Multi-Model Comparison Using the Cross-Fitting Method

Holger Schultheis (schulth@informatik.uni-bremen.de)
Cognitive Systems, University of Bremen, Enrique-Schmidt-Str. 5, 28359 Bremen, Germany

Praneeth Naidu
Computer Science and Engineering, IIT Bombay, Mumbai 400076, India

Abstract

When comparing the ability of computational cognitive models to fit empirical data, the complexity of the compared models needs to be taken into account. A promising method for achieving this is the parametric bootstrap cross-fitting method (PBCM) proposed by Wagenmakers, Ratcliff, Gomez, and Iverson (2004). We contribute to a wider applicability of the PBCM in two ways: First, we compare the performance of the data-informed and the data-uninformed variant of the PBCM. Our simulations suggest that only the data-uninformed variant successfully controls for model complexity in model selection. Second, we propose an extension of the PBCM, called MMPBCM, that is applicable to, in principle, arbitrarily many competing models. We evaluate the MMPBCM by applying it to the comparison of several sets of competing models. The obtained results suggest that the MMPBCM constitutes a more powerful approach to model comparison than the PBCM.

Keywords: Model Evaluation, Multi-Model Comparison, Parametric Bootstrap Crossfitting Method.

Introduction

It is often considered an advantage of computational cognitive models that they allow generating data by simulation and this article concerns this type of data-generating models. One way to evaluate and compare such models is to generate data from them and to compare the model-generated data to empirical data pertinent to the phenomenon that is being modeled. The degree of correspondence between the model-generated and the empirical data is often called the *goodness of fit* (GOF) and it may be used to assess the quality of the competing models: The higher the GOF, the better the model.

However, such a naïve use of GOF measures for model comparison is problematic, because it neglects model complexity. Due to overfitting, more complex models may provide high GOF measures solely by virtue of their complexity. As a result, the naïve use of GOF measures may lead to the selection of a more complex model even if a less complex model actually provides a better approximation to the processes that underlie the phenomenon that is being investigated (Pitt & Myung, 2002).

To address this problem, a number of methods have been proposed that take model complexity into account when comparing how well models can account for empirical data (see Shiffrin, Lee, Kim, & Wagenmakers, 2008; Schultheis, Singhaniya, & Chaplot, 2013, for overviews). One of these methods is the *parametric bootstrap cross-fitting method* (PBCM) proposed by Wagenmakers et al. (2004). Two properties of the PBCM render it particularly appealing for model evaluation and selection: First, the PBCM is applicable to any type of model, since it imposes no constraints on the modeling paradigm or the models' structure. Second, if one of the compared models captures the actual processes that generated the to-be-fitted data, the PBCM has been considered to perform optimally in selecting this model (Shiffrin et al., 2008; Cohen, Sanborn, & Shiffrin, 2008).

Given these properties, employment of the PBCM instead of the naïve use of GOF measures seems highly desirable. At the same time, two aspects of the PBCM – as so far discussed in the literature – may hamper or even preclude use of the PBCM in certain modeling situations. For one, in the article introducing the PBCM, Wagenmakers et al. (2004) propose two different variants of the PBCM called the *data-informed* PBCM (DIPBCM) and the *data-uninformed* PBCM (DUPBCM). Since these two variants differ considerably in their computational complexity, it would be important to know to what extent their performance in model comparison differs. Initial analyses presented in Wagenmakers et al. (2004) suggest that the DIPBCM may generally perform worse than the DUPBCM, but information that allows more detailedly quantifying potential differences between the two variants is currently not available from the literature considering the PBCM (Wagenmakers et al., 2004; Cohen, Sanborn, & Shiffrin, 2008; Cohen, Rotello, & MacMillan, 2008; Jang, Wixted, & Huber, 2011; Perea, Gomez, & Fraga, 2010). Furthermore, both PBCM variants are currently restricted to the comparison of pairs of models. When more than 2 competing models need to be compared, this comparison must be broken down to multiple comparisons of model pairs or the PBCM cannot be applied at all.

In this article, we provide a first systematic quantitive comparison of the DIPBCM and the DUPBCM regarding their model selection performance. We also propose and evaluate an extension of the PBCM that allows comparing more than two competing models. Both contributions facilitate the use of the PBCM and, thus, more generally, are conducive to increasing the frequency with which more sophisticated comparison methods instead of the naïve approach will be employed for model evaluation and comparison.

The PBCM

Let A and B be two competing models and $\mathbf{x}$ a set of observed data (e.g., response times from different experimental conditions). Furthermore, let $\Delta gof^{\mathbf{x}}_{AB}$ be the GOF difference of the two models on the data set $\mathbf{x}$, that is, $\Delta gof^{\mathbf{x}}_{AB} = gof^{\mathbf{x}}_{A} - gof^{\mathbf{x}}_{B}$, where $gof^{\mathbf{x}}_{A}$ and $gof^{\mathbf{x}}_{B}$ are the goodness of fits the models A and B achieve on $\mathbf{x}$, respectively. A naïve approach to model comparison would select A if $\Delta gof^{\mathbf{x}}_{AB} \geq 0$ and B otherwise. The PBCM aims to improve on the naïve approach by tak-

ing into account how well the models are able to mimic each other, that is, the ability of each model to provide good fits to data generated by the other model.

To achieve this, the PBCM generally proceeds as follows:

1. generate a set of parameter values for all parameters of model A,
2. generate a data set $\mathbf{x}_A$ by running model A with the parameter values from the first step,
3. fit both models to $\mathbf{x}_A$ to obtain $\Delta gof_{AB}^{\mathbf{x}_A}$,
4. repeat the above three steps NBS number of times.

These steps will result in NBS many GOF differences for data that has been generated from model A. If the same four steps are repeated with model B as the data-generating model, one obtains a second set of NBS many GOF differences. These two sets of GOF differences constitute two distributions, $dist_A$ and $dist_B$, respectively, that provide information on how well the two models are able to mimic each other (see Figure 1).

In particular, the two distributions can inform model comparison and selection. Distribution $dist_A$ allows to gauge how likely it is to obtain the models' GOF difference on the observed data, $\Delta gof_{AB}^{\mathbf{x}}$, if model A is the generating model. Distribution $dist_B$ allows to gauge how likely it is to obtain $\Delta gof_{AB}^{\mathbf{x}}$, if model B is the generating model. The PBCM selects the model that is associated with the distribution under which $\Delta gof_{AB}^{\mathbf{x}}$ is more likely: If $\Delta gof_{AB}^{\mathbf{x}}$ is more likely under $dist_A$, model A is selected; otherwise, model B is selected.

A crucial question related to the PBCM as described so far is how to sample the parameter values for the data-generating model in step 1 above. Wagenmakers et al. (2004) propose two different ways of generating parameter values. First, parameter values can be determined based on the data that is to be modeled, $\mathbf{x}$. This approach gives rise to the first variant of the PBCM, the DIPBCM. Second, parameter values may be generated independently of $\mathbf{x}$, which yields the variant called DUPBCM. Both variants are described in more detail in the subsequent sections.

DIPBCM

In this variant of the PBCM, parameter values are generated by fitting the data-generating model to bootstrap samples of the observed data $\mathbf{x}$. A bootstrap sample is a new data set $\mathbf{x}^*$ that is obtained by sampling with replacement n values from $\mathbf{x}$, where n is the number of observations in $\mathbf{x}$ (see Efron & Tibshirani, 1993, for more details on bootstrapping). Fitting a model, say A, to a bootstrap sample $\mathbf{x}^*$ yields a set of parameter values that provides the best fit to $\mathbf{x}^*$. These parameter values are then used to generate data from A in step 2 of the PBCM procedure (see above). Drawing NBS many bootstrap samples from $\mathbf{x}$ thus allows to generate NBS many sets of parameter values for each of the two models.

DUPBCM

To generate parameters in the DUPBCM, one first fixes a range of possible values for each model parameter and a probability distribution across each of these ranges. Values for generating data are then sampled from the ranges according to the associated distributions. Uniform distributions were used for all parameter ranges of all models in the simulations reported below.

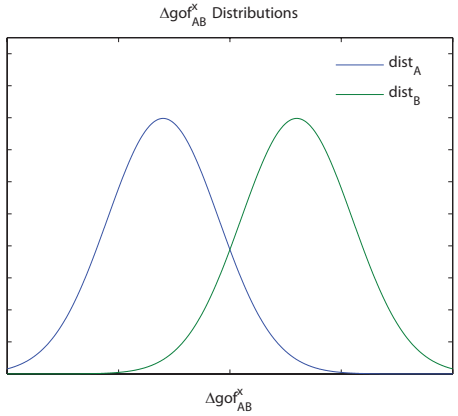

Figure 1: GOF difference distributions as they may arise in using the PBCM. The blue / green curve indicates GOF differences obtained when model A / B have generated data.

Multi-Model PBCM

Wagenmakers et al. (2004) have introduced the PBCM as a method for comparing *pairs* of models. While computing GOF differences works well when considering pairs of models, a comparable measure for 3 or more models does not carry much meaningful information about the relation between the models. Accordingly, it is not immediately clear whether and, if yes, how the PBCM may be extendable to the direct comparison of more than two models.

In the following we propose an extension of the PBCM that allows to compare, in principle, arbitrarily many competing models. The key to our extension lies in taking a different view on the classification problem involved in the PBCM. In its original formulation, use of the PBCM requires solving a classification problem with two classes (data generated from model A vs. data-generated from model B) that contain 1-dimensional instances (the GOF differences): $\Delta gof_{AB}^{\mathbf{x}}$ needs to be assigned to either of the two classes. An alternative but closely related classification problem is to treat the pairs of GOF values from which the differences are computed in the original PBCM as instances of the two classes. Thus, instead of classifying a single value ($\Delta gof_{AB}^{\mathbf{x}}$) on the basis of two 1-dimensional distributions, the alternative problem consists of classifying a value pair, $(gof_A^{\mathbf{x}}, gof_B^{\mathbf{x}})$, on the basis of two 2-dimensional distributions (see Figure 2). This type of classification problem is easily extendable to more than two models: For each additional model a new class is added and the dimensionality of the instances increases by one.

More generally, for a set of k models, $M_1, \ldots, M_k$ ($k \geq 2$) our PBCM extension, the *multi-model* PBCM (MMPBCM) proceeds as follows:

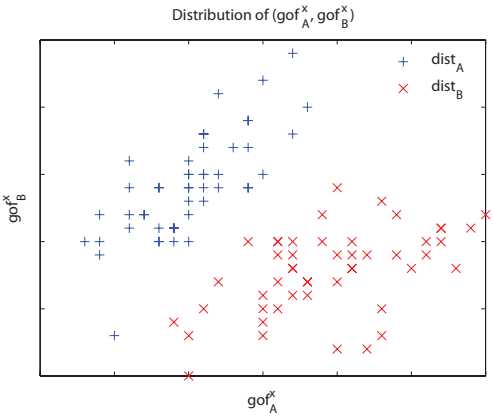

Figure 2: Two-dimensional GOF distributions for comparing two models with the MMPBCM. Blue '+' / red cross indicate GOF pairs when model A / B generated data.

1. generate a set of parameter values for all parameters of model M_1,
2. generate a data set $\mathbf{x}_{M_1}$ by running model M_1 with the parameter values from the first step,
3. fit all models to $\mathbf{x}_{M_1}$ to obtain the GOF vector $(gof_{M_1}^{\mathbf{x}_{M_1}}, \ldots, gof_{M_k}^{\mathbf{x}_{M_1}})$,
4. repeat the above three steps *NBS* number of times,
5. repeat the above 4 steps for models $M_2, \ldots, M_k$,
6. determine the class of $(gof_{M_1}^{\mathbf{x}}, \ldots, gof_{M_k}^{\mathbf{x}})$ based on the resulting k-dimensional distributions and select the model that is the data-generating model of the class.

The ability of the MMPBCM to directly compare arbitrarily many models comes at the cost of a potentially increased complexity of the involved classification problem. Multidimensional multi-class classification can be a hard problem that is often considered to require sophisticated classification methods (Duda, Hart, & Stork, 2001). In particular, many of the comparatively simple methods to classify GOF differences in the PBCM (see Schultheis & Singhaniya, 2013, for an overview) are not feasible for use in the MMPBCM. This creates a tension w.r.t. the aim to foster use of the PBCM for model evaluation, because not all cognitive modelers can be expected to be experts in classification.

To address this potential issue of the usability of the MMPBCM, we tested its performance with three different classifiers: *k-Nearest Neighbor* (k-NN), *Decision Tree* (DT), and *Artificial Neural Network* (ANN). The k-NN was chosen, because it is one of the simplest – if not the simplest – existing multi-class, multi-dimension classifier that has often been reported to yield good results. Due to its simplicity it is easy to implement and use even for the non-expert. The DT and the ANN were chosen because they are two well-known and well-established classifiers. In particular, both of them are available from standard classification libraries (e.g., WEKA, Hall et al., 2009) and thus do not require implementation to be used for the MMPBCM. The DT is also comparatively straightforward in use, but requires more decisions on settings to be made than the k-NN. The ANN is more complex in its use than any of the other two classifiers. It requires decisions on a number of aspects that may be crucial for performance but may be hard to make for non-experts (e.g., regarding network structure, learning algorithm, and stopping criterion). One crucial question to be answered by the simulations reported below is to what extent the use of a more complex classifier pays off in the form of better model comparison performance when employing the MMPBCM.

Substantial detail on the workings and properties of all three employed classifiers can be found in Duda et al. (2001). Below we list the settings and procedural detail associated with each of the classifiers in our simulations.

k-NN This classifier only requires setting k, the number of nearest neighbors to be considered in classification. Our simulations used $k = 10$, because this value yielded near optimal classification accuracies across different PBCM situations in a previous study (Schultheis & Singhaniya, 2013).

ANN The ANN used one hidden layer with 10 units. For training, the available data was partitioned into training data (75%), validation data (15%), and test data (10%). The initial weights were randomly chosen from $[-0.5, 0.5]$ and then trained using backpropagation with a learning rate of 0.1. This learning rate was chosen to optimize performance on (a) standard classification problems and (b) data that was similar to data arising during our simulations. The network was trained in batch mode for 300 epochs and the weights that gave minimum validation error across these 300 epochs were employed for classification.

DT The number of divisions in each dimension was set to 100 and the information gain criterion was used for splitting at each node: A given node was split further only when this yielded an information gain greater or equal to 0.01. This threshold yielded better performance than a value of 0.1 and as good results as a values of 0.001. Since a value of 0.001 would have been more prone to over fitting, we chose a threshold of 0.01

Using these parameterizations each of the classifiers performed very well on five benchmark classification problems from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/).

As with the original PBCM, the MMPBCM can be realized as a data-informed (DIMMPBCM) or a data-uniformed (DUMMPBCM) variant.

Approach

The preceding considerations give rise to 8 different variants of the PBCM: Two variants are the DIPBCM and DUPBCM as originally proposed by Wagenmakers et al. (2004). Both of these employed a k-NN classifier with $k = 10$ for solving the classification problem, because a recent study found that the k-NN yields near optimal results in many situations (Schultheis & Singhaniya, 2013). The remaining six are the

DIMMPBCM and the DUMMPBCM each combined with all three classifiers.

To assess the performance of the different variants of the PBCM, we conducted model recovery studies with three hypothetical models of memory decay. Each of these models, $M1$, $M2$, and $M3$, predicts the probability of recall in dependance on the time that has passed since the to-be-remembered items have been learned. The models are defined as follows (see Pitt & Myung, 2002):

$$M1 : (1+t)^{-a}, a \in [0,2]$$
$$M2 : (b+t)^{-a}, a \in [0,2], b \in [1,2]$$
$$M3 : (1+bt)^{-a}, a \in [0,2], b \in [0,2]$$

All 8 PBCM variants were applied to all 3 possible pairs of models, $M1$ vs. $M2$, $M1$ vs. $M3$, and $M2$ vs. $M3$. Furthermore, the six MMPBCM variants were applied to the full set of all 3 models and compared to the performance of an existing multi-model comparison method: The bootstrap method for model comparison (Efron & Tibshirani, 1997). Despite the similarities in names the bootstrap method of Efron and Tibshirani (1997) works quite differently than the PBCM. To gauge to what extent any of the methods outperforms the naïve approach (i.e., selecting the model with the best fit), we also applied this approach, called *simple recovery* (SR), to all sets of competing models.

Given a set of competing models (i.e., either one of the model pairs or the complete set of the three models), the comparison methods were applied using the following procedure. For one of the competing models, first, a set of parameter values was determined by sampling uniformly from the parameter ranges. Second, probabilities for this set of parameter values were generated from the model. Third, these probabilities were used to randomly sample the number of successful recalls from a binomial distribution assuming a certain number of learned items. Fourth, this set of numbers of successful recalls was treated as if it was a set of empirical observations for which to identify the most appropriate model. Accordingly, the comparison method in question was applied to the set of competing models and the observations. Fifth, which of the compared models was found to be more appropriate was noted. This procedure was repeated $R = 100$ times for each model in the set of competing models. Across all sets of competing models and methods the measure to assess model fits was always the mean squared error and the models were fit using a variant of the Metropolis algorithm (Madras, 2002).

Our simulations varied 4 factors that potentially impact the performance of the comparison methods. These factor variations ensured a more comprehensive view on the methods' performance, that is, a view that is not specific to only one particular combination of factor levels. The considered factors are tightness of fit, strength of noise, number of data points and the number of bootstrap samples and are described in the following.

Tightness of fit Model fits may often be suboptimal to a greater or lesser extent. In view of this, we considered three levels of tightness of fits (loose, medium, and tight fit) by varying how thoroughly the Metropolis algorithm searches the models' parameter space. More precisely, we varied the number of sets of parameters that were sampled (called *swaps*) for model fitting. Three different swaps values, 100, 1000, and 10000, were used.

Strength of noise Since the only noise in the data is sampling noise, the amount of noise in the data is determined exclusively by the number of learned items (NL). The higher NL, the lower is the influence of sampling noise. We employed $NL = 5, 50,$ and 1000 in our simulations.

Number of data points The amount of data available to evaluate and compare competing models may vary considerably depending on the modeling situation. Accordingly, we also varied the number of data points (NDP), using $NDP = 5, 20,$ and 100.

Number of bootstrap samples The number of bootstrap samples (called NBS) determines the amount of information about the compared models that is generated in the scope of applying the methods. Two different NBS values, 100 and 1000, were used.

The combination of all factor variations yields 54 different modeling situations. Assessing the performance of the 8 PBCM variants and the bootstrap method across modeling situations and different sets of competing models allowed obtaining information on a number of key issues related to the employment of the (MM)PBCM. First, it allowed quantifying the performance differences between the data-informed and the data-uniformed variants of the (MM)PBCM. Second, it allowed determining to what extent the type and complexity of the employed classifiers has a substantial (if any) impact on the performance of the PBCM. Third, it allowed evaluating the MMPBCM by both comparing its performance on model pairs to the original PBCM and comparing its performance on all 3 models to an established method for model comparison (i.e., the bootstrap).

Results

The main performance measure we used was the percentage of correct recovery averaged across all models in the competition set. For sets with two / three models, a value of 50% / 33% signifies chance performance and for all model sets, a value of 100% signifies optimal performance.

This performance characteristic was computed for each of the considered modeling situations thus obtaining 54 performance measurements for each method-set of models combination. To convey an impression of the central tendency and variability of each method's performance, we computed the first, second, and third quartile of the performance characteristic across modeling situations. These quartiles and associated standard errors are displayed in Figures 3 and 4. The figures highlight marked performance differences between the

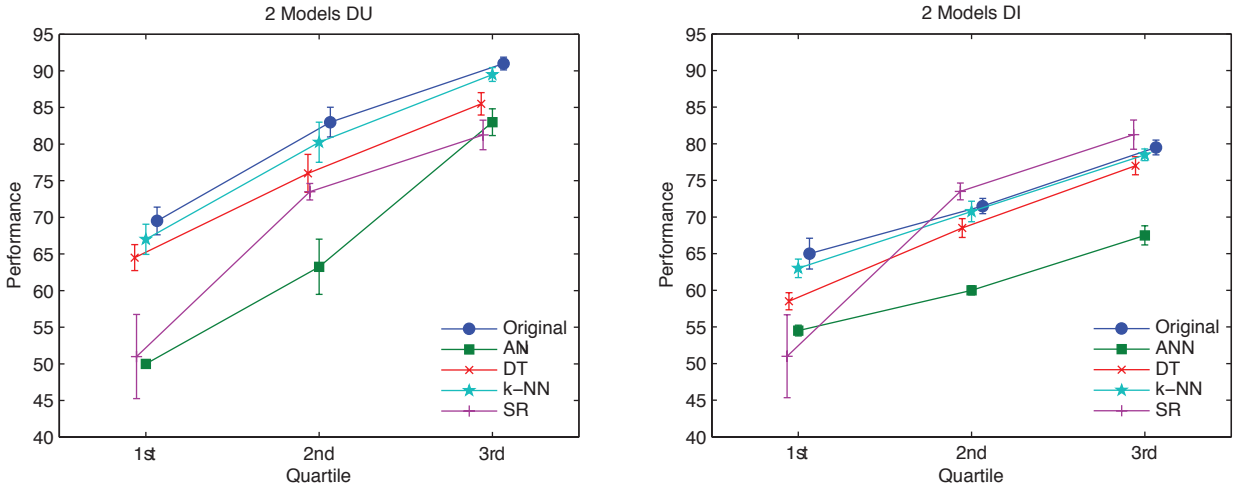

Figure 3: Quartiles with standard errors of model recovery performance for model pairs across modeling situations. Left / Right panel shows DU / DI variants of (MM)PBCM. ANN: Artificial Neural Network classifier; DT: Decision Tree classifier; k-NN: k Nearest Neighbor classifier; SR: Simple Recovery

methods and in the following we discuss these differences with respect to the three key issues mentioned above.

Data-Informed vs. Data-Uninformed

The data-informed variants of the (MM)PBCM perform worse than the corresponding data-uninformed variants. For model pairs (Figure 3), the performance measure of the DU variants is consistently higher than of the DI variants across all quartiles. This indicates a general advantage of the DU variants that is not specific to a small subset of the considered modeling situations. The same holds for the set of 3 competing models (Figure 4): The DIMMPBCM performs generally – and sometimes considerably – worse than the DUMMPBCM across a large proportion of modeling situations.

While one may have expected that the DI variants perform worse than the DU variants given the analyses of Wagenmakers et al. (2004), the extent of this difference may be more pronounced than anticipated. In fact, the DI variants rarely perform better and frequently worse than simple recovery. In contrast, the DU variants consistently outperform the naïve approach. Considering that the computational complexity of the DI variants is higher than the complexity of the DU variants, our results strongly suggest not to employ the DI variants for model comparison and selection.

Classifier Performance

For each set of competing models, the k-NN classifier performs best, the DT classifier second best, and the ANN classifier worst. Particularly noteworthy is the poor performance of the ANN classifier, which rarely performs reliably above chance and never better than the simple recovery method. This is so much more surprising since each classifier performed very well on standard classification problems. Furthermore, the poor model recovery performance of the ANN persisted across different learning parameter settings and stopping criteria. It is currently not clear why the ANN classifier should perform well on standard benchmark problems, but should fail in the scope of the MMPBCM.

In sum, our simulations found no evidence that more complex classifiers yield better model comparison results: The simplest employed classifier, the k-NN, performed best. Even if the poor performance of the ANN were due to inapt application, this would shed doubt on the usability of the ANN for cognitive modelers that are not experts in classification.

MMPBCM Performance

At the heart of the ability of the MMPBCM to compare more than 2 models is the transformation of the classification problem in the original PBCM into a closely related but different classification problem. Therefore, assessment of the MMPBCM should test to what extent the transformation of the classification problem impacts model comparison performance. The quartile plots in Figure 3 suggest that there is no reliable difference between the original PBCM and the best performing MMPBCM for both DI and DU variants. There seems to be a tendency towards slightly better performance by the original PBCM, but given the standard errors associated with the quartile estimates the PBCM variants do not clearly outperform the MMPBCM variants.

Evaluation of the MMPBCM regarding comparisons of more than 2 models is less straightforward, because there is no original available in this case. What then would be a reasonable comparison? If one employs another, arbitrarily chosen, model comparison method $\mathcal{M}$, the results may be hard to interpret. If, for example, $\mathcal{M}$ outperforms the MMPBCM, does that imply (a) that the multi-model extension of the PBCM is somehow flawed or (b) that the PBCM as such is inferior to $\mathcal{M}$? To avoid such ambiguities, we chose the bootstrap method proposed by Efron and Tibshirani (1997) for comparison with the MMPBCM, because our previous inves-

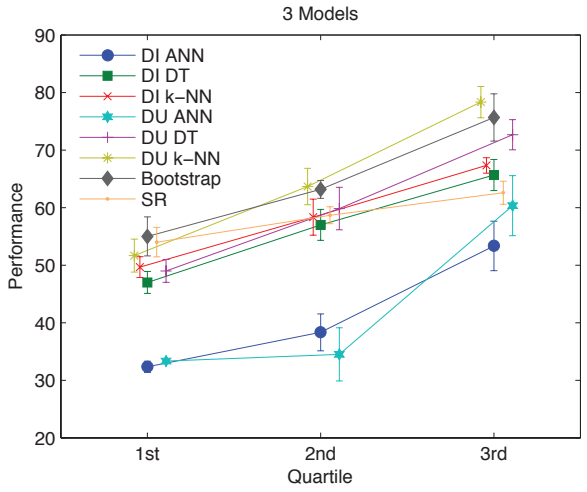

Figure 4: Quartiles with standard errors of model recovery performance for three models across modeling situations. ANN: Artificial Neural Network classifier; DT: Decision Tree classifier; k-NN: k Nearest Neighbor classifier; SR: Simple Recovery

tigations indicated that this bootstrap method and the PBCM perform very similarly (Schultheis et al., 2013).

In line with these previous observations and as shown in Figure 4, the DUMMPBCM and the bootstrap perform virtually identical for the set of three competing models. Accordingly, assessment of the MMPBCM on model pairs as well as on the 3-model set support the validity of the MMPBCM.

Conclusion

The MMPBCM proposed in this contribution appears to constitute a more powerful model comparison method than the original PBCM. In addition to performing similarly to the PBCM on 2-model comparisons it also performs well on comparisons outside the reach of the PBCM: Direct comparisons of more than 2 models. Our simulations furthermore indicated (a) that the MMPBCM yielded best results when employing the simplest classifier (k-NN) and (b) that the computationally more complex data-informed variant of the (MM)PBCM did perform considerably worse – not even better than the naïve approach – than the less complex data-uninformed variant. Accordingly, the data-uninformed MMPBCM provides a generally applicable model comparison method, which takes model complexity into account, has comparatively low computational cost, and is easy to use.

Our future work aims to provide a broader and more solid basis for these conclusions by investigating the performance of the (MM)PBCM on additional sets of competing models. A further interesting issue concerns a possible relation between parameter generation and (MM)PBCM performance. While the data-uninformed variants use the same distribution (uniform) for sampling parameters as employed for generating data from the "true" model, the data-informed variants use a non-uniform distribution for sampling. It seems interesting to examine to what extent the poor performance of the data-informed variants can be explained by this difference.

Acknowledgments

This work was done in the project R1-[ImageSpace] of the SFB/TR 8 Spatial Cognition. Funding by the German Research Foundation (DFG) is gratefully acknowledged. We thank the reviewers for insightful and helpful suggestions.

References

Cohen, A. L., Rotello, C. M., & MacMillan, N. A. (2008). Evaluating models of remember–know judgments: Complexity, mimicry, and discriminability. *Psychonomic Bulletin & Review*, *15*, 906-926.

Cohen, A. L., Sanborn, A. N., & Shiffrin, R. M. (2008). Model evaluation using grouped or individual data. *Psychonomic Bulletin & Review*, *15*, 692-712.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. New York: Wiley.

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York: Chapman & Hall.

Efron, B., & Tibshirani, R. J. (1997). Improvements on cross-validation: The .632+ bootstrap method. *J Am Stat Assoc*, *92*, 548-560.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explorations*, *11*(1).

Jang, Y., Wixted, J. T., & Huber, D. E. (2011). The diagnosticity of individual data for model selection: Comparing signal-detection models of recognition memory. *Psychonomic Bulletin & Review*, *18*, 751-757.

Madras, N. (2002). *Lectures on monte carlo methods*. Providence, Rhode Island: American Mathematical Society.

Perea, M., Gomez, P., & Fraga, I. (2010). Masked nonword repetition effects in yes/no and go/no-go lexical decision: A test of the evidence accumulation and deadline accounts. *Psychon B & Rev*, *17*, 369-374.

Pitt, M., & Myung, J. (2002). When a good fit can be bad. *Trends in Cognitive Sciences*, *6*, 421-425.

Schultheis, H., & Singhaniya, A. (2013). Decision criteria for model comparison using cross-fitting. In *22nd Annual Conference on Behavior Representation in Modeling & Simulation (BRiMS 2013)*.

Schultheis, H., Singhaniya, A., & Chaplot, D. (2013). Comparing model comparison methods. In M. Knauff, M. Pauen, N. Sebanz, & I. Wachsmuth (Eds.), *Proceedings of the 35th annual conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.

Shiffrin, R. M., Lee, M. D., Kim, W., & Wagenmakers, E.-J. (2008). A survey of model evaluation approaches with a tutorial on hierarchical bayesian methods. *Cognitive Science*, *32*, 1248-1284.

Wagenmakers, E.-J., Ratcliff, R., Gomez, P., & Iverson, G. (2004). Assessing model mimicry using the parametric bootstrap. *Journal of Mathematical Psychology*, *48*, 28-50.