Proceedings of Machine Learning Research 58 (2016) 1-11

Decision Heuristics for Comparison: How Good Are They?

Marcus Buckmann BUCKMANN@MPIB-BERLIN.MPG.DE
Özgür Şimşek OZGUR@MPIB-BERLIN.MPG.DE
Center for Adaptive Behavior and Cognition
Max Planck Institute for Human Development
Lentzeallee 94, 14195 Berlin, Germany

Editor: T.V. Guy, M. Kárný, D. Rios-Insua, D.H. Wolpert

Abstract

Simple decision heuristics are cognitive models of human and animal decision making. They examine few pieces of information and combine the pieces in simple ways, for example, by considering them sequentially or giving them equal weight. They have been studied most extensively for the problem of *comparison,* where the objective is to identify which of a given number of alternatives has the highest value on a specified (unobserved) criterion. We present the most comprehensive empirical evaluation of decision heuristics to date on the comparison problem. In a diverse collection of 56 real-world data sets, we compared heuristics to powerful statistical learning methods, including support vector machines and random forests. Heuristics performed surprisingly well. On average, they were only a few percentage points behind the best-performing algorithm. In many data sets, they yielded the highest accuracy in all or parts of the learning curve.

1. Introduction

People and animals spend much of their time choosing among alternative options. For example, a venture capitalist chooses among companies to invest in, a writer among potential publishers, and a bee colony among suitable nest sites. The true values of the alternatives are typically not known in advance. A choice is made by examining other pieces of relevant information. For instance, venture capitalists do not know how much they will earn from investing in a particular company but can examine the track record of the founders.

How do people and animals make such decisions? One theory is that they use simple heuristics (Gigerenzer et al., 1999). These simple decision rules examine only a few pieces of information, perhaps only a single piece of information, and combine the pieces in simple ways. For example, lexicographic heuristics consider the various pieces of information sequentially, one at a time, while tallying heuristics give different pieces of information equal weight. There is evidence supporting the use of such simple models in a wide range of decisions made by people and animals (Gigerenzer et al., 2011; Hutchinson and Gigerenzer, 2005).

Compared to standard statistical decision methods, heuristics are very frugal in their use of information and have very low computational requirements. And yet earlier studies have shown that heuristics can be surprisingly effective when compared to statistical

learning methods, including logistic regression, decision trees, naive Bayes, and nearest neighbor methods (Brighton and Gigerenzer, 2008; Martignon and Laskey, 1999; Şimşek and Buckmann, 2015).

Intrigued by these earlier results, we systematically analyzed the performance of heuristics compared to the very best statistical models, including random forests and support vector machines (SVMs), the two models that performed best in a recent large-scale comparison of 179 classification algorithms across 121 data sets (Fernández-Delgado et al., 2014). In this study, of the five highest ranked classifiers, three were random forest implementations (first, second, and fifth position) while the other two were SVM implementations.

We analyzed a diverse collection of 56 real-world data sets that included two well-known heuristics, take-the-best (Gigerenzer and Goldstein, 1996) and tallying (Czerlinski et al., 1999). Heuristics performed remarkably well. On average, they were only a few percentage points behind the best performing algorithm. In many data sets, they yielded the highest accuracy in all or parts of the learning curve.

In the following sections, we first formally define the decision problem we address and describe decision models that are based on heuristics, classification, and regression. We then provide an overview of earlier results on how well heuristics perform. We continue with a description of our methodology and results. We conclude with a discussion of our findings.

2. Background

The decision problem we address is *comparison,* where the objective is to identify the alternative with the highest criterion value, given m alternatives and k attributes on each alternative. We focus on problems with exactly two alternatives. An example is to determine which of two stocks will have a higher return on investment in 5 years, given attributes such as the name recognition of the company.

Let A and B denote the first and the second alternative, respectively. Let $\mathbf{x}_A$ denote the vector of attribute values of alternative A, and y_A its criterion value. The outcome variable of interest is $o_{AB} = sgn(y_A - y_B) \in \{-1, 0, 1\}$, where sgn is the mathematical sign function. The objective is to construct a decision rule for selecting one or the other alternative using the available attributes, in other words, to learn a decision rule $f(\mathbf{x}_A, \mathbf{x}_B) \in \{-1, 0, 1\}$ from training data $T = \{\mathbf{x}_A^i, \mathbf{x}_B^i, o_{AB}^i\}_{i=1}^N$.

The comparison problem is intrinsically symmetrical. Comparing A to B should return the same decision as comparing B to A. That is, $f(\mathbf{x}_A, \mathbf{x}_B)$ should equal $-f(\mathbf{x}_B, \mathbf{x}_A)$. One can expect better accuracy when imposing this symmetry constraint on the learning method.

Notice that an outcome value of 0 may be useful for training but the learned model need not identify this outcome at all. From the decision maker's perspective, when alternatives have equal value, either alternative would qualify as a correct decision.

In the heuristics literature, attributes are called *cues*; we follow this custom when discussing heuristics.

Below, we describe three approaches to comparison and comment on the informational needs of the various approaches.

2.1 Decision heuristics

We consider three heuristics: single-cue (Hogarth and Karelaia, 2005; Şimşek and Buckmann, 2015), tallying (Czerlinski et al., 1999), and take-the-best (Gigerenzer and Goldstein, 1996). These heuristics associate each cue with a *direction* to determine how the cue decides on its own. Cue direction can be positive or negative, favoring the object with the higher or lower cue value, respectively. It can also be neutral, favoring neither object. Cue directions can be learned in a number of ways, including social learning. In our analysis, they are learned from training examples.

Single-cue is perhaps the simplest decision method one can imagine. It compares the alternatives on a single cue, breaking ties randomly. A model for this heuristic specifies the identity of the cue and its direction. We learn both from training examples. Specifically, among the $2k$ possible models, where k is the number of cues, the single-cue model is the one that has the highest accuracy in the training examples.

Tallying is a voting model. It determines how each cue votes on its own—for alternative A, for alternative B, or for neither—and selects the object with the highest number of votes, breaking ties randomly. A tally model needs only to specify cue directions.

Take-the-best is a lexicographic model. It considers the cues one at a time, in a specified order, until it finds a cue that *discriminates* between the alternatives, that is, a cue whose value differs on the two alternatives. It then decides based on that cue alone. A take-the-best model specifies cue directions and cue order.

Cue directions are learned in the same manner for all heuristics. The direction of each cue is learned independently of the directions of other cues. The information required from each training example is simply the direction d of the cue in that example: $d = sgn((y_A - y_B) \times (x_A - x_B)) \in \{-1, 0, +1\}$. Let p and n respectively denote the number of positive and negative samples in the training set. Specifically, $p = \sum_{i=1}^{N} I(d_i = 1)$ and $n = \sum_{i=1}^{N} I(d_i = -1)$, where I is the indicator function. Cue direction is set to positive, negative, or neutral, respectively, if $p > n$, $p < n$, or $p = n$.

Cue order in take-the-best is set to the order of decreasing validity of the cues in the training sample, where *validity* is $max\{p/(p+n), n/(p+n)\}$. That is, cues are ordered by how often they decide correctly when they are able to discriminate between the alternatives.

2.2 Classification

Because our outcome variable is discrete, any classification algorithm is directly applicable. In principle, learning can be done as a function of attribute values of individual objects, but in practice the training data required will be prohibitive. Thus we explore learning a decision rule as a function of attribute differences, $\mathbf{\Delta x}_{AB} = \mathbf{x}_A - \mathbf{x}_B$. Our objective then is to learn decision rule $f(\mathbf{\Delta x}_{AB}) \in \{1, -1, 0\}$ from training data $T = \{\mathbf{\Delta x}^i_{AB}, o^i_{AB}\}_1^N$.

2.3 Regression

Regression can be used to estimate the difference in criterion values of the two alternatives and deduce which alternative has the higher criterion value based on these estimates. Specifically, we train a regressor $h(\mathbf{\Delta x}_{AB})$ with training data $T = \{\mathbf{\Delta x}^i_{AB}, (y_A - y_B)^i\}_1^N$ to use in decision rule $f(\mathbf{x}_A, \mathbf{x}_B) = sgn(h(\mathbf{\Delta x}_{AB}))$.

2.4 Data requirements

Regression requires the highest level of information. It requires that the difference in the criterion values of the alternatives be known in training data, which may not always be possible. In contrast, the training data required for classification is more easily available because all that is required is the identity of the alternative with the higher criterion value—the criterion values of either alternative or the difference in their criterion values are not needed.

Informational needs of heuristics are substantially less than those of classifiers. They do not even require the differences in cue values to be quantified; they need only the sign of cue differences. For example, if *height of a person* is a cue, heuristics need to know which of two people is taller but not the height of either person or the magnitude of the difference.

3. Earlier work

Czerlinski et al. (1999) compared take-the-best and tallying to multiple linear regression in 20 real-world data sets. In each data set, the authors trained the models on all pairwise comparisons among 50% of the objects and tested them on all pairwise comparisons among the remaining objects. They dichotomized the numerical attributes around the median, converting the attribute to binary to mimic people's typically limited knowledge about attribute values and the potential unreliability of precise values. Take-the-best performed best, with a mean accuracy of 0.72 across data sets, compared to 0.69 for tallying, and 0.68 for multiple linear regression. When the authors repeated their analysis without dichotomizing the attributes but using their exact numerical values, mean accuracies of take-the-best and multiple linear regression were identical at 0.76. The authors did not test tallying with numerical attributes.

Brighton (2006) presented learning curves on eight data sets, where attributes were again dichotomized around the median, comparing take-the-best to neural networks, decision trees, and nearest neighbor methods. In four data sets, take-the-best had the highest accuracy on almost the entire learning curve. In the other four data sets, take-the-best had the highest accuracy on at least some parts of the learning curve.

Katsikopoulos et al. (2010) showed mean accuracy in 19 data sets for training samples of 2 to 10 objects. They compared multiple models, including take-the-best, tallying, multiple linear regression, and naive Bayes, implementing most models with and without dichotomizing the attributes. Take-the-best (with undichotomized cues) had the highest accuracy for all but the smallest training-sample size of two objects, in which case tallying (with dichotomized cues) had the highest accuracy. Again, tallying was not tested with exact numerical values.

Brighton and Gigerenzer (2012) compared take-the-best to SVM in a single data set, where attributes were naturally binary, and found that the accuracy levels of the two models were comparable throughout the learning curve.

Şimşek and Buckmann (2015) presented learning curves for heuristics, logistic regression, and decision trees on 63 natural data sets. On average, tallying was the most accurate method on very small sample sizes. When models were trained on 50% of instances in the data set, mean accuracy across data sets was 0.725 for tallying, 0.743 for take-the-best, 0.746 for CART, and 0.747 for logistic regression.

Table 1: Data sets used in the analysis.

ID	Name	Objects	Cues	ID	Name	Objects	Cues
1	Manpower	17	5	29	Mortality	60	15
2	Waste	20	5	30	Movie	62	12
3	Jet	22	5	31	Dropout	63	18
4	Sperm	24	8	32	Land	67	4
5	Cigarette	25	3	33	Lakes	69	10
6	Galápagos	29	5	34	City	76	9
7	Agriculture	29	6	35	Car	93	21
8	Ice	30	3	36	Basketball	96	4
9	Oxidant	30	4	37	Infant	101	3
10	Recycling	31	7	38	Obesity	136	11
11	Reactor	32	10	39	Contraception	152	6
12	Rebellion	32	6	40	Votes	159	5
13	Excavator	33	5	41	Pitcher	176	15
14	Occupation	36	3	42	Birthweight	189	8
15	Pinot	38	6	43	Athletes	202	8
16	Highway	39	11	44	CPU	209	6
17	AFL	41	4	45	Tip	244	6
18	Air	41	6	46	Bodyfat	252	13
19	Bones	42	6	47	Hitter	263	19
20	Mussels	44	8	48	Diamond	308	4
21	Mines	44	4	49	Algae	340	11
22	Prefecture	45	5	50	Faculty	397	5
23	Crime	47	15	51	Mileage	398	7
24	Homeless	50	7	52	Monet	430	4
25	SAT	50	4	53	Affair	601	8
26	Fuel	51	5	54	Lung	654	4
27	Salary	52	5	55	Rent	2053	10
28	Sleep	58	7	56	Home	3281	4

4. Analysis

We used 56 data sets gathered from a wide variety of sources, including online data repositories, statistics and data-mining competitions, packages for R statistical software, textbooks, and research publications. The subjects were diverse, including biology, business, computer science, ecology, economics, education, engineering, environmental science, medicine, political science, psychology, sociology, sports, and transportation. The data sets varied in size, ranging from 17 to 3,281 objects. They also varied in the amount of information available, ranging from 3 to 21 attributes. Many of the smaller data sets contained the entirety of the population of objects, for example, all 29 islands in the Galápagos archipelago. The data sets are listed in Table 1 and described in the supplementary material. All data sets are publicly available.

Missing attribute values were imputed by the mean, median, and mode value in the data set for interval, ordinal, and categorical attributes, respectively. Objects with missing criterion values were discarded. For ordinal attributes, attribute difference between two objects was recoded into a new ordinal attribute with three values, indicating if the differ-

ence is positive, negative, or zero. This is because the exact value of the difference is not meaningful for ordinal attributes, only its sign. A categorical attribute with c categories was recoded into c binary attributes, each indicating membership in one category.

Almost all earlier studies have dichotomized the numerical attributes as described in Section 3. We did not. All models, including heuristics, generally yield higher accuracy when attributes are not dichotomized.

We examine two performance metrics on decision quality. Our primary metric is accuracy, where a decision is considered to be accurate if it selects the object with the higher criterion value or if the objects are equal in criterion value. We examine, in addition, a linear loss metric that equals 0 if the decision is accurate, and $\frac{|y_A - y_B|}{z}$ otherwise, where z is a normalizing constant for the data set, computed as follows:

$$z = \frac{\sum_{\forall \langle A,B \rangle} |y_A - y_B|}{\sum_{\forall \langle A,B \rangle} 1}. \tag{1}$$

We present results with the following classification algorithms: random forests, SVMs, and naive Bayes; the following regression algorithms: multiple linear regression (MLR) with elastic net penalty and random forest regression; and the following heuristics: single-cue, take-the-best, and tallying. We trained SVMs using their implementation in the R package e1071 (Meyer et al., 2014). We tried both a linear and a Gaussian kernel, with a 10-fold cross-validated grid search for parameter values. We trained random forests and random forest regression using the implementation in the R package randomForest (Liaw and Wiener, 2002), with 10-fold cross-validated search for the best value of the parameter *mtry*. We trained linear regression with elastic net regularization (Zou and Hastie, 2005), using the R package glmnet (Friedman et al., 2009). We selected the parameter values of α and λ using 10-fold cross-validation. We used the naive Bayes implementation in R package e1071 (Meyer et al., 2014), with Laplace smoothing. Additional implementation details are described in the supplementary material.

5. Results

We present results on the performance of each algorithm as training-set size increases, starting from a size of one. Recall that a training instance requires information on two objects. Consequently, a single training instance uses two objects from the data set. These two objects are then discarded, not to be used again in any train or test instance.

To generate learning curves, we randomly sampled m test instances from each data set ($m = n/10$, where n is the number of objects in the data set). We then progressively sampled training sets of increasing size using the remaining objects in the data set. We replicated this procedure 4,000 times for SVM, and 10,000 times for all other algorithms. The smaller numbers for SVM are due to the substantially higher training time required for this model.

Figure 1 shows the mean learning curve across 56 data sets for various algorithms. Table 2 reports mean accuracy and linear loss across 56 data sets for training sets of size 10, 20, 50, and 90 instances. Figure 2 displays individual learning curves on 20 of the data sets.

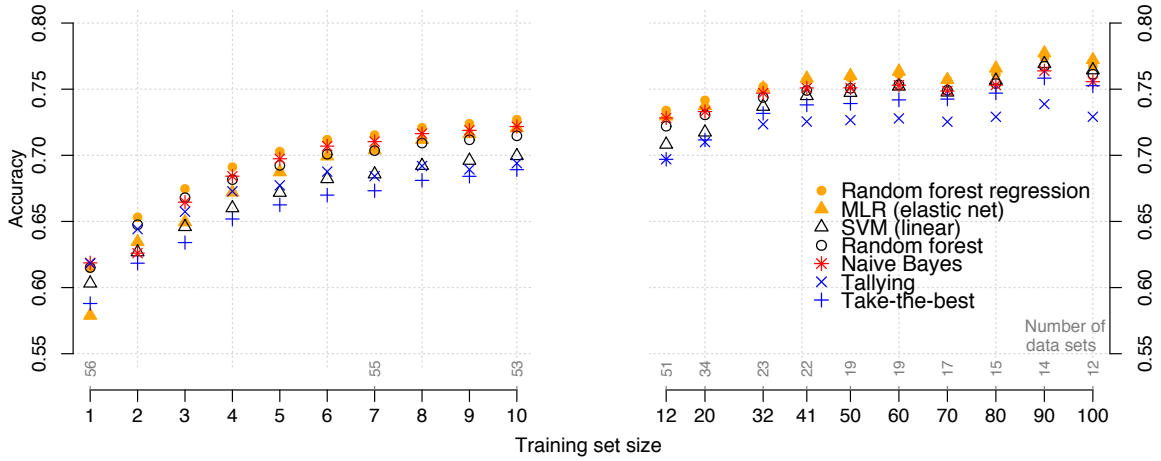

Figure 1: Mean accuracy in 56 data sets as a function of the training-set size. On the horizontal axis, the figure also shows the number of data sets contributing to the mean performance.

Table 2: Mean performance across 56 data sets

	Accuracy				1 − Linear loss			
Training set size	10	20	50	90	10	20	50	90
Data sets	$n = 53$	$n = 34$	$n = 19$	$n = 14$	$n = 53$	$n = 34$	$n = 19$	$n = 14$
Take-the-best	0.689	0.712	0.739	0.758	0.741	0.772	0.797	0.819
Tallying	0.694	0.710	0.727	0.739	0.750	0.772	0.784	0.798
Single-cue	0.673	0.695	0.723	0.746	0.721	0.749	0.775	0.804
Naive Bayes	0.722	0.733	0.751	0.764	0.792	0.807	0.817	0.832
Random forest	0.715	0.731	0.751	0.767	0.781	0.798	0.811	0.829
SVM (linear)	0.700	0.717	0.747	0.769	0.765	0.784	0.811	0.835
SVM (radial)	0.687	0.710	0.741	0.762	0.745	0.776	0.801	0.824
Random forest regression	0.727	0.742	0.759	0.774	0.800	0.814	0.823	0.839
MLR (elastic net)	0.721	0.737	0.760	0.777	0.788	0.809	0.827	0.846

Along the mean learning curves, the differences between the heuristics and the statistical learning algorithms are relatively small. The maximum difference in accuracy between the best heuristic and the best algorithm at a given sample size is at most 0.037. On the early parts of the curve, tallying performed better than take-the-best but is roughly 0.025 percentage points behind the best performing algorithms: random forest, random forest regression, and naive Bayes. With larger training-set sizes, take-the-best performed remarkably well. With training-set sizes of 44 or larger, it trailed the top algorithm by 0.019 on average. We should also note that a relatively simple method, naive Bayes, performed remarkably well. Along the mean learning curve, it closely trailed random forest regression on most training-set sizes.

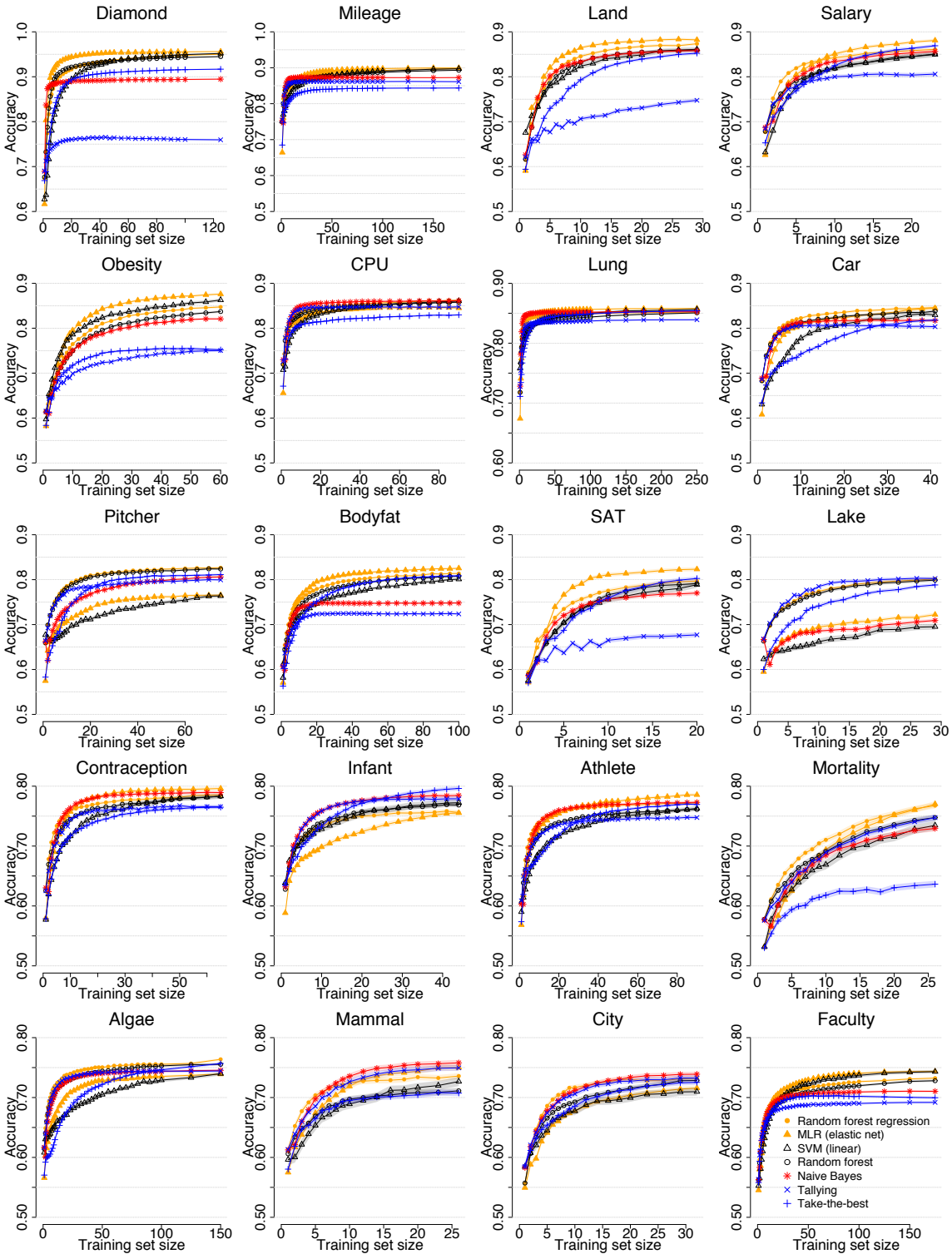

Figure 2: Learning curves on individual data sets.

On individual data sets, there were substantial differences in the performance of the various algorithms. In most of the data sets, at least one heuristic performed as well as or better than the best performing classification algorithm in all or parts of the learning curve. The results on linear loss are very similar to those on accuracy.

We measured the computation time required for training and testing the various models midway along the learning curve, where the size of the training set was roughly 90% of its maximum value. The experiment was run on a single kernel of a cluster (Intel Xeon CPU E5-2670, 4GB memory). For one pass over the data sets, SVM with the radial kernel required on average 283 minutes, SVM with the linear kernel required 16 minutes, random forest required two minutes, and MLR (elnet) required one minute. In contrast, take-the-best and tallying each required only 0.40 seconds. Note that the R packages we used to train SVM and random forests both call highly efficient Fortran and C code, while our implementation of the heuristics is programmed entirely in the much slower R language.

6. Discussion

Among the decision methods tested earlier in the literature, including logistic regression and decision trees, tallying stood out as the best method for sample sizes with 1–10 instances (Şimşek and Buckmann, 2015). We found that tallying (on average) falls short of more powerful statistical algorithms—random forest, random forest regression, and naive Bayes—even when training sets are small.

One surprising result is that with larger training-set sizes, multiple linear regression (with elastic net penalty) performed better on average than any other algorithm. This result has important implications because decision heuristics are often treated as an approximation of a linear decision rule. Several properties of the decision environment are known to allow heuristics to approximate a linear algorithm (Hogarth and Karelaia, 2006; Baucells et al., 2008; Martignon and Hoffrage, 2002; Katsikopoulos, 2011). Furthermore, these properties are prevalent in natural decision problems (Şimşek, 2013; Şimşek et al., 2016).

It is fair to conclude that in a diverse collection of natural environments, heuristics fared remarkably well when compared to powerful statistical learning algorithms. To put this result into context, it is useful to remember that the computational, informational, and memory requirements of heuristics, both at training and decision time, are extremely low.

One possible reason for the success of heuristics is that comparison is an easy problem, at least when compared to regression or classification. Given the fundamental importance of comparison for intelligent behavior, it would be fruitful to examine this problem theoretically and to develop statistical learning algorithms that address it directly, taking advantage of its special properties.

We hope these results will encourage further study of decision heuristics. In particular, we hope they will motivate further mathematical analysis as well as development of additional heuristic models.

References

Manel Baucells, Juan A. Carrasco, and Robin M. Hogarth. Cumulative dominance and heuristic performance in binary multiattribute choice. *Operations research*, 56(5):1289–1304, 2008.

Henry Brighton. Robust Inference with Simple Cognitive Models. In C. Lebiere and R. Wray, editors, *AAAI Spring Symposium: Cognitive Science Principles Meet AI Hard Problems*, pages 17–22. American Association for Artificial Intelligence, 2006.

Henry Brighton and Gerd Gigerenzer. Bayesian brains and cognitive mechanisms: Harmony or dissonance? In Nick Chater and Mike Oaksford, editors, *The probabilistic mind: Prospects for Bayesian cognitive science*, pages 189–208. Oxford University Press, New York, 2008.

Henry Brighton and Gerd Gigerenzer. Are rational actor models "rational"outside small worlds. *Evolution and Rationality: Decisions, Co-operation, and Strategic Behavior*, pages 84–109, 2012.

Jean Czerlinski, Gerd Gigerenzer, and Daniel G. Goldstein. How good are simple heuristics? In Gerd Gigerenzer, Peter M. Todd, and the ABC Research Group, editors, *Simple heuristics that make us smart*, pages 97–118. Oxford University Press, New York, 1999.

Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15(1):3133–3181, 2014.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. *glmnet: Lasso and elastic-net regularized generalized linear models*, 2009. URL `http://CRAN.R-project.org/package=glmnet`.

Gerd Gigerenzer and Daniel G. Goldstein. Reasoning the fast and frugal way: Models of bounded rationality. *Psychological Review*, 103(4):650–669, 1996.

Gerd Gigerenzer, Peter M. Todd, ABC Research Group, et al. *Simple heuristics that make us smart*. Oxford University Press, New York, 1999.

Gerd Gigerenzer, Ralph Hertwig, and Thosten Pachur, editors. *Heuristics: The Foundations of Adaptive Behavior*. Oxford University Press, New York, 2011. ISBN 9780199744282.

Robin M. Hogarth and Natalia Karelaia. Ignoring information in binary choice with continuous variables: When is less "more"? *Journal of Mathematical Psychology*, 49(2):115–124, 2005.

Robin M. Hogarth and Natalia Karelaia. "take-the-best" and other simple strategies: Why and when they work "well" with binary cues. *Theory and Decision*, 61(3):205–249, 2006.

John Hutchinson and Gerd Gigerenzer. Simple heuristics and rules of thumb: where psychologists and behavioural biologists might meet. *Behavioural Processes*, 69(2):97–124, 2005.

Konstantinos V. Katsikopoulos. Psychological heuristics for making inferences: Definition, performance, and the emerging theory and practice. *Decision Analysis*, 8(1):10–29, 2011.

Konstantinos V. Katsikopoulos, Lael J. Schooler, and Ralph Hertwig. The robust beauty of ordinary information. *Psychological Review*, 117(4):1259–1266, 2010.

Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002. URL `http://CRAN.R-project.org/doc/Rnews/`.

Laura Martignon and Ulrich Hoffrage. Fast, frugal, and fit: Simple heuristics for paired comparison. *Theory and Decision*, 52(1):29–71, 2002.

Laura Martignon and Kathryn B. Laskey. Bayesian benchmarks for fast and frugal heuristics. In Gerd Gigerenzer, Peter M. Todd, and the ABC Research Group, editors, *Simple heuristics that make us smart*, pages 169–188. Oxford University Press, New York, 1999.

David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, and Friedrich Leisch. *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*, 2014. URL `http://CRAN.R-project.org/package=e1071`. R package version 1.6-4.

Özgür Şimşek. Linear decision rule as aspiration for simple decision heuristics. In Chris J. C. Burges, Leon Bottou, Max Welling, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2904–2912. Curran Associates, Inc., Red Hook, New York, 2013.

Özgür Şimşek and Marcus Buckmann. Learning from small samples: An analysis of simple decision heuristics. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3159–3167. Curran Associates, Inc., Red Hook, New York, 2015.

Özgür Şimşek, Simón Algorta, and Amit Kothiyal. Why most decisions are easy in tetris—and perhaps in other sequential decision problems, as well. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1757–1765, 2016.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.