Fast and Frugal Use of Cue Direction in States of Limited Knowledge

Magnus Persson (Magnus.Persson@psyk.uu.se)
Department of Psychology, Uppsala University
Box 1225, SE-751 42, Uppsala, Sweden

Peter Juslin (Peter.Juslin@psy.umu.se)
Department of Psychology, Umeå University
SE-901 87, Umeå, Sweden

Abstract

An exemplar-based algorithm, PROBEX, (Juslin & Persson, 1999) is shown to make robust decisions in multiple-cue inference tasks when very few exemplars are known. We demonstrate the crucial role of knowledge of cue directions for performance and confront PROBEX with an artificial environment specifically construed to favor a non-compensatory algorithm like Take The Best (Gigerenzer & Goldstein, 1996). PROBEX is demonstrated to perform well even in these unfavorable conditions. The explanation for the robust performance is that PROBEX approximates Dawes Rule by using information of the cue directions for all cues, while yet making few a priori assumptions about the structure of the environment.

Introduction

In this paper we will discuss the importance of knowledge of cue directions for probabilistic inference. The cue direction refers to the valence of the relationship between two variables, for example, as represented by the sign of a correlation, or a coefficient in a linear equation (e.g., a beta weight in linear regression). We will concentrate on a simple binary choice task format of the following sort, “Which German city has the higher population: a) Bonn, b) Hamburg?”. In this case, knowledge of a cue direction corresponds to knowing if a binary probability cue, say, that Hamburg, but not Bonn, has a soccer team in the Bundesliga, increases or decreases the probability that alternative (a) (Bonn) is the correct answer to the question.

The constraints on a plausible cognitive mechanism for learning cue-directions are complex and multifaceted. In some tasks, learning cue-directions is not important because they are known a priori. In those cases, it is the relative weight of the cues that is important. On the other hand, it is sometimes proposed that knowledge of cue directions is *the* crucial aspect of learning to make probabilistic inferences (Dawes & Corrigan, 1974). Moreover, inferences have to be made for new and unexpected tasks for which little previous experience is available. This means that the system cannot rely extensively on pre-computed knowledge—which cues are predictive, and the direction of predictive validity, has to be detected on the spot. Finally, a flexible algorithm should map both linear and nonlinear aspects of an environment.

These constraints boil down to whether we can find an algorithm integrating the ability to represent non-linearity with on-the-spot detection of cue directionality. Is this possible without violating psychological plausibility? In this paper, we show that PROBEX (PROBabilities from EXemplars: Juslin & Persson, 1999) is such an algorithm. It relies on similarity-guided retrieval of exemplars to capture nonlinear relationships and to estimate cue directions. An added bonus is that because PROBEX belongs to the class of *lazy algorithms* (Aha, 1997), which do not require pre-computed knowledge, this is achieved in a fast and frugal fashion.

An important development in the judgment literature is the concern with evaluating cognitive algorithms within real environments (Gigerenzer & Goldstein, 1996; Gigerenzer, Todd, & the ABC-group, 1999). Gigerenzer and Goldstein (1996) demonstrate that, when applied to the structure of a real environment, simple heuristics that only rely on a single cue perform on par with complex algorithms that integrate multiple cues. Take-The-Best (TTB) relying on the single most valid cue that is applicable, performed as well as linear multiple regression that integrates 9 cues. It was concluded that although TTB falls short of classical norms of rationality, it provides the same accuracy at a minimum of computation: It is ”fast and frugal”. One shortcoming of the simulations in Gigerenzer and Goldstein was that all algorithms were provided with a priori knowledge of cue directions.

Connectionist, exemplar-based and decision-tree architectures have been shown to compete evenly with TTB in regard to accuracy (Chater et al, 1999). Specifically, in Juslin and Persson (1999) it was shown that PROBEX outperformed TTB and linear multiple regression in regard to accuracy while relying on no pre-computed knowledge. PROBEX further provided a good quantitative fit to the quantitative point-estimates, binary decisions, and probability judgments made by human participants (see Dougherty, Gettys, & Ogden, 1999, for a similar approach).

In this paper, we complement Juslin and Persson (1999) in three respects: First, we illustrate the crucial role of estimating cue directions for the performance of any algorithm. Second, we explore boundary conditions for the robust performance of PROBEX by exposing it to an environment deliberately construed to favor a non-compensatory algorithm like TTB and a linear, additive algorithm like linear multiple regression. Finally we present a simple demonstration to elucidate why evolution should favor decisions algorithms that are robust in states of limited knowledge.

PROBEX—The Algorithm

Many theories in cognitive science stress the storage of exemplars (traces, instances) (e.g.; Kruschke, 1992; Logan, 1988; Medin & Schaffer, 1978; Nosofsky, 1984, Nosofsky & Palmeri, 1997). One property of exemplar-based models is that they describe algorithms that respond to both frequency and similarity. In this respect, they map onto well-known properties of human probability judgment (Juslin & Persson, 1999). PROBEX was developed from one of the well-known and successful exemplar-based model, the *context model* (Medin & Schaffer, 1978; Nosofsky, 1984). PROBEX amends the context model in the following respects (see Juslin & Persson, 1999): (a) With a sequential sampling mechanism that allows prediction of response times (as such PROBEX provides a humble cousin of the EBRW model presented by Nosofsky & Palmeri, 1997); (b) A dampening in order to predict pre-asymptotic performance (see also Nosofsky et al., 1992); and (c) response rules that allows prediction of point-estimates and subjective probability judgments. The simple stopping rule and the mechanisms for point-estimation and probability judgment are the main differences from previous exemplar-based models.

Knowledge of the environment is modeled by an $R \times C$ matrix, with *R* exemplars, *C* cue dimensions and one vector with *R* target values. The exemplars in the knowledge matrix represent distinct psychological entities, either traces of dated events from episodic memory or semantic knowledge. Exemplars are described in terms of binary feature values, except for the continuous target dimension t. Each exemplar is represented by *D* binary features $\bar{x}_i = \lfloor x_{i1}, x_{i2}, \ldots x_{iD} \rfloor$, where *1* denotes presence of the feature and *0* its absence. The participant is presented with a new exemplar $\bar{t}$ and is required to make a judgment or a decision. The similarity between $\bar{t}$ and stored exemplar y is computed by the multiplicative similarity rule of the context model,

$$S(\bar{t}, y) = \prod_{j=1}^{D} d_j \text{, } d_j = \begin{cases} 1 & if\ t_{.j} = y_{.j} \\ s & if\ t_{.j} \neq y_{.j} \end{cases} , \qquad (1)$$

where d_j is 1 if the values on a feature match and *s* if they mismatch. Similarity *s* is a parameter in the interval [0, 1] for the impact of mismatching features. For low values of *s*, the similarity is close to one only for an exemplar that is almost identical to the new exemplar, but for high values of *s* all of the stored exemplars are deemed very similar.

We examine PROBEX with *s=0.5* for all cue dimensions which is a compromise between these extremes. The idea is that this compromise, referred to as *similarity-graded probability*, is a particularly robust and efficient way to exploit states of limited knowledge (Juslin & Persson, 1999).

The stored exemplars race to determine the response (Logan, 1988; Nosofsky & Palmeri, 1997). The stored exemplars are retrieved one-by-one from an initial set K to yield a sequence $x_1, x_2, \ldots x_N$. The probability that exemplar $\bar{y}$ is the sampled exemplar x_n at iteration n is:

$$\forall y\left(y \in K_n\right) \text{, } P_n\left(\bar{x}_n = \bar{y}\right) = \frac{S\left(\bar{t}, y\right)}{\sum_{z \in K_n} S\left(\bar{t}, \bar{z}\right)} \qquad (2)$$

The summation in the denominator is performed across exemplars not yet sampled. A response is generated at iteration N, where the decision rule specified below terminates the sampling process. *N* is a random variable, the distribution of which can be used to predict response times.

To estimate the target value $v'(\bar{t})$ of the new exemplar $\bar{t}$, the target values $v(\bar{x}_i)$ of the retrieved exemplars $\bar{x}_i$ are considered. The estimate of the target value at iteration *n* is,

$$v'(\bar{t}, n) = \frac{\sum_{i=1}^{n} S(\bar{t}, \bar{x}_i) v\left(\bar{x}_i\right)}{\sum_{i=1}^{n} S(\bar{t}, \bar{x}_i)} , \qquad (3)$$

a weighted average of the retrieved target values, where the similarities are weights. The final estimate is $v'(\bar{t}) = v'(\bar{t}, N)$ where *N* is the first iteration where the conditions for the stopping rule are satisfied. Eq. 3 can also be produce probability assessments (Juslin & Persson, 1999).

The sampling of exemplars is terminated at the first iteration *N* where the following condition has been satisfied. The stopping rule is:

$$\left|v'(\bar{t}, n) - v'(\bar{t}, n-1)\right| < k \cdot \left|v'(\bar{t}, n)\right| . \qquad (4)$$

The free parameter *k* decides the sensitivity of the stopping rule. One can interpret this rule as a way of judging when the change in the point estimate from $v'(\bar{t}, n-1)$ to $v'(\bar{t}, n)$ is too small to merit further sampling.

Is PROBEX Frugal?

Gigerenzer and Goldstein (1996) proposed the notion of 'fast and frugal' as a conceptual threshold that a model of human decision making must reach in order to be plausible. At first sight, the mathematics of PROBEX that model retrieval of exemplars seems too complex to be "fast and frugal". But there are two major issues that resolve this dilemma. First, the complex part of PROBEX models quick and effortless memory processes that operate in parallel. Secondly, it is not enough to prove that an algorithm is efficient at the moment the decision is made, without concern for the requirements on pre-computation. PROBEX belongs to the class of *lazy algorithms* (Aha, 1997) in artificial intelligence, that avoid the processing of data before the task is given. On the other hand, all algorithms discussed in this paper, except PROBEX, rely on pre-computed representations. TTB (Gigerenzer & Goldstein, 1996) for example need to compute a sorted list of cue validities for the specific task, which in it self is computationally demanding. Arguably, if PROBEX makes good decisions by retrieving few exemplars and without using pre-computed representation, it is fast and frugal in a more general and important sense.

The Ecological Rationality of Five Algorithms

The German City-population Task The task in the initial study on ecological rationality was the German city-population task (Gigerenzer & Goldstein, 1996; see Gigerenzer, et al., 1999, for applications to other environments). The task is to answer questions such as ”Which city has the larger population: Heidelberg or Erlangen?” The decision process is modeled by an algorithm that relies on some strategy to make intelligent guesses about the populations of German cities. The simulation also requires an environment-model containing facts about German cities which—once known to the algorithm—can be used to infer city-populations. The environment is represented by nine binary cues that characterize each city, for example, whether a city is a state capital or not, whether it has a university or not, where the nine cues vary in predictive validity.

The Algorithms PROBEX was compared to four algorithms for guessing which of two objects have the largest value on the target dimension: (1) A linear multiple regression model with cues as independent variables and population as dependent variable. In a pair-wise comparison task, the algorithm decides on the city with the higher estimate. The direction of the cues is the sign of the regression weights. Linear multiple regression is included because it is routinely claimed to provide robust and accurate predictions. However, it cannot handle situations with few observations unless one amends it with a method such as Ridge Regression to compensate for cue dimensions with no information[1].

(2) Dawes' Rule (Gigerenzer, et al., 1999): A heuristic version of the linear model that counts how many of the cues support each of the two cities and decides on the city implied by more cues. Cue direction is represented as 1 or -1 if there is positive or negative correlation in the training data respectively and zero if it was not computable. As detailed below, two versions of Dawes' Rule were implemented in order to investigate the importance of a priori knowledge of the cue directions. (3) TTB (Gigerenzer & Goldstein, 1996): In pair-comparisons, TTB decides on the task implied by the first most valid cue that differentiates between the pair.

(4) QUICKEST (Gigerenzer, et al., 1999) is an algorithm appropriate for skewed distributions like the German city-populations, where most cities have small populations. For each cue, the mean population for cities with negative cue-values is computed (negative cue values are those that go with small populations, e.g., not being a state capital). Cues are rank-ordered from the cue with lowest mean given a negative cue value to the cue with the highest mean given a negative cue value[2]. To estimate a population the algorithm starts by checking if a city has the cue value that is first in this rank-order, then the next, and so on until a match is encountered. Then the mean population for the cities which have this cue is the estimate. Given the skew of the city-population distribution with mostly small cities, this algorithm is frugal in the sense of minimizing the number of cues that have to be accessed (i.e., for most cities the algorithm will stop for the first negative cue values in the rank-order). In pair-comparisons, QUICKEST decides on the city with the larger estimate.

Gigerenzer and Goldstein (1996) tested the algorithms by feeding them with all the pair-wise comparisons between German cities with more than 100 000 inhabitants. One weakness of this procedure was that the knowledge of the algorithms was assumed to consist of all the cities. A better test (Gigerenzer, et al., 1999) is to split the set of German cities into a training set and a test set. The training set is used to train the algorithm and the test set is the pool from which the test questions are constructed. This cross-validation is a true test of the robustness and detects any over-fitting to the training set. Moreover, it highlights the issue of learning the cue directions.

Learning Cue Directions In the simulations, all algorithms except A Priori Dawes' Rule have no a priori knowledge of the cues, but have to learn them from the known exemplars. In some tasks it, may be possible to infer cue direction by reasoning, but this topic is not addressed in this paper.

To illustrate the importance of knowing the cue directions, two versions of Dawes' Rule were implemented. The first version is the A Priori Dawes' Rule and it assumes that cue directions are known a priori. Its purpose is to show the theoretical upper limit of Dawes' Rule with a priori knowledge of cue directions. The second version, Dawes' Rule, relies on observed training exemplars to estimate cue directions by calculating whether each cue is positively or negatively correlated with the target dimension among the training exemplars. The difference between the variants indicates the importance of a priori knowledge of cue directions.

It is important to make special solutions for several of the algorithms below. With Dawes' Rule, insufficient data can make the correlation between a cue and the target variable undefined and then this cue is never used in the test phase. TTB must also be treated with care, because it use a sorted list of cue validities. When there are few exemplars it is often the case that several cue validities get the same numerical value and then these have to be listed in random order within the list. If this is not done the computer implementation can lead to a biased cue order which may either increase or decrease the accuracy of the algorithm.

[1] Ridge regression has the drawback of biasing the predictions towards the mean, and thus lowers the predictive accuracy when the weights are calculated from many observations without problems with correlated variables. We hand-picked an intermediate ridge constant, 0.1, that increased accuracy with limited information (small training set) but did not lower performance with much information (large training set).

[2] We did not implement QUICKEST exactly as Gigerenzer et al. (1999) did as we did not use approximations to natural numbers which probably implies that our implementation gives slightly better predictions.

Study 1: Pair-Comparisons in the German City-Population Task

From an evolutionary perspective, it is important that an algorithm is good also when information is limited, because a decision maker has to survive as a "beginner" (see Study 3 below). Performance with small training sets is therefore a most important aspect of the robustness of an algorithm. Also, each algorithm soon reaches an asymptote that depends on both the cue structure and the algorithm itself. In the first simulation we thus compared the algorithms in the standard binary choice task with a particular eye to their performance in states of severely limited knowledge.

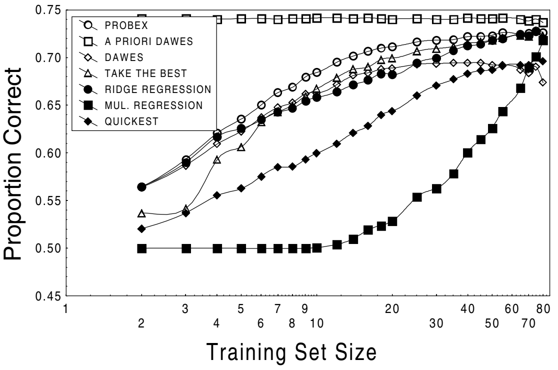

Figure 1. Accuracy and robustness of the algorithms. For each training set size, 1000 sets were randomized.

Method The dependent variable was proportion of correct inferences among the pair-comparisons of the test set. For each training set-size (2-80), as many as 1000 participants were simulated in order to make the errors of the means negligible. For each simulated participant, the German cities were randomly partitioned into training and test sets. Each algorithm was given the training set and the remaining cities were combined into all possible unique pairs and used as the test set. The data for the 83 German cities were collected from Gigerenzer and Goldstein (1996).

Results and Discussion Aside from A Priori Dawes' Rule that gets a head-start by virtue of its initial knowledge, PROBEX seems to be the winner in Figure 1. At minimum knowledge, PROBEX, Ridge Regression, and Dawes' Rule utilize the information equally well. TTB seems to have problems sorting the cues in a good order with little information, but does very well with 6 or more training exemplars. QUICKEST trade off accuracy for speed which is seen clearly in the cross validation paradigm. Note that A Priori Dawes' Rule does not define the asymptote that the other algorithms converge on, because it does not depend on the training set and does not suffer from over-fitting. This effect of cross-validation explains why it is constant at the high proportion correct of .74. It is surprising that Dawes' Rule performs worse with more information, but this is because the correlation between two cues in the full training set is negative. With less information there is a greater chance to get a training set with only positive correlations. If cue validities had been calculated instead of correlations, Dawes' Rule would have done better.

In sum: The algorithms that use all information, such as PROBEX, perform robustly in states of limited knowledge. PROBEX, however, also performs best when more information is available. The superior performance of A Priori Dawes' Rule shows that a crucial aspect to be acquired by any algorithm is knowledge of the cue directions.

PROBEX and Dawes Rule

In order to understand why PROBEX performs better than TTB with few training exemplars, it is instructive to consider only two optimal training exemplars: Big-city with all cues set to 1 and Small-city with all cues set to 0. For simplicity, all cues are assumed to be positively correlated with population. For PROBEX the similarity of the probe to Big-city decreases monotonically as a function of the number of cues in the probe that are not 1. The opposite holds for the similarity to Small-city which increases for each cue not set to 1. Because it is the number of cues set to 1 that differentiate the probes, PROBEX has the same high accuracy as Dawes' Rule.

TTB computes the cue direction for all cues but cannot apply this information, because the order of the cues is selected at random when the cue validity is the same for all cues. The data point with two training exemplars in Figure 1 suffers from this problem because the search order will be picked at random from those cues that have a well defined direction. PROBEX, Ridge Regression and Dawes' rule, on the other hand, integrate cue direction information from all cues in every decision in a similar way, which explains why they have identical proportions correct for the case of two training exemplars in Figure 1.

Study 2: A Non-Compensatory Data Set

When is PROBEX outperformed by other algorithms, like linear multiple regression and TTB? A good guess is in an environment with additive and non-compensatory cues. The cues are non-compensatory if the optimal regression weights are such that the largest weight is bigger than the sum of the smaller weights. The second largest cue should likewise be larger than the sum all the remaining cues, and so on (Gigerenzer et al., 1999). An environment with linear, additive relations favors the regression model and a non-compensatory cue-structure favors TTB. A simple environment for which this is true can be defined by ordinary binary numbers. Each binary number can be used as a cue. For example, 5 is written as 00101 in binary numbers, and gives the cues c_1= 0, c_2= 0, c_3=1, c_4=0 and c_5= 1. The optimal weights here are {16, 8, 4, 2, 1}, respectively (i.e., $16\cdot0+8\cdot0+4\cdot1+2\cdot0+1\cdot1=5$).

How much better than PROBEX will linear multiple regression and TTB perform in this environment specifically construed to fit the latter two algorithms?

Method The same procedure was used as in Study 1, except that the data were binary numbers between 0 and 32. Analogously with the German city-population task, the task was to guess which of two binary vectors has the highest number.

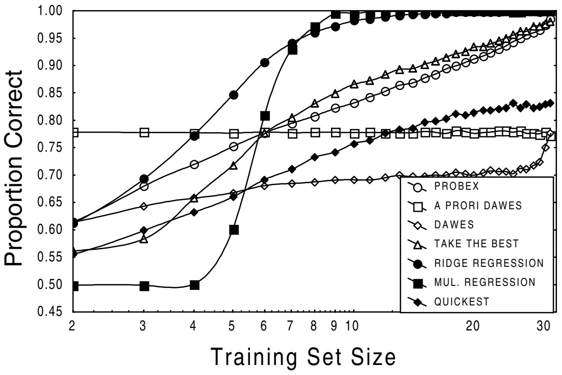

Figure 2. PROBEX in an artificial environment composed of the binary numbers.

Results and Discussion Ridge regression is clearly superior in Figure 2, but suffers slightly from biased estimates with 8-15 exemplars in the training set. TTB does not reach the asymptote unless it is trained with almost every exemplar, and is easily beaten by PROBEX when there are few training exemplars. The two variants of Dawes Rule cannot use the non-compensatory nature of the cues and are stuck at about the same level of performance as in Study 1.

As expected, multiple regression is perfectly suited for this environment, but both TTB and PROBEX converge on the same asymptote. More surprisingly, PROBEX is better than TTB with few exemplars. TTB only evens the score when more training exemplars are available, despite the fact that this cue structure is optimal for TTB. Thus, linear multiple regression converges more rapidly on the asymptote, but TTB enjoys no clear advantage over PROBEX.

Study 3: An Unforgiving Environment

Does it matter if a decision algorithm is a few percentages better for decisions made in states of limited knowledge? The answer to this question, of course, depends on the consequences of these decisions. In this final section, we provide an simple demonstration that in an environment where poor decisions are fatal, and experience is only gained conditional on the survival of previous decisions, a small difference in decision quality may add up quickly. Arguably, these are the living conditions of many animals, including those of humans for a large portion of the evolutionary history.

Method In order to cover a wider range of possibilities and simplify the demonstration, we modeled three ideal decisions strategies, *Early Learner* (EL), *Late Slow Learner* (LSL) and *Late Fast Learner* (LSF), as linear functions roughly similar to the functions in Figure 1 and 2. EL is PROBEX-like in the sense that it has a slight advantage early in the learning process. LSL and LSF are more TTB-like in that they start at a lower level but increases in accuracy, where LSF increases at a higher speed.

$$p_{EL}(correct) = 0.55 + 0.010 \cdot i \qquad (5)$$
$$p_{LSL}(correct) = 0.50 + 0.015 \cdot i \qquad (6)$$
$$p_{LFL}(correct) = 0.50 + 0.020 \cdot i \qquad (7)$$

Equation 5, 6 and 7 define the probability of a correct decision if the decision maker has *i* training exemplars as guidance. Two simulations were made where EL was pitted once against LSL and once against LFL. Ten generations were simulated, where the relative proportion of surviving decision makers decided the relative proportion in the next generation. Each generation made 11 decisions, where *i* was varied from 0 to 10. For example, a member of the EL-species had a 0.55 chance to survive the first decision and 0.65 chance to survive the last. LSL and LFL both had to start off at 0.5, but ended with 0.65 and 0.7, respectively.

LSL is an example of a decision strategy that starts off poorly and barely catches up with EL. LFL, on the other hand, is better than EL for the last 5 trials. Note that, if the decisions were not fatal, LFL and EL would both make the same overall amount of correct decisions.

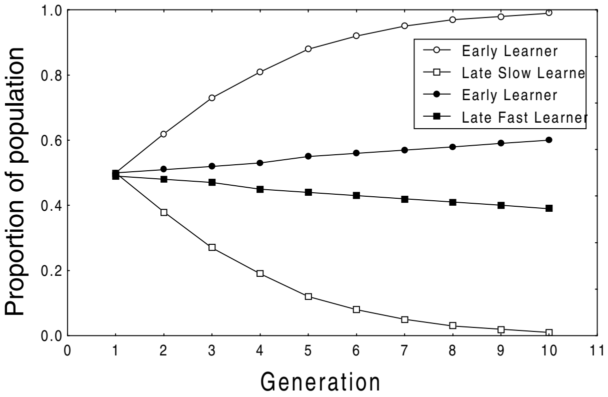

Figure 3: Two simulations of decision making in a unforgiving environment. Open symbols present Early Learners vs. Late Slow Learners, filled symbols present Early Learners vs. Late Fast Learners.

Results and Discussion Figure 3 presents the relative population proportions in a competition between EL and LSL, on the one hand, and EL and LFL, on the other. LSL would vanish very quickly in such a harsh environment. LFL are better but the losses in the beginning of every generation cannot be repaired despite the superior performance at the end of each generation (experienced decision makers).

This example is artificial and simplified, but shows that it is important to be a few percent better with little information than a few percent better with a lot of information, if learning is potentially dangerous. Indeed, the differences need not be large if they sum up over thousands of generations.

General Discussion

We propose that a plausible model of the cognitive processes that underlie memory-based judgment and decision making should have at least three properties: *First*, the model should be consistent with—and preferably extend on—previous models with independent support in the cognitive science literature. PROBEX is a moderately modified version of one of the most successful models from the categorization literature—the context model (Medin & Schaffer, 1978; Nosofsky, 1984). *Second*, algorithm-details that pertain to implementation in judgment and decision making needs to be tested. First steps along these lines have been taken by fitting the predictions by PROBEX to empirical judgment data (Juslin & Persson, 1999).

Third, as implied by the research on ecological rationality (Gigerenzer et al., 1999), a cognitive algorithm should make sense also from an evolutionary perspective. An algorithm favored by natural selection should produce accurate judgments when applied to the constraints of a real environment, require a minimum of mental effort, and be robust in states of limited knowledge. In this paper we have scrutinized the ability of PROBEX to infer and use cue direc tions, in comparison with a number of fast-and-frugal algorithms discussed by Gigerenzer et al. (1999).

PROBEX provides a flexible and efficient way to compute and use directions of many cues on the spot, that is, without requiring any pre-computed knowledge. Moreover, TTB enjoys no systematic advantage over PROBEX in an environment specifically designed to favor a non-compensatory strategy. Importantly, in contrast to the other algorithms, PROBEX brings no strong commitment in regard to the presumed structure of the environment (e.g., linear, compensatory), but applies equally well to nonlinear environments (such as the classic X-OR-problem). Arguably, this is the kind of flexibility favored by evolution in adaptation to a complex and uncertain environment.

Acknowledgments

The research reported in this paper was supported by the Swedish Council for Research in the Humanities and Social Sciences.

References

Aha, D. W. (1997). *Lazy Learning*. Kluwer, Norwell, MA.

Aha, D. W., & Goldstone, R. L. (1992). Concept learning and flexible weighting. *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society* (pp.534-539). Mahwah, New Jersey: Lawrence Erlbaum.

Chater, N., Oaksford, M., Nakisa, R., Redington, M. (1999). *Fast, frugal and rational: How rational norms explain behavior.* Submitted for publication.

Dawes, R. M., & Corrigan, B. (1974). Linear models in decision making, *Psychological Bulletin, 81,* 95-106.

Dougherty, M. R. P., Gettys, C, F., Ogden, E. E. (1999). MINERVA-DM: A memory processes model for judgments of likelihood. *Psychological Review, 106,* 180-209.

Gigerenzer, G. (1993). The bounded rationality of probabilistic mental models. In K. I. Manktelow, & D. E. Over (Eds.), *Rationality: Psychological and philosophical perspectives* (pp. 129-161). London: Routledge.

Gigerenzer, G., & Goldstein, D. G. (1996). Reasoning the fast and frugal way: Models of bounded rationality. *Psychological Review, 103*, 650-669.

Gigerenzer, G., Hoffrage, U., & Kleinbölting, H. (1991). Probabilistic mental models: A Brunswikian theory of confidence. *Psychological Review, 98*, 506-528.

Gigerenzer, G., Todd, P., & the ABC Research Group (1999). *Simple heuristics that make us smart*. New York: Oxford University Press.

Juslin, P., & Persson, M. (1999). *PROBabilities from EXemplars: On the role of similarity and frequency in probability judgment*. Submitted for publication.

Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review, 99*, 22-44.

Logan, D. G. (1988). Towards an instance theory of automatization. *Psychological Review, 95*, 492-527.

Medin, D. L., & Schaffer, M. M. (1978). Context model of classification learning. *Psychological Review, 85,* 207-238.

Nosofsky, R. M. (1984). Choice, similarity, and the context theory of classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 10,* 104-114.

Nosofsky, R. M., Kruschke, J., & McKinley, S. C. (1992). Combining exemplar-based category representations and connectionist learning rules. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18*, 211-233.

Nosofsky, R. M., & Palmeri, T. J. (1997). An exemplar-based random walk model of speeded classification. *Psychological Review, 104,* 266-300.