frontiers in
ROBOTICS AND AI

PERSPECTIVE ARTICLE
published: 25 September 2014
doi: 10.3389/frobt.2014.00005

Two challenges of correct validation in pattern recognition

Thomas Nowotny *

Centre for Computational Neuroscience and Robotics, School of Engineering and Informatics, University of Sussex, Brighton, UK

Edited by:
Fabrizio Riguzzi, Università di Ferrara, Italy

Reviewed by:
Dalal Alrajeh, Imperial College London, UK
Annalisa Appice, University of Bari Aldo Moro, Italy

****Correspondence:***
Thomas Nowotny, Centre for Computational Neuroscience and Robotics, School of Engineering and Informatics, University of Sussex, Falmer, Brighton BN1 9QJ, UK
e-mail: t.nowotny@sussex.ac.uk

Supervised pattern recognition is the process of mapping patterns to class labels that define their meaning. The core methods for pattern recognition have been developed by machine learning experts but due to their broad success, an increasing number of non-experts are now employing and refining them. In this perspective, I will discuss the challenge of correct validation of supervised pattern recognition systems, in particular when employed by non-experts. To illustrate the problem, I will give three examples of common errors that I have encountered in the last year. Much of this challenge can be addressed by strict procedure in validation but there are remaining problems of correctly interpreting comparative work on exemplary data sets, which I will elucidate on the example of the well-used MNIST data set of handwritten digits.

Keywords: pattern recognition, validation, crossvalidation, overfitting, meta-learning

1. INTRODUCTION

Pattern recognition is the process of mapping input data, a pattern, to a label, the "class" to which the input pattern belongs. Among the common approaches to pattern recognition, supervised machine learning approaches have gained a lot of momentum with headline successes, e.g., on the MNIST data set of handwritten digits (LeCun and Cortes, 1998), which I will use for illustration later. In this paradigm, researchers use a labeled set of examples, the training set, to "teach" algorithms to predict the correct label for each input pattern. The success of learning is then tested on a separate set of labeled examples, the test set. If no such separate test set exists, crossvalidation or bootstrapping methods are employed to assess the success of a developed method.

Popular examples of supervised learning methods for pattern recognition include k-nearest-neighbor classification (kNN) (Cover and Hart, 1967), support vector machines (Boser et al., 1992; Cortes and Vapnik, 1995), and artificial neural networks (McCulloch and Pitts, 1943; Farley and Clark, 1954; Werbos, 1974), e.g., multilayer perceptrons (Rosenblatt, 1958), and their recent reincarnation in the form of deep learning networks (Fukushima, 1980; Schmidhuber, 1992; Hinton, 2007) and convolutional neural nets (LeCun et al., 1998). These are complemented by a large variety of more exotic methods and ensemble learning methods such as boosting (Schapire, 1990; Freund and Schapire, 1997) and random forests (Breiman, 2001). While there are still developments, e.g., in SVM technology (Huerta et al., 2012) and deep learning architectures, there is an increasing number of studies focusing on preprocessing of data, refining meta-parameters, and applying the established methods to novel real world applications. This trend is driven by scientists who are not necessarily experts in machine learning but want to apply machine learning methods in their own application domain.

It is on this background that I would like to highlight two aspects of validation and crossvalidation that do not seem to be fully appreciated in the larger community of applied machine learning practitioners: (i) the need for correct and strict procedure in validation or crossvalidation and (ii) the need for careful interpretation of validation results if multiple studies use the same reference data set. Both issues are different aspects of the same problem of overfitting and hence closely related, but while the former has known solutions that applied researchers can be informed about the latter is more involved and invites further research by machine learning experts.

2. STRICT PROCEDURE OF VALIDATION IN INDIVIDUAL STUDIES

The idea of validation in pattern recognition rests on the principle that separate data are used to develop a method (the training set) and to subsequently test its performance (the test set). This principle has been established to avoid overfitting, i.e., the situation in which a method is so specific to particular training data that it does not generalize to unseen new examples.

To avoid overfitting and give a reasonable prediction of the performance of a method on new unseen data, the correct procedure stipulates that the test data *shall not be used in any way for training the classifier or developing the classification method.* Notably, this includes that one shall never perform "intermediate tests" on the test data to check different versions of the method. The same principle applies to crossvalidation in which training and test sets are formed repeatedly by splitting a single available data set (see **Figure 1** for an introduction). This principle of strict validation is well known; yet, in the last year alone, I have encountered three violations of the principle in published and unpublished work.

2.1. EXAMPLE 1: ADJUSTING SVM META-PARAMETERS

In this example, researchers investigated a new method for recognizing odors with an electronic nose, eventually employing a support vector machine (SVM) algorithm with radial basis function (RBF) kernel on the data collected with their novel method. To adjust the meta-parameters of the SVM, they ran a grid search of parameters using crossvalidation with their entire data set to

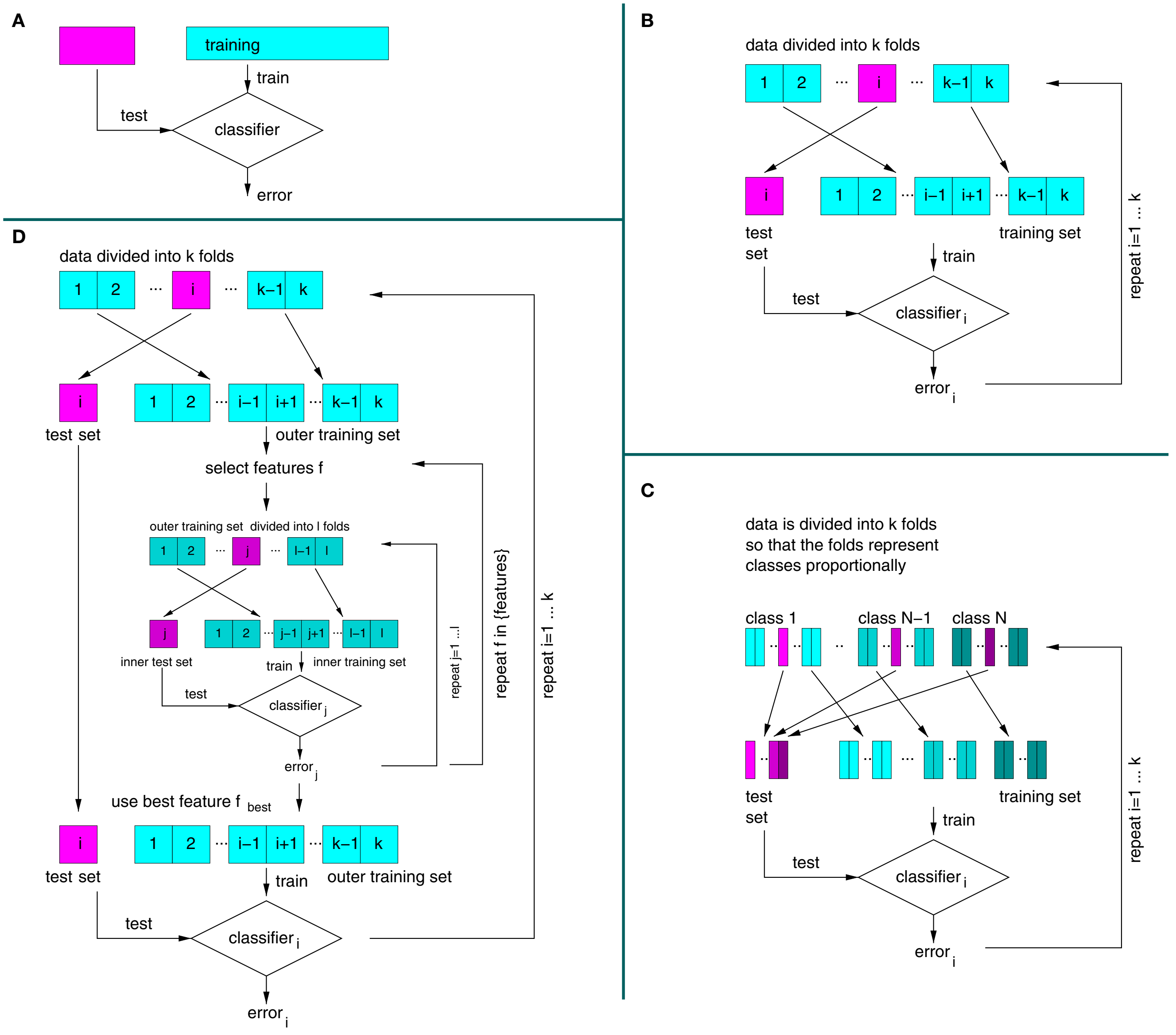

FIGURE 1 | Diagrammatic illustration of validation (A) and crossvalidation (B–D) methods. **(A)** Normal validation with a true test set (holdout set). A part of the data (purple) is permanently withheld from training and used for testing after training has been completed. **(B)** *k*-fold crossvalidation. The data are divided into *k* parts (folds) and one of the folds is withheld from training and used for testing (purple). The procedure is repeated until all folds have been withheld once and the average error is reported. **(C)** Stratified crossvalidation. In stratified crossvalidation, the folds are formed so that they contain the same proportion of members of each class (as much as this is possible), indicated by the purple slices taken from each class. Crossvalidation then proceeds as described in **(B)**. The variability of error estimates that arises from the choice of folds is reduced in stratified crossvalidation. **(D)** Inner and outer crossvalidation for feature selection. If feature selection is itself based on crossvalidation, e.g., in wrapper methods, then an inner and outer crossvalidation procedure must be applied. First, the "outer test set" (light purple) is withheld, then the remaining "outer training set" (light cyan) is again split into an "inner holdout set" (dark purple) and "inner training set" (dark cyan), which are used to select features with standard crossvalidaton, i.e., the features that give the best error estimate in the "j loop" will be chosen. Once feature selection is complete, the "outer training set" is used for training, and only in the very end, the "outer test set" is used for testing.

judge parameter suitability. The study then proceeded to use the identified meta-parameters to report crossvalidation results for classification performance. This is in violation of the principle of strict separation of training and testing (holdout) data but when asked, the authors replied that they thought "it was common practice."

2.2. EXAMPLE 2: CHOOSING AN OBSERVATION WINDOW

In this example, the research was focused on characterizing how well the spiking activity of an identified neuron in the brain of crickets would represent the quality of a stimulus. The researchers used activity from within a time window and crossvalidation of a naive Bayes classifier to determine how well the neuronal activity could predict whether a stimulus was attractive or not. This was repeated for different time window sizes and the performance of the classifier for the best time window was reported. Doing so inadvertently created the impression that the animal may be able to perform at the reported best performance level when utilizing the activity of the neuron in question. However, such a conclusion is not supported by the study because the same data were used for model selection (selecting the observation time window) and final crossvalidation.

2.3. EXAMPLE 3: FEATURE SELECTION FOR HIGH-DIMENSIONAL DATA

In this example, published work on pattern recognition of high-dimensional data from gas chromatography and mass spectrometry for the analysis of human breath involved a preprocessing pipeline including a statistical test for selecting significant features, principal component analysis (PCA), and canonical analysis (CA). Data preprocessing was performed on the entire data set and subsequently the performance of a kNN classifier was evaluated in crossvalidation. This is a clear violation of the principle of strict separation of training and test sets.

In this example, we were able to get an estimate for the possible impact of overfitting by using our own data of similar nature (Berna et al., unpublished). We collected the breath of 10 healthy adults on 4 days each, with 3 repetitions each day, and analyzed it with a commercial enose. We found that when applying the method of feature selection described above on the entire data set and leave-one-out (LOO) crossvalidation only for the subsequent kNN classification, the 10 classes (10 different healthy subjects) could be identified with 0% error. If we used LOO crossvalidation more strictly for the entire procedure of feature selection and classification, the error rate increased to 29.2%. Much worse, when we performed crossvalidation so that strictly all 3 repetitions from a day were removed together (stratified 40-fold crossvalidation), the error rate was worse than chance levels (92.5%). These numbers illustrate that the correct procedure for crossvalidation in the context of high-dimensional data, few samples, and long processing pipelines is not just a detail but can determine success or failure of a method.

It is essential that this knowledge is passed on to applied researchers with the increasing popularity of machine learning methods in applications [see also Ransohoff (2004), Broadhurst and Kell (2006), and Marco (2014)]. Furthermore, it is important to be clear that *any* use of the test or holdout data introduces serious risks of overfitting. This includes the following examples that at times seem to be tolerated as "common practice":

1. Adjusting meta-parameters. If crossvalidation is used for adjusting meta-parameters, an inner and outer crossvalidation procedure must be performed (see **Figure 1D**).
2. Model selection. Testing different methods with crossvalidation and reporting the best one constitutes overfitting because the holdout sets are used in choosing the method.
3. Excluding outliers. If the identification of outliers depends on their relationship to other inputs, test data should not be included in the decision process.
4. Clustering or dimensionality reduction. These preprocessing methods should only have access to training data.
5. Statistical tests. If using statistical tests for feature selection, test data cannot be included.

3. INTERPRETATION OF MULTIPLE STUDIES ON A COMMON DATA SET

We have seen above that problems with the correct use of validation methods occur, in particular in applied work, which can be addressed by adhering to established correct procedures. However, I would like to argue that beyond the established best practice, there is also room for further research on the risks of overfitting in the area of collaborative work on representative data sets. To give a concrete example, researchers use tables like the table of classifier performance for the MNIST data set on LeCun's website (LeCun and Cortes, 1998) to choose the best method for a given application. When doing so it would be natural to expect a performance close to the reported validation accuracy from the table. However, this expectation is not fully justified because the MNIST test set was used multiple times: every study reported on the MNIST website (LeCun and Cortes, 1998) used the same MNIST test set to indicate predicted performance. Therefore, if we use the reported studies for model selection we inadvertently introduce a risk of overfitting. The practical implication is that we cannot be sure to have truly selected the best method and we do not know whether and how much the reported accuracy estimate may be inflated by overfitting. When Fung et al. (2008) investigated the possible scope of this problem in the context of model selection using crossvalidation, they tested an increasing number of different classifiers using leave-one-out (LOO) crossvalidation on synthetic data of 100 samples and 16 dimensions that had a true prediction accuracy of 0.5 (pure chance). They found that the best predicted accuracy for the repeated crossvalidation with different algorithms varied from 0.619 when selecting from 10 algorithms to 0.856 for selecting from 10^6 algorithms. The crossvalidation estimates of accuracy in this example are hence largely over-optimistic, and this is already the case for only 10 tested algorithms (the MNIST table reports 69 different algorithms).

Related work by Isaksson et al. (2008) suggests Bayesian confidence intervals (Jaynes, 1976, 2003; Webb, 2002) as a promising method for identifying the level of expected variability of validation results. Assuming no prior knowledge, the posterior distribution underlying Bayesian confidence intervals only depends on the number of correct predictions and the total number of test samples, so that we can calculate confidence intervals for the MNIST results directly from the published table (LeCun and Cortes, 1998). **Figure 2A** shows the results. As indicated by the gray bar, all methods to the right of the vertical dashed line have confidence intervals that overlap with the confidence interval of the best method. We can interpret this as an indication that they should probably be treated as equivalent.

Attaching confidence intervals to the predicted accuracies is an important step forward both for underpinning the selection process and the judgment on the expected accuracy of the selected method. However, it is important to be aware that the proposed Bayesian confidence intervals rest on the assumption that the test samples are all statistically independent. In a real world data set like the MNIST handwritten digits, this assumption may only hold partially. The MNIST digits were written by separate writers who

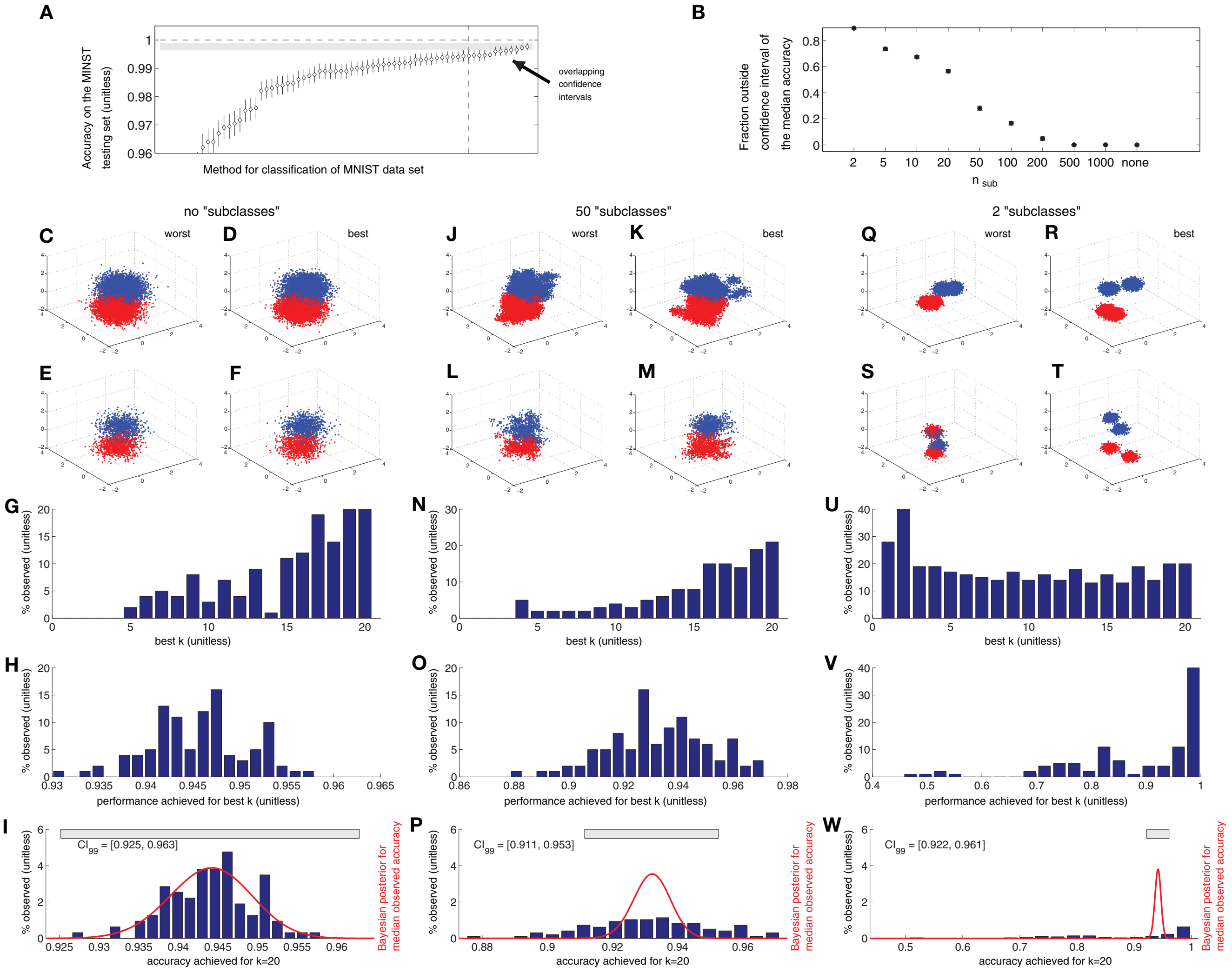

FIGURE 2 | (A) Reported accuracy of classification on the MNIST test set and Bayesian confidence intervals. The confidence intervals were calculated assuming no prior knowledge on the classification accuracy, and for confidence level 0.99. The gray bar indicates the confidence interval for the best method and the dashed line separates the methods whose confidence intervals intersect with the bar from the rest. **(B)** Fraction of observed accuracy values on the test set that lie outside the Bayesian confidence interval of the median observed accuracy [99% confidence level, see **(I)**, **(P)**, and **(W)** for illustration]. Bullets are the mean observed fraction and errorbars the standard deviation across all values of *k* (most errorbars are too small to be visible). For values of n_{sub} of 500 and above, this fraction is 0 for all *k*, i.e., the confidence interval contains all observed values of the accuracy. **(C–W)** Illustration of the synthetic data experiment with random vectors underlying **(B)**. **(C–F)** example training **(C,D)** and test **(E,F)** sets for random vectors of class 0 (red) and 1 (blue) for unstructured training and test sets. **(C,E)** are the worst performing example out of 50 repetitions and **(D,F)** the example where the classifiers perform best. **(G)** Histogram of the observed distribution of occurrence of the optimal *k* value in kNN classification of the test set. **(H)** Histogram of the distribution of observed classification accuracies for the test set, pooled for all *k*. **(I)** Histogram for the distribution of observed classification accuracies for $k = 20$ (bars) and posterior Bayesian distribution for the probability of observed accuracy, given the median observed classification accuracy. The gray bar demarcates the Bayesian confidence interval for the median observed accuracy at 99% confidence level. **(J–P)** Same plots as **(C–I)** for training and test sets that consist of 50 sub-classes. **(Q–W)** Same plots for training and test sets that consist of 2 sub-classes.

will have different ways of writing each of the digits and variations between writers would be larger than variations between repeated symbols from the same writer.

To investigate whether such correlations could have an effect on the validity of confidence intervals, I have created the following synthetic problem. I consider a classification problem that has two classes $\{0, 1\}$ and the input patterns of the two classes are three-dimensional random vectors, with entries that are taken from a Gaussian distribution with mean $\mu = 0$ (class 0) and $\mu = 1$ (class 1). Both have entries with standard deviation $\sigma = \sqrt{0.5^2 + 0.2^2} \approx 0.539$ (the reason for this choice will become apparent in a moment). I have generated 6000 examples for each class for the training set ($n_{\mathrm{train}} = 12000$) and 1000 independent inputs per class in a test set ($n_{\mathrm{test}} = 2000$), mimicking the size of the classes in the MNIST set. **Figures 2C,D** show two examples of training sets and 2E,F of test sets. Then, I used kNN classifiers to classify samples, where k varied, $k \in \{1, \ldots, 20\}$. The analogy to the MNIST example is that each k value would correspond to one of the different methods used on the MNIST data set. I test each classifier on the independent test set, from which we can determine the expected performance as it would be reported in the MNIST table.

Because the data are purely synthetic, I can generate as many of these experiments as I would like, which would correspond to an arbitrary number of MNIST sets in the analogy. When I repeat the described experiment with 100 independently generated training and test sets, I observe the effects illustrated in **Figures 2G–I**. The k value leading to the best performance varies between instances (**Figure 2G**), and so does the distribution of achieved maximal performance (**Figure 2H**). Furthermore, the posterior distribution underlying the Bayesian confidence interval corresponds well with the observed distribution of performances (**Figure 2I**). In summary, for the synthetic data with fully independent samples, the Bayesian confidence intervals with naive prior work very well, as expected.

I then introduced a sub-structure into the data by choosing n_{sub} 3-dimensional random vectors with components drawn from a Gaussian distribution of mean $\mu = 1$ and standard deviation $\sigma = 0.5$ and adding $n_{\text{train}}/n_{\text{sub}}/2$ independent 3-dimensional random vectors with Gaussian entries of $\mu = 0$ and $\sigma = 0.2$ to each of them to create the $n_{\text{train}}/2$ training vectors of class 1. Class 0 was generated similarly but with $\mu = 0$. As a result of this, the training and test sets now consist of n_{sub} "sub-clouds" of points as can be seen in the example plots in **Figures 2J–M,Q–T**. Each such sub-cloud would correspond to the digits written by a different person in the MNIST analogy. The standard deviation in the unstructured example above was chosen to match the overall standard deviation of the structured samples here. I generated the structured data for $n_{\text{sub}} \in \{2, 5, 10, 20, 50, 100, 200, 500, 1000\}$. The differences to the unstructured case are quite drastic (**Figures 2J–W**). The preference for large k in successful classifiers diminishes until there is no preferred value for 2 sub-classes. Similarly, the best achieved performance becomes more volatile across instances and the Gaussian posterior distribution for the probability of the observed accuracy is no longer a good description of the distribution of observed accuracies for $k = 20$ (**Figures 2P,W**) and similarly for other k values (data not shown).

Figure 2B shows the percentage of observed accuracy values in the 100 instances that fall outside the Bayesian confidence interval of the median accuracy (99% confidence level). It rises from zero for the unstructured case to its maximal value of 82.8% for $n_{\text{sub}} = 2$. This indicates that Gaussian confidence intervals of this kind are surprisingly vulnerable to correlations in the data.

I should repeat at this point that this discussion is not about criticizing the work with the MNIST data set, which has been very valuable to the field, and I have used MNIST myself in the context of bio-mimetic classification (Huerta and Nowotny, 2009; Nowotny et al., 2011). The problem that I am trying to expose is that while some headway has been made with investigating the possible bias in model selection (Fung et al., 2008) and quantifying uncertainty about accuracy estimates with confidence intervals (Isaksson et al., 2008), we as of yet have no definite answer how to assess the exact risk of overfitting when selecting models based on a comparison on the same test set. One radical solution to the problem would be to ban testing on the same test set altogether. However, this leads to a difficult contradiction: to make fair comparisons, we need to compare algorithms on equal terms (the same training and test set) but due to the discussed unknown biases, we would rather like to avoid using the same training and test set multiple times.

Another simple but potentially costly improvement would be to completely halt work on a data set, acquire a (suitably defined) equivalent new data set (new test set), select the method that appears best in spite of the possible bias and test this method with the new sets. The observed performance in this fully independent test could then at least be interpreted as an unbiased prediction of this method's performance, albeit still not endowed with a measure of uncertainty and with no guarantees that indeed the best method has been chosen.

Alternatively, we could employ outer and inner crossvalidation as suggested by Cawley and Talbot (2010) and illustrated in **Figure 1D**. For example, crossvalidation on the same folds of the MNIST training set could be used for model selection and the test set would only be used for the final prediction of accuracy. *Post hoc* analysis of this kind would be an interesting future direction for the most prominent "headline data sets."

4. DISCUSSION AND CONCLUSION

In this perspective, I have highlighted the problem of inadvertent overfitting in pattern recognition work. One element to avoid overfitting is strict procedure, possibly beyond what is common practice today. Going a step further, I have then attempted to illustrate that proper strict procedure will, however, not fully remove overfitting due to an inherent conflict between working comparatively on an example data set and avoiding meta-learning when using the results for model selection. Fung et al. (2008) refer to this problem as "overfitting in (cross)validation space" and Isaksson et al. (2008) have suggested Bayesian confidence intervals to get an estimate for potential biases. I illustrated the problem on synthetic data and found that Bayesian confidence intervals work well if the data contain fully independent samples but begin to underestimate potential biases if correlations between samples are introduced. It would be an interesting area for further research how confidence intervals could be formulated also in the case of dependent samples.

The discussed work on confidence intervals applies to situations with a true holdout set. Other related work has addressed the question which statistical tests are most appropriate to judge the accuracy of different pattern recognition methods when using crossvalidation. While initially t-tests and binomial tests were employed, more modern approaches use ANOVA and Wilcoxon signed ranks tests with *post hoc* multicomparison corrected Friedman tests (Demsar, 2006).

When using crossvalidation, it is also good to be aware of its vulnerability to small sample sizes and high dimensionality. It has been shown in the context of hard margin support vector machines that in the limit of infinite dimension and finite sample size, crossvalidation can fail systematically (Hall et al., 2005; Klement et al.,

2008) and that this effect already appears for moderately large dimensionality.

In conclusion, it is very important to remember that a strict procedure has to be followed to avoid inadvertent overfitting, in particular when a good amount of "data preprocessing" is involved. It may be timely for machine learning experts to reach out to the wider community of applied researchers and clearly communicate the appropriate use of machine learning methods and how to avoid common pitfalls. Furthermore, as we have seen above, very careful discipline is needed in comparative work on representative data sets. More work on quantifying the uncertainty of accuracy estimations in such situations would be very useful. For applications where the outcome may lead to critical decisions, e.g., in health- or high-risk technological applications, (cross)validation of any suggested final solution on a further, independent, unseen data set will be essential.

ACKNOWLEDGMENTS

This work was partially supported by The Engineering and Physical Sciences Research Council [grant number EP/J019690/1].

REFERENCES

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (New York, NY: ACM), 144–152. COLT '92.

Breiman, L. (2001). Random forests. *Mach. Learn.* 45, 5–32. doi:10.1023/A:1017934522171

Broadhurst, D., and Kell, D. (2006). Statistical strategies for avoiding false discoveries in metabolomics and related experiments. *Metabolomics* 2, 171–196. doi:10.1007/s11306-006-0037-z

Cawley, G. C., and Talbot, N. L. C. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.* 11, 2079–2107.

Cortes, C., and Vapnik, V. (1995). Support-vector networks. *Mach. Learn.* 20, 273–297. doi:10.1023/A:1022627411411

Cover, T., and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* 13, 21–27. doi:10.1109/TIT.1967.1053964

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* 7, 1–30.

Farley, B. G., and Clark, W. (1954). Simulation of self-organizing systems by digital computer. *Inf. Theory Trans. IRE Prof. Group* 4, 76–84. doi:10.1109/TIT.1954.1057468

Freund, Y., and Schapire, R. E. (1997). A decision-theoretic generalization of online learning and an application to boosting. *J. Comput. Syst. Sci.* 55, 119–139. doi:10.1006/jcss.1997.1504

Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* 36, 193202.

Fung, G., Rao, R. B., and Rosales, R. (2008). "On the dangers of cross-validation. an experimental evaluation (SIAM)," in *Proceedings of the 2008 SIAM International Conference on Data Mining*, eds C. Apte, H. Park, K. Wang, and M. J. Zaki. 588–596. doi:10.1137/1.9781611972788.54

Hall, P., Marron, J. S., and Neeman, A. (2005). Geometric representation of high dimension, low sample size data. *J. R. Stat. Soc.* 67, 427–444. doi:10.1111/j.1467-9868.2005.00510.x

Hinton, G. E. (2007). Learning multiple layers of representation. *Trends Cogn. Sci.* 11, 428–434. doi:10.1016/j.tics.2007.09.004

Huerta, R., and Nowotny, T. (2009). Fast and robust learning by reinforcement signals: explorations in the insect brain. *Neural Comput.* 21, 2123–2151. doi:10.1162/neco.2009.03-08-733

Huerta, R., Vembu, S., Amigo, J. M., Nowotny, T., and Elkan, C. (2012). Inhibition in multiclass classification. *Neural Comput.* 24, 2473–2507. doi:10.1162/NECO_a_00321

Isaksson, A., Wallman, M., Göransson, H., and Gustafsson, M. (2008). Cross-validation and bootstrapping are unreliable in small sample classification. *Pattern Recognit. Lett.* 29, 1960–1965. doi:10.1016/j.patrec.2008.06.018

Jaynes, E. (1976). "Confidence intervals vs Bayesian intervals," in *Foundations of Probability Theory, Statistical Inference, and Statistical Theories of Science*, Vol. II, eds W. Harper and C. Hooker (Dordrecht: D. Reidel), 175257.

Jaynes, E. (2003). *Probability Theory: The Logic of Science*. Cambridge: Cambridge University Press.

Klement, S., Madany Mamlouk, A., and Martinetz, T. (2008). "Reliability of cross-validation for SVMs in high-dimensional, low sample size scenarios," in *Artificial Neural Networks - ICANN 2008, Volume 5163 of Lecture Notes in Computer Science*, eds V. Kuková, R. Neruda, and J. Koutník (Berlin: Springer), 41–50.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc IEEE* 86, 2278–2324. doi:10.1109/5.726791

LeCun, Y., and Cortes, C. (1998). *The MNIST Database*. Available at: http://yann.lecun.com/exdb/mnist/, accessed 27/08/2014

Marco, S. (2014). The need for external validation in machine olfaction: emphasis on health-related applications. *Anal. Bioanal. Chem.* 406, 3941–3956. doi:10.1007/s00216-014-7807-7

McCulloch, W., and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* 5, 115–133. doi:10.1007/BF02478259

Nowotny, T., Muezzinoglu, M. K., and Huerta, R. (2011). Biomimetic classification on modern parallel hardware: realizations on NVidia® CUDA™ and OpenMP™. *Int. J. Innov. Comput.* 7, 3825–3838.

Ransohoff, D. F. (2004). Rules of evidence for cancer molecular-marker discovery and validation. *Nat. Rev. Cancer* 4, 309–314. doi:10.1038/nrc1322

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* 65, 386–408. doi:10.1037/h0042519

Schapire, R. (1990). The strength of weak learnability. *Mach. Learn.* 5, 197–227. doi:10.1023/A:1022648800760

Schmidhuber, J. (1992). Learning complex, extended sequences using the principle of history compression. *Neural Comput.* 4, 234–242. doi:10.1162/neco.1992.4.2.234

Webb, A. (2002). *Statistical Pattern Recognition*. Chichester: Wiley.

Werbos, P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Ph.D. Thesis, Harvard University, Cambridge, MA.

Conflict of Interest Statement: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 23 June 2014; accepted: 27 August 2014; published online: 25 September 2014.
Citation: Nowotny T (2014) Two challenges of correct validation in pattern recognition. Front. Robot. AI ***1****:5. doi: 10.3389/frobt.2014.00005*
This article was submitted to Computational Intelligence, a section of the journal Frontiers in Robotics and AI.