RESEARCH

TITLE GOES HERE

Alexander K Christensen[1*] and Richard S Savage[1,2]

Correspondence:
.kier-christensen@warwick.ac.uk
Systems Biology Centre,
Jniversity of Warwick, CV4 7AL
Coventry, UK
ull list of author information is
vailable at the end of the article
Equal contributor

Abstract

Background: Rapid expansion of the amount of medical data available to the healthcare industry is driving demand for more accurate, precise and reliable tools for exploiting these data. Real-world experience suggests that combining the predictions of individual machine-learning classifiers may allow us to meet all three criteria at a low cost, potentially saving scarce funds and patients' lives. We therefore attempt to go beyond the existing literature on individual machine learning algorithms to discover the effectiveness of combining their predictions, and to explore the potential benefits of such 'ensemble methods' to personalised diagnostic and prognostic medicine

Results: Consideration of a range of classifiers is an important aspect in getting the best predictive performance for a given data set. For each of the 10 biomedical data sets considered in this paper, a range of classifier ensemble methods performed competitively with respect to individual classifiers. Furthermore, it was possible to reliably automate the ensemble methods, making them straightforward to apply to new data sets. It was noteworthy that unsupervised methods such as averaging of prediction probabilities generally performed well despite their simplicity. Strikingly, we found that simply selecting the best-performing individual model was actually a very strong strategy, outperforming most of the model 'blending' approaches leading to best or near-best predictive performance on all 10 data sets.

We conclude that while it is clearly important to consider a range of classifiers for a given data set, simple strategies such as selecting the best-performing bas classifier should not be discounted. We also speculate that the widely observed predictive performance improvements seen in e.g. data science competitions may be more the result of averaging over different set of features and feature representations, rather than over different types of model. We then conclude that there may be significant merit in the development of formal methods for selecting between a wide range of different classifiers.

Keywords: ensemble; machine-learning; diagnostic; medicine; meta-learning

Background

As medicine continues to evolve into a data-rich discipline, there is an increasing need for machine learning methods which can take full advantage of the rapidly-growing data volumes that doctors are amassing on their patients and the illnesses which afflict them. In non-medical applications of machine learning, such as the well-known Netflix competition[**?**], it is often the case that incremental gains in performance are not worth the extra computational cost, but the nascent field of personalised medicine is particularly well-placed to benefit from even relatively modest improvements in predictive performance, since early and precise identification of a disease can save both treatment costs and patients' lives.

Ensemble methods have been shown to offer noticeable performance gains over individual classification algorithms[1][2][3], however it is unclear precisely from what aspect of the ensembling this improved performance stems. It is therefore important to determine more precisely if and when these gains can be exploited by the medical community.

There exists already a significant amount of literature studying the variety and effectiveness of individual machine learning classifiers (see [**?**] for one exhaustive example). In practise, real-world results suggest that it is usually possible to obtain a statistically significant boost in predictive ability by combining the predictions from a number of different learners. Such combination methods (which we shall refer to as "ensemble methods" or "ensemble classifiers") exist in many forms, from simple averaging of predictions or majority voting, to more complex so-called "meta-learning" approaches. Our task is to obtain data on the performance of a variety of such ensemble classifiers on a collection of medical datasets, in order to evaluate whether or not ensembling does in general offer a genuine performance increase.

Methods

We aim to assess the effectiveness of different combination strategies compared to each other and to the individual classification algorithms whose predictions they employ. We focus on datasets of a medical nature in order to obtain results specifically relevant to the field of diagnostic and prognostic medical machine learning.

We have assembled a collection[1] of datasets from the UCI machine learning repository, selected for meeting the following conditions:

- Data of a medical nature
- Binary outcome variable
- At least 250 instances without missing data

We use a collection of 11 base classifiers (drawn from a variety of existing R packages) and 7 ensemble classifiers (both existing and of our own implementation) to train and obtain predictions. The predictions of each ensemble classifier on each dataset are evaluated by the AUC metric to determine their effectiveness compared to each other and to the best performing base classifier on that dataset.

All of the classifiers with the exception of the Independent Bayesian Classifier Combination (IBCC) ensembles were implemented in R. IBCC was run in Python, using the implementation given in [**?**]. In order to avoid bias, predictions were created independently before being passed to any ensemble classifier, and all were given the same set of predictions for each data set, whether via R or via Python.

Datasets

We used the following data sets in our comparison, all taken from the UCI machine learning repository[**?**].

1 **Breast Cancer Wisconsin (Original)** (1991): classification of tumours into 'benign' or 'malignant'. Courtesy of Dr. William H Wolberg, Universiy of Wisconsin Hospitals, Madison.

[1]These datasets also form part of the collection of datasets in Fernandez-Delgado's paper [**?**] comparing base classifiers.

2 **Breast Cancer Wisconsin (Diagnostic)** (1994): classification of tumours into 'benign' or 'malignant'. Courtesy of Dr. William H. Wolberg, University of Wisconsin, Clinical Sciences Center, Madison.

3 **Haberman's Survival Dataset** (1999): classification of patients undergoing breast cancer surgery into 'deceased within 5 years' or 'survived at least 5 years'. Courtesy of University of Chicago's Billings Hospital & Tjen-Sien Lim, Department of Statistics, University of Wisconsin, Madison.

4 **Heart Disease Dataset (Hungarian)** (1988): classification of patients into 'presence' or 'absence' of heart disease. Courtesy of Andras Janosi, M.D., Hungarian Institute of Cardiology, Budapest & David W. Aha, Institute of Information and Computer Science, UCI.

5 **Indian Liver Patient Database** (2012): classification of patients into 'liver patient' or 'non liver patient'. Courtesy of B. V. Ramana & Prof N. B. Venkateswarlu, Aditya Institute of Technology and Management & Prof M. S. Prasad Babu, Andhra University College of Engineering.

6 **Mammographic Mass Dataset** (2007): classification of mammographic masses into 'benign' or 'malignant'. Courtesy of Matthias Elter, Fraunhofer Institute for Integrated Circuits, Erlangen, Germany & Dr Rudiger Schulz-Wendtland, Institute of Radiology, University Erlangen-Nuremberg, Germany.

7 **Single Proton Emission Computed Tomography (SPECT) Heart-imaging Dataset** (2001): classification of SPECT heart images into 'normal' or 'abnormal'. Courtesy of Krzysztof J. Cios & Lukasz A. Kurgan, University of Colorado, Denver & Lucy S. Goodenday, Medical College of Ohio, Ohio.

8 **SPECTF Heart-imaging Dataset** (2001): classification of SPECTF heart images into 'normal' or 'abnormal'. Courtesy of Krzysztof J. Cios & Lukasz A. Kurgan, University of Colorado, Denver & Lucy S. Goodenday, Medical College of Ohio, Ohio.

9 **Statlog (Heart) Dataset** : classification of patients into 'presence' or 'absence' of heart disease. Source anonymised by UCI.

10 **Vertebral Column Dataset** (2011): classification of orthopaedic patients into 'normal' or 'abnormal'. Courtesy of Guilherme de Alencar Barreto & Ajalmar da Rocha Neto, Department of Teleinformatics Engineering, Federal University of Ceará, Fortaleza, Brazil & Dr. Henrique da Mota Filho, Hospital Monte Klinikum, Fortaleza, Brazil.

The following table provides further details of each of the above datasets, each of which is henceforth referred to by a shortened label or acronym, in the interest of compactness.

Data Conversion & Processing

Missing data have been removed from the datasets, either by removing only the single instance with the missing data point or by removing an entire variable in the few cases where data for a specific variable is missing for all but a very small number of instances.

In the cases where non-numeric values are given (e.g. M/F to represent gender) a simple numeric conversion is applied: if a variable x can attain (discrete) non-

datasets	#instances	#variables	Majority%
bc-wisc-original	683	9	52.2
bc-wisc-diag	569	30	62.7
haberman-survival	306	3	73.5
heart-hungary	261	10	62.5
ilpd-indian-liver	579	10	71.5
mammographic	830	4	51.4
spect	267	22	79.4
spectF	267	44	79.4
st-heart	270	13	55.6
vertebral-col	310	6	67.7

Table 1 Our collection of 10 datasets from the UCI repository. Columns represent the number of instances, number of variables and the percentage of the majority class for each of the datasets after missing data is removed. Further detail on each dataset can be found in the list following the table.

numeric values $\{a_1, a_2, ..., a_n\}$ then the value a_i is converted to the number $i \in \{1, ..., n\}$.

Before being passed to the k-NN base classifier, each variable is processed to have zero mean and standard deviation one. Since k-NN is a distance-based classifier, variables will be weighted according to the size of the range in their numerical values, rather than according to any genuine underlying medical importance. We therefore apply this pre-processing to 'smooth out' this weighting more uniformly. We apply the same normalisation before passing data to the Neural Network base classifier, following best practise for this kind of classifier. We do not use further pre-processing, data transformation or feature selection.

Base Classifiers

We use the following Base Classifiers, implemented in R, to obtain predictions on which to train and test ensemble methods.

1 **Sparse Logistic Regression**, using *cv.glmnet* in the *glmnet* package. We run with the flags *maxit=1e5, alpha=1, family="binomial"*.

2 **Random Forest**, using *randomForest* in the *randomForest* package. We run with the flags *importance=TRUE, ntree=1500*.

3 **Generalised Boosting Model**, using *gbm* in the *gbm* package. Flags *distribution="bernoulli", ntrees=3000, interaction.depth=4, cv.folds=3, n.cores=2, n.minobsinnode ¡ 10*.

4 **Gaussian Process**, using *gausspr* in the *kernlab* package. We use set.seed to obtain consistent results in the interests of obtaining results that reflect the classification ability of the ensembling methods, not of the underlying base classifiers. Flags *kernel="rbfdot"*.

5 **Support Vector Machine**, using *ksvm* in the *kernlab* package. Flags *kernel="rbfdot", prob.model=TRUE, kpar=list(sigma=0.05), C=5, cross=3*.

6 **Neural Network**, using *neuralnet* in the *neuralnet* package. Flags *hidden=1, threshold=0.01, linear.output=FALSE*.

7 **Decision Tree**, using *C5.0* in the *C50* package. Flags *trials=1, rules=FALSE*.

8 **Rule-Based Method**, using *C5.0* in the *C50* package. Flags *trials=1, rules=TRUE*.

9 **k-Nearest Neighbour**, using *knn3* in the *caret* package. Flags *k=40*.

10 **Naive Bayes**, using *NaiveBayes* in the *klaR* package.

11 **Linear Discriminant Analysis**, using *lda* in the *MASS* package.

Ensembling Process

Each dataset is partitioned randomly into three subsets:

1 Btrain (base classifier training set, 50% of data instances)

2 Mtrain (ensemble classifier training set, 25% of data instances)

3 Mtest (ensemble classifier testing set, 25% of data instances)

The ensemble predictions are obtained through the following procedures:

Ensemblers with no meta-learning stage: First, the base classifiers are trained on the combined set Btrain ∪ Mtrain, and predictions for Mtest are obtained. These predictions are passed to the ensemble classifier, to combine into a final set of predictions.

Ensemblers with a meta-learning stage: First, the base classifiers are trained on the set Btrain, and separate predictions for Mtrain and Mtest are computed. The predictions for Mtrain are passed to the ensemble classifier to train the meta-learner, which then is given the predictions for Mtest in order to ensemble them into a final set of predictions.

Ensemble Classifiers

To ensemble the base classifier predictions, we use the following Ensemble Classifiers, implemented in R (with the exception of IBCC, for which the Python implementation in [**?**] is used).

1 **Average**, simple average of the probabilistic base classifier predictions for each instance. No meta-learning stage is needed by this method.

2 **Weighted Average**, as with Average, but with a meta-learning stage in which each contributing base classifier has its prediction weighted according to its AUC score when classifying Mtrain.

3 **Majority Vote**, converts probabilistic base classifier predictions to "votes" (binary 0/1 predictions) by rounding. Then combines the votes for each instance by summing them all and dividing by the number of instances to obtain a prediction between 0 and 1. No meta-learning stage.

4 **Rank Average**, ranks the predictions for each base classifier separately, then normalises these to obtain uniformly spaced values between 0 and 1, before computing a simple average of the ranks for each instance. No meta-learning stage.

5 **Stacking w/ Logistic Regression** uses glm, a logistic regression algorithm from the *stats* package, as a meta-classifier, learning to ensemble the base classifier predictions for Mtrain, and returning ensembled predictions for Mtest. Flags *family="binomial"*.

6 **Stacking w/ Sparse Logistic Regression** uses glmnet from the *glmnet* package, as a meta-classifier, learning to ensemble the base classifier predictions for Mtrain, and returning ensembled predictions for Mtest. Flags *family="binomial", maxit=1e5, alpha=1*

7 **Stacking w/ Random Forest** uses randomForest from the *randomForest* package, as a meta-classifier, learning to ensemble the base classifier predic-

tions for Mtrain, and returning ensembled predictions for Mtest. Flags *importance=TRUE, ntree=500*

8 **IBCC** (supervised mode), "Bayesian classifier combination, using the computationally efficient framework of variational Bayesian inference."[**?**]. In supervised mode, IBCC acts as a meta-classifier, learning to ensemble the base classifier predictions.

9 **IBCC** (unsupervised mode). In unsupervised mode, IBCC performs unsupervised learning on the predictions obtained by training the base classifiers on the combined set Btrain ∪ Mtrain, thereby learning to ensemble base classifier predictions.

Comparison Metric

The performance of the classifiers is evaluated using the AUC metric, implemented in R using the roc function from the `pROC` package. For each classifier we use this function to obtain the AUC and a 95% confidence interval. Tables including all of these data are included at the end of the document, in Tables **??** & **??**.

Results & Discussion

In our experimental analysis we evaluate 9 ensemble methods, combining the predictions of 11 base classifiers over 10 datasets. We find that one of our base classifiers (NaiveBayes) strictly requires non-zero variances for all variables, and therefore will return errors when analysing the spect (but not spectF) dataset. We therefore omit the NaiveBayes base classifier when obtaining predictions for the spect dataset.

Figure **??** displays the AUC scores of each base classifier on each dataset. We include this for two principal reasons; firstly to demonstrate that the base classifiers are reasonably effective on their own, and therefore our ensembles are not "polluted" by including routinely impotent base predictions, and secondly to demonstrate that there is nonetheless a significant range in performance across datasets and classification algorithms. As the well known "No Free Lunch" Theorems[4] would lead us

to expect, we do not observe that any particular base classifier performs best on every dataset (or even a significant majority of them), but there are naturally some which tend to perform well by comparison to the rest.

Figure **??** displays the AUC scores of each ensemble classifier on each dataset. It was to be expected (once again in accordance with the "No Free Lunch" principle) that no individual ensemble method would prove consistently superior to the others. Our aim, therefore, is rather to determine whether or not it is usually possible to obtain an improvement in accuracy over the base classifiers by implementing *some kind* of ensembling (i.e. is there usually at least one ensemble classifier which outperforms the best base classifier?). Conventional wisdom and the results from many real-world applications of machine learning (in particular competitions such as the Netflix Prize[**?**] and Kaggle competitions[**?**]) suggest that a noticeable improvement is usually possible using even relatively unsophisticated ensembling techniques.

We are therefore surprised to note that our results suggest that the best base classifier surpasses the accuracy of every ensemble classifier on 5 of our 10 datasets, and equals the accuracy of the best ensemble classifier on a further 2 datasets. Stacking with some form of logistic regression as the meta-learner is the only ensembling approach that manages to outperform the best base classifier on any of the datasets. This suggests that perhaps ensembling for model uncertainty as we have done here is less important than selecting the best base classifier for a given task (as stacking is broadly attempting to do).

We observe also a few general trends:

1 Average and Stack SLR are the stand-out best performers amongst the ensemble methods, with Rank Average also showing strong results.

2 Average performs in the top 3 ensemble classifiers on 8 out of 10 datasets (although it never outperforms the best base classifier) and does not perform catastrophically on any of the datasets (unlike some of the more sophisticated

ensemble methods), displaying a respectable level of consistency. These observations are also true to a slightly lesser extent for the similar Rank Average ensemble.

3 Stacking with sparse logistic regression (as in Stack SLR) significantly outperforms stacking with logistic regression (as in Stack LR) on every dataset except st-heart. It seems that there is some aspect of these kinds of dataset which Stack SLR is generally better able to capture than Stack LR.

4 Unsupervised IBCC significantly outperforms Supervised IBCC on 7 of our 10 datasets, and on the 2 datasets where Supervised IBCC wins, it does so by a much smaller margin.

5 Along the same lines as the previous point, the ensemble classifiers which forego a meta-learning stage in favour of a larger base classifier training set generally exhibit a respectable performance both in terms of AUC score and consistency.

The contrasting nature of the ensemble methods which include a meta-learing stage and those which do not, leads to a difference in the way that data is fed to each. As explained in Section , for an ensemble classifier which does not require the training of a meta-learning on the Mtrain subset, we opt instead to include Mtrain along with Btrain in the training data for the base classifiers, for two reasons. Firstly, to withold the subset Mtrain completely from these ensemble classifiers would be to impose on them a data deficit in comparison to the meta-learning ensemblers, resulting in an unfair comparison. Secondly, any real-world implementation of these ensemble classifiers would surely aim to make full use of the available data, rather than arbitrarily withholding a 25% subset, so for the purposes of obtaining results that are applicable to real-world best practise, ours is the more reasonable approach. Figure **??** supports these assertions, demonstrating that a larger base classifier training set almost universally results in significantly greater prediction accuracy.

Conclusion

The most immediate conclusion to be drawn from our results is that ensembling for model uncertainty is not always more effective than simply selecting the best base classifier. Indeed, the only ensemble methods to outperform the best base classifier were Stack SLR and Stack LR, both of which work broadly by attempting to 'learn' which are the best base classifiers and then weighting their predictions the most heavily.

How might this be reconciled with the fact that real-world applications often do show a performance boost from ensembling? We note that many such real-world applications (e.g. Kaggle competition winners and the top performers in the Netflix Prize) include varying degrees of feature learning and ensembling over feature sets, rather than our approach of ensembling over only a set of base classifiers, each applied to exactly the same feature set. This suggests that the observed success of ensembling in these cases may have more to do with these alternative approaches to ensembling - a hypothesis which we believe merits future investigation.

Our results suggest also that it is important to consider the 'cost' of some ensemble methods in terms of training data. We see with regards to IBCC that the advantages conferred by meta-learning from labelled training examples in Supervised IBCC appear to be outweighed by the larger volume of training data given to the base classifiers in Unsupervised IBCC. More generally, the relative performance of the meta-learning and non-meta-learning ensemble methods supports the idea that the performance gain from ensembling with a meta-learner may not outweigh the cost of a reduced training set, particularly if the entire dataset is small to begin with, as may often be the case in medical applications. It is therefore clearly important to carefully consider the nature of a dataset, and not just which classifiers to use, when attempting to obtain good predictions.

Finally, we mention the respectable performance of the very basic Average and Rank Average ensemble methods, relative to more complex approaches such as stacking, which perform extremely poorly on a small number of the datasets. With regards to medical applications in diagnosis, for example, it is clear to see why one would prefer a consistently good classifier over an occasionally excellent classifier whose consistency varies greatly, even if the latter is sometimes the best performing classifier of all. Furthermore, as the volume of medical data available to doctors increases, it will become increasingly important to consider the computational cost of machine learning methods, in which simpler methods such as Average and Rank Average are clear winners over Stacking and IBCC.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

Text for this section . . .

Acknowledgements

Text for this section . . .

Author details

[1]Systems Biology Centre, University of Warwick, CV4 7AL Coventry, UK. [2]Warwick Medical School, University of Warwick, CV4 7AL Coventry, UK.

References

1. Opitz, D., Maclin, R.: Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research **11**, 169–198 (1999)
2. Polikar, R.: Ensemble based systems in decision making. IEEE Circuits and Systems Magazine **6**, 21–45 (2006)
3. Rokach, L.: Ensemble-based classifiers. Artificial Intelligence Review **33**, 1–39 (2010)
4. Wolpert, D.H.: The lack of a priori distinctions between learning algorithms. Neural Computation **8**, 1341–1390 (1996)

Figures

Figure 1 Sample figure title. A short description of the figure content should go here.

Figure 2 Sample figure title. Figure legend text.

Table 2 Sample table title. This is where the description of the table should go.

	B1	B2	B3
A1	0.1	0.2	0.3
A2	...	..	.
A3	..	.	.

Tables

Additional Files

Additional file 1 — Sample additional file title

Additional file descriptions text (including details of how to view the file, if it is in a non-standard format or the file extension). This might refer to a multi-page table or a figure.

Additional file 2 — Sample additional file title

Additional file descriptions text.