	0.54	0.52 , 0.57
2xkf5	1.95	1.90 , 2.01
kf10	2.39	2.32 , 2.45
5xboot	1.24	1.20 , 1.28
10xboot	2.28	2.21 , 2.35
20xboot	4.43	4.28 , 4.56
80/20	0.31	0.30 , 0.32
resub	1.45	1.37 , 1.52
invkf5	0.46	0.40 , 0.51
20/80	0.22	0.19 , 0.25
20/20	0.11	0.09 , 0.13
5x20/20	0.33	0.27 , 0.39
20/10	0.17	0.14 , 0.20
10/10	0.09	0.07 , 0.12
5x10/10	0.28	0.22 , 0.35

Table 4: Time ratio between the resampling procedure and the kf5.

db	size	kf2 loss	kf3 loss	kf2 BF	kf3 BF	kf5 BF	kf2 time	kf3 time
pendigits	5496	−0.0004	0.0002	11.8	11.7	11.9	0.56	0.40
nursery	6480	0.0014	0.0014	10.2	10.2	10.8	0.21	0.44
magic	9510	−0.0004	0.0018	8.8	8.7	8.8	0.20	0.40
letter	10000	0.0007	−0.0000	10.4	10.6	10.6	0.20	0.39
chess-krvk	14028	0.0073	0.0000	6.6	8.8	8.8	0.13	0.31
adult	24421	−0.0002	−0.0008	9.6	9.6	9.6	0.14	0.31
statlog-shuttle	29000	−0.0001	0.0	15.5	15.6	15.6	0.32	0.45
connect-4	33778	0.0012	0.0012	9.9	9.9	10.1	0.27	0.36
miniboone	65032	−0.0002	−0.0005	11.3	11.2	11.4	0.17	0.52

Table 5: kf2 and kf3 excess loss in relation to kf5, the 2*log BF of kf2, kf3 and kf5 (which shows the maximum value of the BF) and the time ratio for kf2 and kf3

and kf3 resampling procedures in relation to the kf5 resampling to select hyperparameters are also valid for the linear and polynomial SVM.

5.1 Limits of this research

The strength of the conclusions of this research rests on the assumption that the set of 121 data set used are a good sample of "real life" data sets. There are limits to this assumption. Fernández-Delgado et al. (2014) did not include large data sets (large number of data points

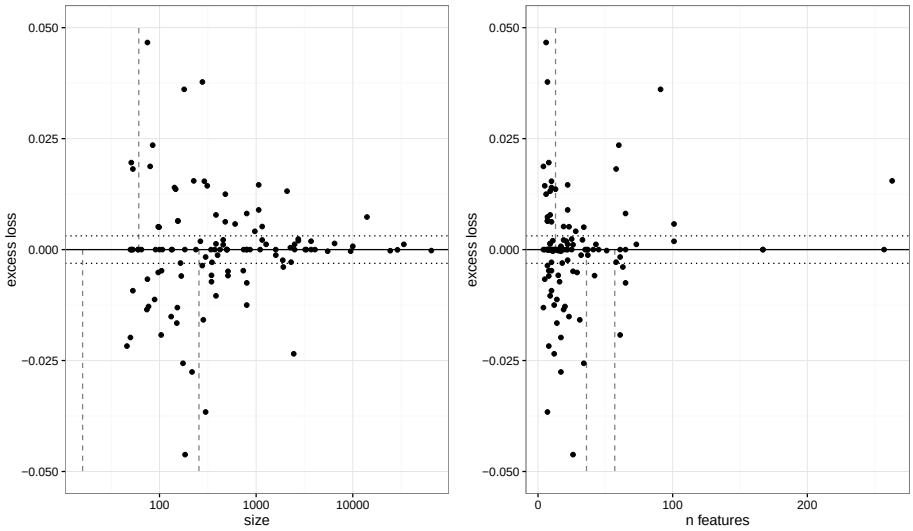

Figure 3: The excess loss of the kf2 procedure in relation to the dataset size and dimensionality. The dotted horizontal line is our threshold of irrelevance. The dashed transparent vertical line indicates that there is a data point at that position that was clipped from the graph.

Kernel	procedure	mean	min	max
Polynomial	kf2	**0.006**	0.002	0.010
	kf3	0.001	-0.003	0.004
Linear	kf2	0.002	0.000	0.005
	kf3	0.002	0.000	0.007

Table 6: Excess loss for the kf2 and kf3 procedures on SVm with Polynomial and Linear kernels.

or large number of features). Thus, our sample does not reflect large data sets (especially text-classification datasets which also have a high dimensionality). Practitioners working with these kinds of data sets are advised to check our conclusions before following them.

It has been suggested that the outer 2-fold procedure to estimate $\breve{\epsilon}$ was probably too noisy since kf2 is a high variance estimator. Appendix E shows that even using a 5xkf2 (on the 42 data sets with at most 400 data) the results do not change significantly — but the confidence intervals are somewhat reduced. Therefore, even if a lower variance estimator of $\breve{\epsilon}$ was used, we do not believe the results would change substantively.

This research assumed that kf5 was the "standard" resampling procedure for hyperparameter selection, and compared all other procedures against kf5. The many results in the paper show, among others, kf2 and kf3 are equivalent from a practical point of view to kf5. But that might not be the case for more costly, low variance procedures such as kf10 or 20xboot. Appendix F shows the result of the Nemenyi test for comparing all resampling

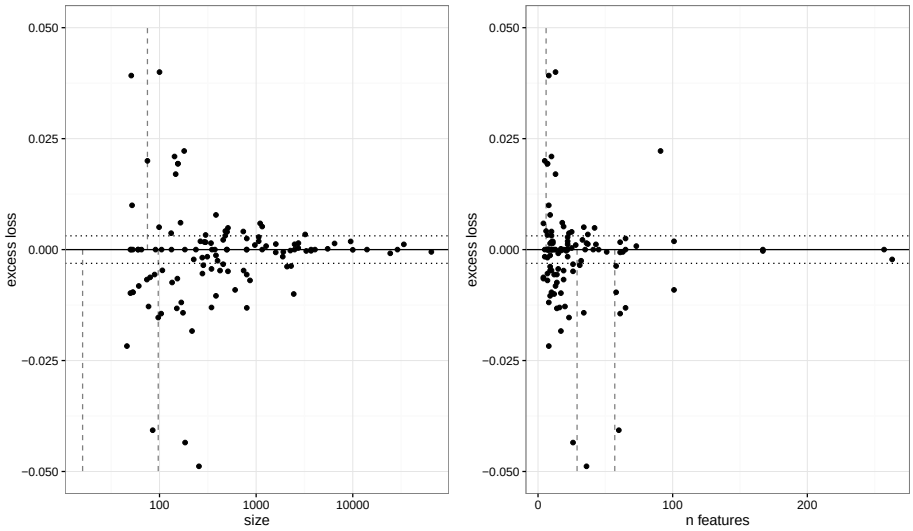

Figure 4: The excess loss of the kf3 procedure in relation to the dataset size and dimensionality. The dotted horizontal line is our threshold of irrelevance. The dashed transparent vertical line indicates that there is a data point at that position that was clipped from the graph.

procedures against each other. None of the large-training and medium-training procedures are statistically significantly different from each other, with the exception of 80/20 and resub which were also not equivalent to kf5.

5.2 Methodological discussion

We believe that besides the important result of a strong suggestion of using 2-fold or 3-fold cross validation to select the hyperparameters of an SVM this research also opens an important methodological discussion on at least three topics. The first is the use of a Bayesian framework for the comparison of classifiers. The second is the use of concepts of "practical irrelevance" or "practical equivalence" in machine learning, and in particular the use of the kf5bis to discover the threshold of equivalence. The final issue, is whether it is methodologically sound to average excess losses. The Bayes factor approach resulted in a metric that is not very sensitive. Most of the results in Table 3 are similar to each other and almost all are in the range of positive evidence, but as we have shown using the results of kf5, this seems to be the strongest evidence possible in this problem. Our decisions regarding the BF should be reviewed by future researchers in search of a more sensitive metric. For example, we used only the number of correct and incorrect predictions in the Bayesian calculations. Barber (2012) implicitly suggest both these measures and also using the 4-tuple of true positives, true negatives, false positives, and false negatives.

We must explain why we used two other analysis procedures beyond the usual Demšar proposal. As we discussed, the Demšar (2006) procedure falls within the NHST framework, and other empirical sciences realized that NHST are rarely the correct tool to analyse data

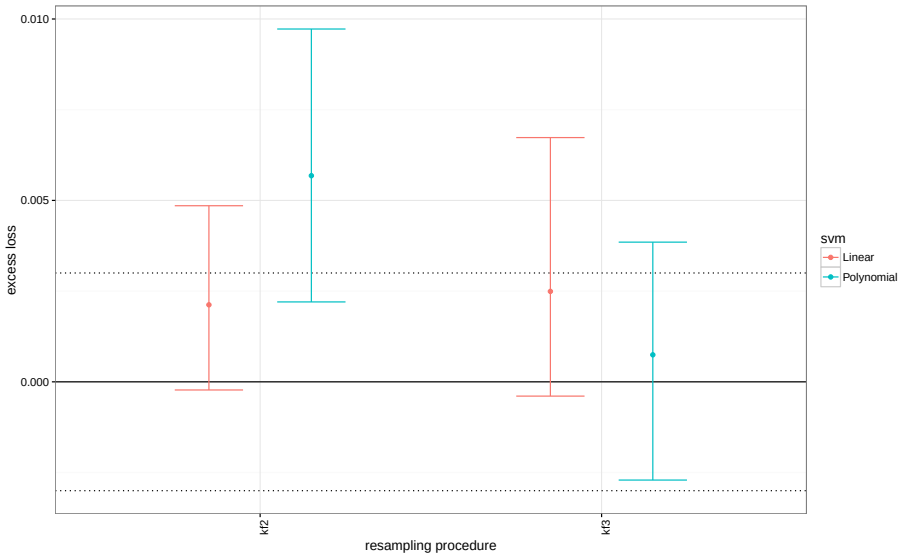

Figure 5: The excess loss of the kf2 and kf3 procedures for Linear and Polynomial SVM, measured on the 112 smallest dataset. The dotted line is the threshold of practical equivalence.

(Gardner and Altman, 1986; Thompson, 2002; Woolston, 2015). Significance tests can tell us that two samples are different enough that it is unlikely that they came from the same population, but not whether the difference matters for any practical purpose. Alternatively, significance tests can tell us that there is not enough evidence to make the claim that the two samples came from the same population, which in no way means that the two samples are "the same." Our proposal of using a confidence interval on an effect size and a measure of practical equivalence is one possible direction, as is our usage of Barber (2012) Bayesian analysis.

Another methodological proposal was to use the kf5bis (a different random selection for the kf5) as a way of determining a threshold of equivalence. The differences between kf5 and kf5bis should on average over many data sets be irrelevant, either because they are too small to matter or because there is no way to reduce them.

The final discussion is whether computing the mean (and confidence interval) of a excess loss is a sound procedure. An argument against it is that error rates (and therefore differences in error rates) cannot be compared across different data sets. One can argue that an improvement of 0.05 in the error rate would be a substantial improvement if the error rate was originally 0.08; however an improvement of 0.05, where the original error rate was 0.28, is rather less significant. The same number (0.05) seems to mean different things in these two situations, and therefore adding those two numbers (to compute the average) does not seems a reasonable operation. We believe that the source of this problem is that the variance of the error rates across different resampling procedures, for example, varies as a function of the error rate itself. If the variance of error rates for low error rates is low, then 0.05 is a considerable magnitude (in standard deviations) for an error rate of 0.08. The

variance should be much larger for an error rate of 0.28, which in turn means that 0.05 is not such a large change in that case. But as Appendix D shows, the variance of the excess loss (for all procedures) does not increase as the error rate (for the kf5 procedure) increases. In fact the variance can be seen as constant across the different error rates, and that would allow one to compute the average of the excess losses.

6. Conclusions

This research has evaluated the impact of different resampling procedures in the selection of SVM hyperparameters, by comparing 17 different procedures in 121 data sets. The conclusion is that the 2-fold procedure should be used in data sets with more than 1000 points. In these cases the user may expect a difference of -0.0031 to 0.0031 in the error rate of the classifier if a 5-fold procedure was used, which we believe is the limit of what one should consider an irrelevant change in the classifier error rate. For smaller data sets, we could not detect any significant difference (on average) between 5-fold and computationally more costly procedures such as 10-fold, 5 to 20 times repeated bootstrap, or 2 times repeated 5-fold. We believe that a 3-fold is appropriate for smaller data sets.

Acknowledgments

The authors would like to thank the anonymous reviewers and the editor for their detailed and constructive comments that have significantly improved this paper, and Nicola Talbot for her careful proof-reading and copy-editing. The first author would like to thank a Microsoft Azure Research Award that allowed to run some of the experiments in this research in the Azure cloud.

Appendix A. Terminology

resampling vs cross-validation. What we called *resampling* is frequently called *cross-validation*. We decided to use *resampling* instead of the most common term *cross-validation*, because we understand that the "cross" term in *cross validation* implies a subset that is used in training and then again in testing, as in the k-fold procedure. That is not necessarily true for the bootstrap procedure — there is no subset that is by construction used in the training and then in the testing. Resampling procedures are also called *out-of-sample* techniques (Anguita et al., 2012).

k-fold also known as *k-fold cross validation* or *cross-validation* by itself (Kuhn et al., 2014).

hold out is sometimes called *leave group out* (Kuhn et al., 2014) or *split sample* (Molinaro et al., 2005).

model selection (Cawley and Talbot, 2010) is also called *selection of hyperparameters* or *hyperparameter optimization* (Bergstra and Bengio, 2012) or *hyperparameter tuning* (Duan et al., 2003)

bootstrap: in the version of bootstrap described here the error rate is measured solely on the test set which contains only the data not included in the training set. Two other variations of bootstrap are commonly used: the .632-bootstrap (Efron, 1983) and the .632+-bootstrap (Efron and Tibshirani, 1997). For these methods the error estimate is calculated using not only the test set, but also the error rate of the training set itself.

Appendix B. Data sets

Table 7 lists the characteristics of all data sets, ordered by size. The name of the data set is the same as those used in Fernández-Delgado et al. (2014). The size refers to one half of the data set (please refer to section 2 regarding the methodology to compute the excess log loss). The *nfeat* column is the number of features or number of dimensions of the data in the data sets; *ndata* the number of patterns; *prop* the proportion of data in the positive class. *Notes* has the following values:

- r the data set was removed from all analysis because all cross validation tests have less than 5 examples of a particular class
- s some of the experiments did not run because their test sets had less than 5 examples of a particular class
- m the data set describes a multiclass problem, and so was converted to a binary problem using the procedure discussed in section 2
- l the data set has more than 5000 patterns so it was used in the large data set validation.
- t the data set was originally divided into a test and train set, where the test set was not standardized (see section 2).

Figure 6 displays the histogram of the proportion of the positive class.

Table 7: The data sets

data set	nfeatures	ndata	prop	note
trains	30	5	0.40	r
balloons	5	8	0.50	r
lenses	5	12	0.83	r
lung-cancer	57	16	0.56	r
pittsburg-bridges-SPAN	8	46	0.52	m
fertility	10	50	0.96	s
zoo	17	50	0.62	m
pittsburg-bridges-REL-L	8	51	0.82	m
pittsburg-bridges-T-OR-D	8	51	0.86	
pittsburg-bridges-TYPE	8	52	0.62	m
breast-tissue	10	53	0.53	m
molec-biol-promoter	58	53	0.49	
pittsburg-bridges-MATERIAL	8	53	0.94	ms
acute-inflammation	7	60	0.52	
acute-nephritis	7	60	0.67	
heart-switzerland	13	61	0.38	m
echocardiogram	11	65	0.74	
lymphography	19	74	0.46	m
iris	5	75	0.72	m
teaching	6	75	0.64	m
hepatitis	20	77	0.22	
hayes-roth	4	80	0.57	mt
wine	14	89	0.60	m
planning	13	91	0.74	
flags	29	97	0.47	m
parkinsons	23	97	0.25	
audiology-std	60	98	0.64	mt
breast-cancer-wisc-prog	34	99	0.77	
heart-va	13	100	0.52	m
conn-bench-sonar-mines-rocks	61	104	0.54	
seeds	8	105	0.64	m
glass	10	107	0.40	m
spect	23	132	0.67	mt
spectf	45	133	0.19	t
statlog-heart	14	135	0.53	
breast-cancer	10	143	0.69	
heart-hungarian	13	147	0.63	
heart-cleveland	14	151	0.75	m
haberman-survival	4	153	0.75	
vertebral-column-2clases	7	155	0.68	

continued in the next page

Table 7: The data sets (continued)

data set	nfeatures	ndata	prop	note
vertebral-column-3clases	7	155	0.67	m
primary-tumor	18	165	0.68	m
ecoli	8	168	0.64	m
ionosphere	34	175	0.30	
libras	91	180	0.51	m
dermatology	35	183	0.64	m
horse-colic	26	184	0.64	t
congressional-voting	17	217	0.59	
arrhythmia	263	226	0.67	m
musk-1	167	238	0.58	
cylinder-bands	36	256	0.38	
low-res-spect	101	265	0.25	m
monks-3	7	277	0.48	t
monks-1	7	278	0.51	t
breast-cancer-wisc-diag	31	284	0.63	
ilpd-indian-liver	10	291	0.72	
monks-2	7	300	0.64	t
synthetic-control	61	300	0.51	m
balance-scale	5	312	0.53	m
soybean	36	341	0.42	mt
credit-approval	16	345	0.43	
statlog-australian-credit	15	345	0.32	
breast-cancer-wisc	10	349	0.66	
blood	5	374	0.76	
energy-y1	9	384	0.80	m
energy-y2	9	384	0.75	m
pima	9	384	0.66	
statlog-vehicle	19	423	0.52	m
annealing	32	449	0.81	mt
oocytes_trisopterus_nucleus_2f	26	456	0.41	
oocytes_trisopterus_states_5b	33	456	0.98	m
tic-tac-toe	10	479	0.34	
mammographic	6	480	0.56	
conn-bench-vowel-deterding	12	495	0.53	mt
led-display	8	500	0.51	m
statlog-german-credit	25	500	0.72	
oocytes_merluccius_nucleus_4d	42	511	0.31	
oocytes_merluccius_states_2f	26	511	0.93	m
hill-valley	101	606	0.49	t
contrac	10	736	0.79	m

continued in the next page

Table 7: The data sets (continued)

data set	nfeatures	ndata	prop	note
yeast	9	742	0.55	m
semeion	257	796	0.50	m
plant-texture	65	799	0.50	m
wine-quality-red	12	799	0.56	m
plant-margin	65	800	0.52	m
plant-shape	65	800	0.52	m
car	7	864	0.28	m
steel-plates	28	970	0.66	m
cardiotocography-10clases	22	1063	0.39	m
cardiotocography-3clases	22	1063	0.86	m
titanic	4	1100	0.66	
image-segmentation	19	1155	0.58	mt
statlog-image	19	1155	0.55	m
ozone	73	1268	0.97	
molec-biol-splice	61	1595	0.75	m
chess-krvkp	37	1598	0.48	
abalone	9	2088	0.68	m
bank	17	2260	0.88	
spambase	58	2300	0.61	
wine-quality-white	12	2449	0.48	m
waveform-noise	41	2500	0.66	m
waveform	22	2500	0.68	m
wall-following	25	2728	0.78	m
page-blocks	11	2736	0.92	m
optical	63	2810	0.51	mt
statlog-landsat	37	3217	0.56	t
musk-2	167	3299	0.85	m
thyroid	22	3600	0.94	mt
ringnorm	21	3700	0.49	
twonorm	21	3700	0.49	
mushroom	22	4062	0.51	
pendigits	17	5496	0.51	mlt
nursery	9	6480	0.68	ml
magic	11	9510	0.65	l
letter	17	10000	0.50	ml
chess-krvk	7	14028	0.53	ml
adult	15	24421	0.76	lt
statlog-shuttle	10	29000	0.84	mlt
connect-4	43	33778	0.75	l
miniboone	51	65032	0.28	c

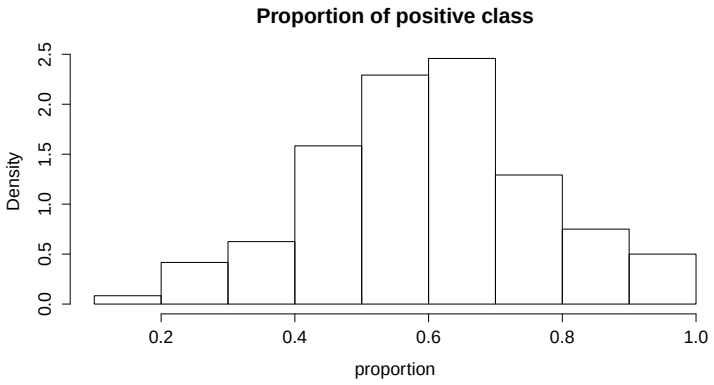

Figure 6: Histogram of the proportion of the positive class for the 121 data sets analysed.

Appendix C. Derivation of the Bayes factor formula

The derivation below is taken from chapter 12 of Barber (2012) by fixing the data for each classification (e_i) to be a pair number-of-correct-predictions and number-of-incorrect-predictions, and by computing the BF as $\frac{p(H_{\text{same}}|e_a,e_b)}{p(H_{\text{indep}}|e_a,e_b)}$ instead of $\frac{p(h_{\text{indep}}|e_a,e_b)}{p(h_{\text{same}}|e_a,e_b)}$. Let us assume that classifier a when running on the test set results in $e_a = \langle c_a, i_a \rangle$ where c_a is the number of correct predictions, and i_a the number of incorrect predictions, and similarly for classifier b. The standard Bayesian model selection method is to calculate:

$$\frac{P(H_1 \mid e_a, e_b)}{P(H_2 \mid e_a, e_b)} = \frac{P(e_a, e_b \mid H_1)}{P(e_a, e_b \mid H_2)} \frac{P(H_1)}{P(H_2)}.$$

In this case, H_1 is the hypothesis that the two data e_a and e_b come from independent multinomial distributions (H_{indep}) and H_2 that they come from the same distribution (H_{same}). Also let us assume that there is no prior preference for choosing any of these two hypothesis, and thus $P(H_{\text{indep}}) = P(H_{\text{same}})$.

For both hypothesis we will assume that the two values c_i and i_i are sampled from a multinomial distribution, $P(\langle c_a, i_a \rangle = \langle a_1, a_2 \rangle \mid \alpha =\rangle \alpha_1, \alpha_2 \rangle) = \alpha_1^{a_1} \alpha_2^{a_2}$, with an unknown α. α is sampled from a Dirichlet distribution:

$$P(\alpha) = \frac{1}{Z(u)} \alpha_1^{u_1 - 1} \alpha_2^{u_2 - 1},$$

where $Z(u)$ is the normalizing constant

$$Z(u) = \frac{\Gamma(u_1)\Gamma(u_2)}{\Gamma(u_1 + u_2)},$$

$\Gamma(.)$ is the gamma function and $u = \langle u_1, u_2 \rangle$ is the parameters of the Dirichelet distribution. $u = \langle 1, 1 \rangle$ results in a uniform distribution over the possible values of α_1 and α_2. The same is assumed for c_b and i_b.

For the independent hypothesis, we assume that the two data e_a and e_b comes from independent distributions, with independent α and β, that is

$$\begin{aligned} P(e_a, e_b \mid H_{\text{indep}}) &= \int P(e_a, e_b \mid \alpha, \beta, H_{\text{indep}}) P(\alpha, \beta \mid H_{\text{indep}}) d\alpha d\beta, \\ &= \int P(e_a \mid \alpha, H_{\text{indep}}) P(\alpha \mid H_{\text{indep}}) d\alpha \int P(e_b \mid \beta, H_{\text{indep}}) P(\beta \mid H_{\text{indep}}) d\beta, \\ &= \frac{Z(u + e_a)}{Z(u)} \frac{Z(u + e_b)}{Z(u)}. \end{aligned}$$

For the same hypothesis, we assume that both e_a and e_b were sampled from the same (unknown) multinomial distribution:

$$\begin{aligned} P(e_a, e_b \mid H_{\text{same}}) &= \int P(e_a, e_b \mid \alpha, H_{\text{same}}) P(\alpha, \mid H_{\text{same}}) d\alpha, \\ &= \int P(e_a \mid \alpha, H_{\text{same}}) P(e_b \mid \alpha, H_{\text{same}}) P(\alpha \mid H_{\text{same}}) d\alpha, \\ &= \frac{Z(u + e_a + e_b)}{Z(u)}. \end{aligned}$$

Thus, the Bayes factor is:

$$BF = \frac{p(H_{\text{same}} \mid e_a, e_b)}{p(H_{\text{indep}} \mid e_a, e_b)} = \frac{Z(u) Z(u + e_a + e_b)}{Z(u + e_a) Z(u + e_b)}. \tag{6}$$

Appendix D. Variance of the excess loss

Figure 7 is a scatter plot of all excess losses for all procedures as a function of the kf5 error rate. The figure seems to indicate that the variance ~~is~~ does not increase as the error rate increases, as discussed in the text. Figure 8 computes the variance of the excess losses for all procedures for different kf5 error rates. The kf5 error rate was divided into 20 equal ranges, and the variance of the excess loss was computed for each range. The figure places the variance at the mean kf5 error rate for each range. Not all ranges are represented in the figure because 3 of them fall between the last two kf5 error rates.

Appendix E. The outer loop

The outer loop in the experiments is a 2-fold, that is we run the experiments in half of the data set and measure the error rate in the second half, then we change the train and test halves, and average the two measures. But kf2 is a high variance estimation. To evaluate the consequence of this choice, we ran an experiment where the outer loop is a 5xkf2, a 5-times repeated kf2, for all data sets with less than 400 points. The results are reported in Figure 9. The figure plots the mean and confidence intervals for the original experiment with a kf2 outer loop and the 5xkf2 experiment, for 42 data sets with less than 400 data. The results show that there was a reduction in the confidence interval for the large- and mid-training procedures, but no systematic difference in the mean excess loss. For the small training procedures there seems to be a small decrease in the mean excess loss, and usually

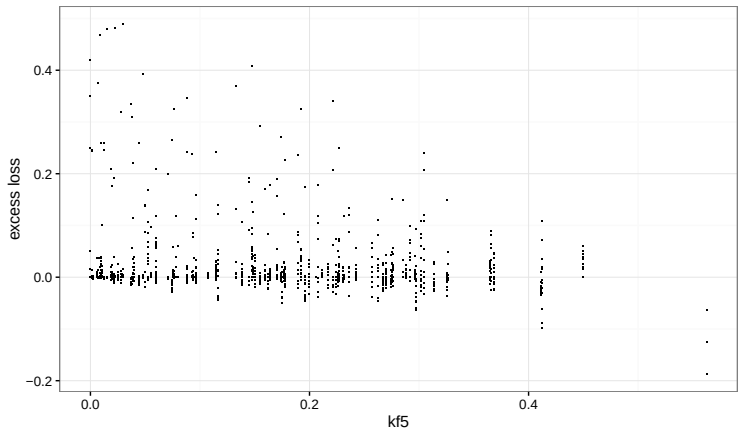

Figure 7: Scatter plot of all excess losses (for all procedures) as a function of the kf5 error rate.

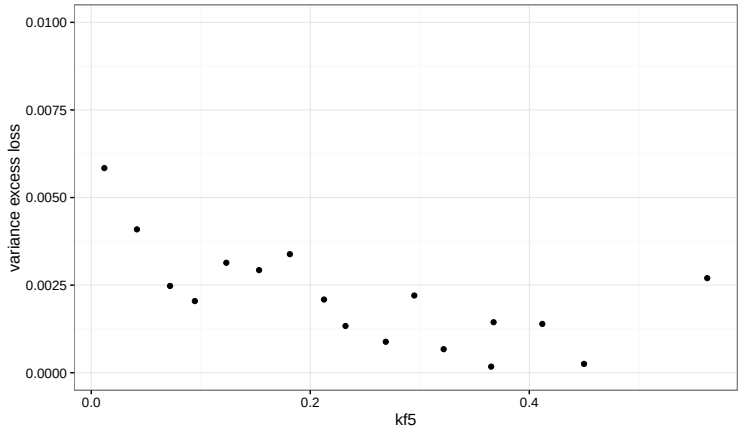

Figure 8: Variance of all excess losses for different kf5 error rates.

a small decrease in the confidence intervals, but not for all procedures. These differences should diminish as the data set increases in size. Thus had we used a repeated kf2 in the outer loop, the results would likely be similar, with maybe a small reduction on the sizes of the confidence intervals.

Appendix F. The full significance test comparison of all procedures

	kf5	kf5bis	kf2	kf3	2xkf5	kf10	5xboot	10xboot	20xboot	80/20	resub	invkf5	20/80	20/20	5x20/20	20/10	10/10
kf5bis	1.00																
kf2	1.00	1.00															
kf3	1.00	1.00	1.00														
2xkf5	1.00	1.00	1.00	1.00													
kf10	1.00	1.00	1.00	1.00	1.00												
5xboot	1.00	1.00	1.00	1.00	1.00	1.00											
10xboot	1.00	1.00	1.00	1.00	1.00	1.00	0.99										
20xboot	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00									
80/20	0.45	0.12	0.04	0.04	0.06	0.02	0.58	0.01	0.01								
resub	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00							
invkf5	1.00	1.00	0.96	0.96	0.98	0.88	1.00	0.82	0.78	0.94	0.00						
20/80	0.81	0.38	0.17	0.17	0.24	0.09	0.90	0.06	0.05	1.00	0.00	1.00					
20/20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.91	0.00	0.02	0.61				
5x20/20	0.94	0.61	0.34	0.34	0.44	0.20	0.98	0.15	0.12	1.00	0.00	1.00	1.00	0.37			
20/10	0.01	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	1.00	0.00	0.14	0.95	1.00	0.82		
10/10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.55	0.00	0.00	0.21	1.00	0.10	1.00	
5x10/10	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.99	0.00	0.09	0.89	1.00	0.72	1.00	1.00

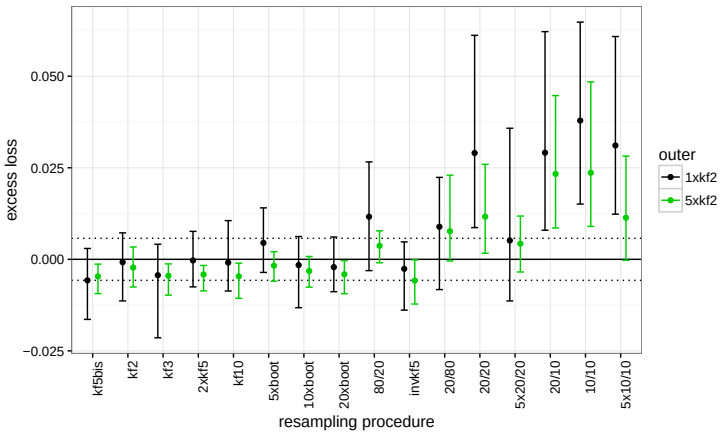

Figure 9: Comparison of using kf2 and 5xkf2 for the outer loop, for the 42 data sets with less than 400 data.

References

Davide Anguita, Andrea Boni, Sandro Ridella, Fabio Rivieccio, and Dario Sterpi. Theoretical and practical model selection methods for support vector classifiers. In *Support vector machines: theory and applications*, pages 159–179. Springer, 2005.

Davide Anguita, Alessandro Ghio, Luca Oneto, and Sandro Ridella. In-sample and out-of-sample model selection and error estimation for support vector machines. *IEEE Transactions on Neural Networks and Learning Systems*, 23(9):1390–1406, 2012. doi: 10.1109/TNNLS.2012.2202401.

David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(1):281–305, 2012.

Bernhard Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

Léon Bottou and Chih-Jen Lin. Support vector machine solvers. In *Large Scale Kernel Machines*, pages 301–320, 2007.

Ulisses Braga-Neto and Edward Dougherty. Bolstered error estimation. *Pattern Recognition*, 37(6):1267–1281, 2004.

Gavin C. Cawley and Nicola L.C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11:2079–2107, 2010.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, 2011.

Olivier Chapelle, Vladmir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1–3):131–159, January 2002.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995.

Mark A. Davenport, Richard G. Baraniuk, and Clayton D. Scott. Tuning support vector machines for minimax and Neyman-Pearson classification. *IEEE Transactions on Pattern Analysis and Machine Learning*, 32(10):1888–1898, 2010.

Bruno F. de Souza, André C.P.L.F. de Carvalho, Rodrigo Calvo, and Renato P. Ishii. Multiclass SVM model selection using particle swarm optimization. In *International Conference on Hybrid Intelligent Systems*, page 31, 2006.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

Kaibo Duan, Sathiya Keerthi, and Aun Neow Poo. Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, 51:41–59, 2003.

Bradley Efron. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78(382), 1983.

Bradley Efron. Better bootstrap confidence intervals. *Journal of the American Statistical Association*, 82(397):171–185, 1987.

Bradley Efron and Robert Tibshirani. Improvements on cross-validation: The 632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997.

Kai-Tai Fang, Dennis K.J. Lin, Peter Winker, and Yong Zhang. Uniform design: theory and application. *Technometrics*, 42(3):237–248, 2000.

Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15:3133–3181, 2014.

Frauke Friedrichs and Christian Igel. Evolutionary tuning of multiple SVM parameters. *Neurocomputing*, 64:107–117, 2005.

Terrence S. Furey, Nello Cristianini, Nigel Duffy, David W. Bednarski, Michl Schummer, and David Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

Martin J. Gardner and Douglas G. Altman. Confidence intervals rather than p values: estimation rather than hypothesis testing. *BMJ*, 292(6522):746–750, 1986.

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification, 2010. `https://www.cs.sfu.ca/people/Faculty/teaching/726/spring11/svmguide.pdf` accessed 1/2015.

Chien-Ming Huang, Yuh-Jye Lee, Dennis K.J. Lin, and Su-Yun Huang. Model selection for support vector machines via uniform design. *Computational Statistics & Data Analysis*, 52(1):335–346, 2007.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Machine Learning: ECML-98*, volume 1398 of *Lecture Notes in Computer Science*, pages 137–142. Springer, 1998.

Thorsten Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

Thorsten Joachims. *The Maximum-Margin Approach to Learning Text Classifiers: Methods, Theory, and Algorithms*. PhD thesis, Department of Computer Science, University of Dortmund, 2000.

Robert E. Kass and Adrian E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

Sathiya Keerthi. Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. *IEEE Transactions on Neural Networks*, 13(5):1225–1229, 2002.

Sathiya Keerthi and Chih-Jen Lin. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003.

Sathiya Keerthi, Vikas Sindhwani, and Olivier Chapelle. An efficient method for gradient-based adaptation of hyperparameters in SVM models. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems*, pages 673 – 680. MIT Press, 2007.

Roger E. Kirk. Practical significance: A concept whose time has come. *Educational and psychological measurement*, 56(5):746–759, 1996.

Tammo Krueger, Danny Panknin, and Mikio Braun. Fast cross-validation via sequential testing. *Journal of Machine Learning Research*, 16:1103–1155, 2015.

Max Kuhn. Futility analysis in the cross-validation of machine learning models. *arXiv preprint arXiv:1405.6974*, 2014.

Max Kuhn et al. *Package caret: Classification and Regression Training*, version 6.0-37 edition, 2014. `http://cran.r-project.org/web/packages/caret/caret.pdf`.

Shutao Li and Mingkui Tan. Tuning SVM parameters by using a hybrid CLPSO-BFGS algorithm. *Neurocomputing*, 73(10-12):2089–2096, 2010.

Moshe Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Annette M. Molinaro, Richard Simon, and Ruth M. Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–7, 2005.

John A. Nelder and Roger Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

Bruce Thompson. What future quantitative social science research could look like: Confidence intervals for effect sizes. *Educational Researcher*, 31(3):25–32, 2002.

Vladimir Vapnik. *Statistical learning theory*. Adaptive and learning systems for signal processing, communications and control series. Wiley, 1998.

Vladimir Vapnik and Olivier Chapelle. Bounds on error expectation for support vector machines. *Neural computation*, 12(9):2013–2036, 2000.

Grace Wahba. Advances in kernel methods. chapter Support Vector Machines, Reproducing Kernel Hilbert Spaces, and Randomized GACV, pages 69–88. MIT Press, Cambridge, MA, USA, 1999.

Chris Woolston. Psychology journal bans p values. *Nature*, 519(7541), 2015.