Probabilistic Formulation of the Take The Best Heuristic

Tomi Peltola (tomi.peltola@aalto.fi)
Helsinki Institute for Information Technology HIIT, Department of Computer Science
Aalto University, Espoo, Finland

Jussi Jokinen (jussi.jokinen@aalto.fi)
Department of Communications and Networking
Aalto University, Espoo, Finland

Samuel Kaski (samuel.kaski@aalto.fi)
Helsinki Institute for Information Technology HIIT, Department of Computer Science
Aalto University, Espoo, Finland

Abstract

The framework of cognitively bounded rationality treats problem solving as fundamentally rational, but emphasises that it is constrained by cognitive architecture and the task environment. This paper investigates a simple decision making heuristic, Take The Best (TTB), within that framework. We formulate TTB as a likelihood-based probabilistic model, where the decision strategy arises by probabilistic inference based on the training data and the model constraints. The strengths of the probabilistic formulation, in addition to providing a bounded rational account of the learning of the heuristic, include natural extensibility with additional cognitively plausible constraints and prior information, and the possibility to embed the heuristic as a subpart of a larger probabilistic model. We extend the model to learn cue discrimination thresholds for continuous-valued cues and experiment with using the model to account for biased preference feedback from a bounded rational agent in a simulated interactive machine learning task.

Keywords: Bayesian models; bounded rationality; heuristics; Take The Best

Introduction

Natural environments require agents to make decisions with incomplete information and in limited time. This, combined with the agent's limited information processing capacity, results in use of heuristic 'good enough', or 'satisficing', algorithms, that do not necessarily consider all possible alternatives (Simon, 1956). One such algorithm is *Take The Best* (TTB), which uses subjectively ranked informative cues to discriminate between alternatives (Gigerenzer & Goldstein, 1996). It searches through the ranked list of cues, until it finds a cue that discriminates between two choices. At this point, the search is stopped and decision made solely based on the last cue. This type of decision-making behaviour can be analysed under the concept of *bounded rationality* or *computational rationality*, where agents aim to maximise expected utility of their actions, given their architectural bounds as well as those of the task environment (Gershman et al., 2015; Howes et al., 2009; Simon, 1956). The power of this approach is that rational behaviour arises from the structure of the environment and the task.

TTB has been shown to approximate one of the core strategies in human-level decision-making (Bröder, 2000). Schulz et al. (2016) showed how the heuristic decision strategy arises from its components, using an approximate Bayesian computation inspired algorithm (ABC-TTB) to fit probability distributions on the parameters. We build on this idea and formalize a *Probabilistic Take The Best* model. Compared to ABC-TTB, the probabilistic TTB formulated here uses a proper likelihood-based model, with a noise model included in the model specification. With the proper likelihood model, we can use standard Bayesian computational tools for posterior inference, and the decision strategy arises from the posited model structure by conditioning on the training dataset. When used with non-informative (uniform) priors, the model has no tunable parameters. Informative prior distributions could be used to include prior knowledge.

The benefits of this approach are: (1) It lays out explicitly the assumptions in the TTB heuristic, and separates computation and model. (2) It provides a principled approach to learning the parameters of the model and their uncertainty, and suggests a possibility to extend the model without needing to change the computational framework. (3) It suggests a possibility of including prior information (or biases) in the prior distributions. (4) The TTB heuristic can be used as a component in a larger probabilistic model, such that the uncertainty of parameters in one component propagates in a natural way to other components.

Probabilistic extensions of TTB has also been considered by others. For example, Lee (2016) introduces a similar error model to ours, but focuses on model selection among multiple heuristics. Heck et al. (2017) considers a more elaborate error model, assuming higher rates of errors the more steps a decision takes. Neither, however, learns the cue search order and directions probabilistically. Our model could be naturally extended with more complex error models and used in model selection contexts.

In the next sections, we specify the model and give computational details. We then demonstrate the benefits of the formulation by proposing an extension for learning of cue discrimination thresholds, that is, the minimum differences in the cues for continuous or ordinal-valued cues required for discrimination. After comparing the performance to classic TTB, ABC-TTB, and logistic regression on benchmark datasets, we use the probabilistic TTB model as a component of a larger probabilistic model and, in particular, simulate a

function learning case. In the case study, a boundedly rational (using TTB) agent is assumed, giving us pairwise preferences on the function evaluated at pairs of points. We show how to accommodate this bias in the function learning.

Probabilistic Take The Best Model

Let $x_i \in \mathbb{R}^M$ and $x_j \in \mathbb{R}^M$ be feature vectors of items i and j. Let $\delta_{ij} = x_i - x_j \in \mathbb{R}^M$ be their difference. Take The Best (TTB) looks at the features, called cues, one by one in a specific order to select which item is preferred. When it finds the first cue that discriminates between the items (that is, $\delta_{ij}^{(m)} \neq 0$), it chooses one of the items as a winner depending on which direction is preferred for this cue. For example, assume the mth cue discriminates and $\delta_{ij}^{(m)} > 0$, and a large value is preferred for the mth cue. The winner is then item i and the output for the comparison is $y_{ij} = 1$, whereas if a smaller value would be preferred, then $y_{ij} = 0$.

The parameters of the TTB model are the cue order and directions. These are classically learned by looking at the correlations of the cues with the criterion quantity in a training set (Gigerenzer & Goldstein, 1996).

To formulate the probabilistic model, let g denote the cue order (takes a value of a permutation of the M cues) and $d \in \{-1,1\}^M$ the directions. For the probabilistic model to tolerate noise, we allow the outcome of the comparison to randomly flip (a flip noise likelihood), which brings in a third parameter, the flip probability $\varepsilon \in (0, \frac{1}{2})$. The probabilistic model with uniform priors on g and d and a beta distribution prior on ε (restricted to $(0, \frac{1}{2})$) with parameters $1,1$ (uniform) is

$$\begin{aligned} p(g) &= \frac{1}{M!}, \\ p(d_m) &= \frac{1}{2}, \qquad\qquad m = 1,\ldots,M, \\ p(\varepsilon) &= 2I(0 < \varepsilon < \frac{1}{2}), \end{aligned}$$

where the indicator function $I(C) = 1$ if the condition C holds and 0 otherwise.

Let $T_{g,d}(x_i,x_j)$ give the prediction of the non-probabilistic TTB given the cue order g and directions d. Given g, d, and ε, the observation model (implied by the flip noise assumption) for y_{ij} in the probabilistic model we introduce is

$$\begin{aligned} p(y_{ij} \mid x_i,x_j,\varepsilon,g,d) =& I(T_{g,d}(x_i,x_j) = 1)\,\mathrm{Ber}(y_{ij} \mid 1-\varepsilon) \\ &+ I(T_{g,d}(x_i,x_j) = 0)\,\mathrm{Ber}(y_{ij} \mid \varepsilon) \\ &+ I(T_{g,d}(x_i,x_j) = \varnothing)\,\mathrm{Ber}(y_{ij} \mid \frac{1}{2}). \end{aligned} \tag{1}$$

The Bernoulli (Ber) distributions model the flip noise. The last branch, $T_{g,d}(x_i,x_j) = \varnothing$ corresponds to the case of no discriminating cues between i and j, with the outcome chosen randomly with probability $\frac{1}{2}$.

The number of configurations of cue order and directions is $M! \times 2^M$. For a small number of cues, to compute the posterior distribution $p(g,d,\varepsilon \mid \mathcal{D})$ given a dataset $\mathcal{D}$, one can enumerate all the possibilities. For larger numbers, Markov chain Monte Carlo sampling can be used. Computational details are given in the next section.

Computational Details

Computational details of the posterior distribution, parameter inference, and predictive distribution are given in this section. We first discuss how to simplify the likelihood and integrate out the flip noise parameter ε analytically, allowing posterior computation to focus on the cue search order and directions. We give a Markov chain Monte Carlo algorithm for parameter inference for cases where exhaustive computation over the cue orders and directions is infeasible.

Marginalizing ε and Posterior Distribution

The likelihood given g, d, and ε is given by Equation 1. This simplifies to ε for the pairs $(y_{ij} = 0, T_{g,d}(x_i,x_j) = 1)$ and $(y_{ij} = 1, T_{g,d}(x_i,x_j) = 0)$, that is, when the observed value and predicted value are different (called "wrong predictions" below), and to $1-\varepsilon$ for $(y_{ij} = 1, T_{g,d}(x_i,x_j) = 1)$ and $(y_{ij} = 0, T_{g,d}(x_i,x_j) = 0)$, that is, when observed and predicted value are the same (called "correct predictions" below). When $T_{g,d}(x_i,x_j) = \varnothing$ (called "undecided predictions" below), the likelihood of either outcome is $\frac{1}{2}$.

We can then compute the likelihood for multiple observations of comparisons from a set of pairs $\mathcal{P}$, $Y = \{y_{ij}; (i,j) \in \mathcal{P}\}$, $X = \{(x_i,x_j); (i,j) \in \mathcal{P}\}$, and marginalize out ε:

$$\begin{aligned} p(Y \mid X,g,d) &= \int_0^{\frac{1}{2}} \prod_{(i,j)\in\mathcal{P}} p(y_{ij} \mid x_i,x_j,\varepsilon,g,d)\,p(\varepsilon)\,d\varepsilon \\ &= \frac{1}{Z_\varepsilon}\left(\frac{1}{2}\right)^{N_\varnothing} \int_0^{\frac{1}{2}} \varepsilon^{N_i+\alpha-1}(1-\varepsilon)^{N_c+\beta-1}\,d\varepsilon \\ &= \frac{1}{Z_\varepsilon}\left(\frac{1}{2}\right)^{N_\varnothing} B_{\frac{1}{2}}(N_i+\alpha, N_c+\beta), \end{aligned} \tag{2}$$

where $N_\varnothing$, N_i, and N_c are the numbers of undecided, incorrect, and correct predictions by the TTB model with the cue order g and directions d. Here, Z_ε is the normalizing constant of the beta prior with parameters α and β restricted to $(0, \frac{1}{2})$ (where $\alpha = 1$, $\beta = 1$ for the uniform prior), and $B_{\frac{1}{2}}$ is the incomplete beta function[1].

Let $\mathcal{D} = (Y,X)$ be the observed training set. The posterior probabilities $p(g,d \mid \mathcal{D})$ are proportional to Equation 2, with the sum $Z = \sum_{g,d} p(g,d \mid \mathcal{D})$ being the normalizing constant.

The conditional posterior distribution of ε given g,d is

$$p(\varepsilon \mid \mathcal{D},g,d) = \frac{\varepsilon^{N_i+\alpha-1}(1-\varepsilon)^{N_c+\beta-1}}{B_{\frac{1}{2}}(N_i+\alpha, N_c+\beta)}.$$

This is a beta distribution with parameters $N_i+\alpha$ and $N_c+\beta$ restricted to $(0, \frac{1}{2})$.

[1] "Unregularized" incomplete beta function, as defined by the integral in the second-to-last line of Equation 2.

Markov Chain Monte Carlo Sampling

A collapsed Gibbs sampling algorithm is used. We first integrate over the noise flip probability ε and only sample over the cue order g and directions d using Algorithm 1.

Algorithm 1: Markov chain Monte Carlo sampler.

Data: $\mathcal{D} = (Y, X)$ and g and d priors.
Result: S samples from $p(g, d \mid \mathcal{D})$.
Initialize g and d, e.g., randomly.;
for $s \leftarrow 1$ **to** S **do**
  for $m \leftarrow 1$ **to** M *in random order* **do**
    1. Form all cue orders such that the cue m takes all positions and the other cues remain in the same order, giving M different cue orders.;
    2. With the cue orders above, form all TTB models with those orders and with the cue m taking either direction and the directions of the other cues kept constant, giving $2M$ models (configurations of g and d).;
    3. Sample the next configuration from these with probabilities proportional to their unnormalized posterior probabilities, that is, $\propto p(g, d \mid \mathcal{D})$.;
  end
  Save the configuration of g and d as the sth sample.;
end

Predictive Distribution

Given a pair of new data points $\tilde{x}_1$ and $\tilde{x}_2$, the posterior predictive distribution of their comparison $\tilde{y}$ is

$$
\begin{aligned}
&p(\tilde{y} \mid \tilde{x}_1, \tilde{x}_2, \mathcal{D}) \\
&= \sum_{g,d} \int_0^{\frac{1}{2}} p(\tilde{y} \mid \tilde{x}_1, \tilde{x}_2, \varepsilon, g, d) p(\varepsilon, g, d \mid \mathcal{D}) d\varepsilon \\
&= \sum_{g,d} p(g, d \mid \mathcal{D}) \int_0^{\frac{1}{2}} p(\tilde{y} \mid \tilde{x}_1, \tilde{x}_2, \varepsilon, g, d) p(\varepsilon \mid g, d, \mathcal{D}) d\varepsilon \\
&\approx \frac{1}{S} \sum_{s=1}^{S} \int_0^{\frac{1}{2}} p(\tilde{y} \mid \tilde{x}_1, \tilde{x}_2, \varepsilon, g^{(s)}, d^{(s)}) p(\varepsilon \mid g^{(s)}, d^{(s)}, \mathcal{D}) d\varepsilon,
\end{aligned}
$$

where the last line applies in the case we have sampled S samples $(g^{(s)}, d^{(s)})$ from the posterior $p(g, d \mid \mathcal{D})$.

The integral over ε again gives incomplete beta functions

$$
\begin{aligned}
&\int_0^{\frac{1}{2}} p(\tilde{y} \mid \tilde{x}_1, \tilde{x}_2, \varepsilon, g, d) p(\varepsilon \mid g, d, \mathcal{D}) d\varepsilon \\
&= \left(\frac{1}{2}\right)^{I_\varnothing} \frac{B_{\frac{1}{2}}(N_i + \alpha + I_i, N_c + \beta + I_c)}{B_{\frac{1}{2}}(N_i + \alpha, N_c + \beta)},
\end{aligned}
$$

where $I_\varnothing = I(T_{g,d}(\tilde{x}_1, \tilde{x}_2) = \varnothing)$, $I_i = I(T_{g,d}(\tilde{x}_1, \tilde{x}_2) \neq \tilde{y})$, and $I_c = I(T_{g,d}(\tilde{x}_1, \tilde{x}_2) = \tilde{y})$.

Extensions

Handling Correlation in the Comparisons

In the above, we have assumed that the y_{ij} are independent (conditional on the model parameters). This, however, would often be expected to be violated. For example, we might often assume (some degree of) transitivity: if i is preferred to j and j to k, i could be assumed to be preferred to k.

In this work, we use a heuristic approach to account for transitivity dependencies by downweighting the likelihood terms of each y_{ij}. When we observe the full set of pairwise comparisons for N items, we assume that the ranking of the items would be enough to decide all pairwise comparisons. Since each pairwise comparison is 1 bit of information and there are $N!$ ways to rank N items (leading to $\log_2 N!$ bits of information), we assign a weight $\frac{\log_2 N!}{\binom{N}{2}}$ for each likelihood term (the terms are raised to this power). More formal ways of dealing with dependencies are left for future work.

Cue Discrimination Thresholds

For real-valued cues, practically all elements of $\delta_{ij} = x_i - x_j$ will be non-zero and the first cue will always discriminate. Yet, a small difference in a cue might be non-informative or imperceptible for humans (especially, if the cues are not given as precise numbers but, for example, visually), and large differences are more salient. We can extend the model to include non-negative threshold parameters $t_m \in [0, \infty)$ for each cue, such that only differences $|\delta_{ij}^{(m)}| > t_m$ are considered to discriminate between the items.

We have extended the MCMC sampling to allow sampling over a discrete set of pre-specified thresholds for each cue (assumed to have uniform prior). Extension to continuous-valued thresholds would also be possible.

Results

We first establish that the probabilistic Take The Best model is effective at learning from training data, and then demonstrate the model as a part of a larger probabilistic model.

Performance on Benchmark Datasets

The accuracy of the probabilistic TTB (PTTB) model is compared with the classic TTB, ABC-TTB, and logistic regression on four benchmark datasets. Heuristica R-package[2] is used for the classic TTB and logistic regression. ABC-TTB code is from `https://github.com/ericschulz/TTBABC`[3]. We used the same parameters for ABC-TTB as were used for the city dataset by Schulz et al. (2016) and made no effort to tune them for the other datasets. The other models have no tuning parameters. MCMC computation with 1000 samples after burn-in of 100 was used for PTTB models. The datasets homeless, profsalary, and city were obtained from `https://github.com/ericschulz/TTBABC`

[2] `https://cran.r-project.org/web/packages/heuristica/`, version 1.0.1.

[3] With minor bug fixes.

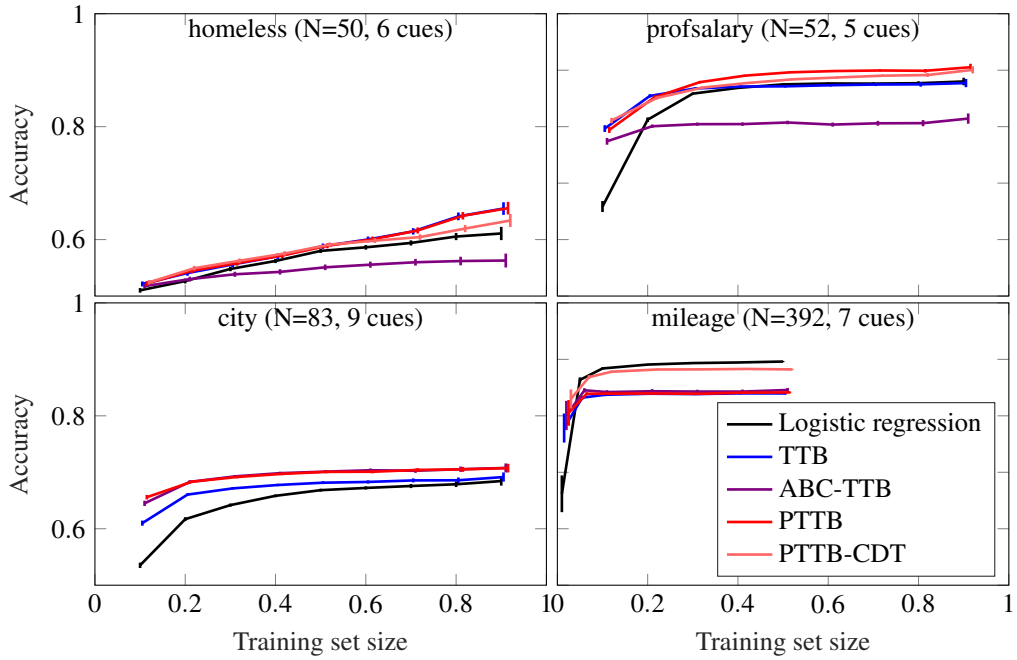

Figure 1: Accuracy on homeless, profsalary, city, and mileage datasets as a function of training dataset size (fraction of full data). Mileage data has a large number of samples and was run only up to 50% of full data (with fractions 0.01, 0.05 included). TTB=Take The Best; ABC-TTB=Approximately Bayesian Computed Take The Best (Schulz et al., 2016); PTTB=Probabilistic TTB; PTTB-CDT=PTTB with cue discrimination threshold learning.

and mileage dataset is from Matlab ("carbig.mat" with data points containing missing data values removed).

Figure 1 shows the discrimination accuracies as a function of training set size over 1000 replications of training and test set splits. The PTTB performs comparably or better compared to TTB and ABC-TTB methods. It also outperforms logistic regression except in the mileage dataset, where logistic regression is the best method expect in the smallest tested training set size case. The cue discrimination threshold learning (PTTB-CDT) decreases performance modestly in homeless and profsalary datasets, but increases performance considerably in the mileage dataset.

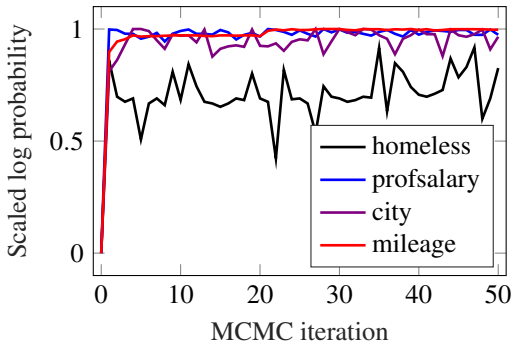

Figure 2: Scaled logarithm of the (unnormalized) posterior probability for 50 first iterations of MCMC in each dataset, starting from a random initial setting of parameters. For clear visualization in one figure, each curve is scaled (over full trace) such that its lowest value is 0 and highest is 1.

To examine the computational burden of finding good configurations of cue order and directions, Figure 2 shows the logarithm of the posterior probability of the first 50 iterations of a single chain of MCMC for each of the datasets, starting from a random configuration. Already a single iteration of the MCMC algorithm is enough to locate good models. Figure 3 shows the posterior uncertainty in the cue search order for PTTB, comparing it to the TTB cue validities. Although they have roughly similar trends, there are also clear differences.

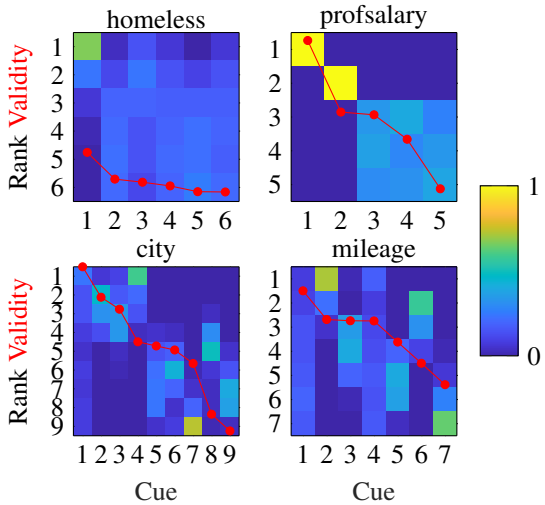

Figure 3: PTTB cue rank posterior probabilities (heatmap) and TTB cue validities (red line; y-axis runs from 0.5 to 1) learned from full datasets.

TTB as a Part of a Larger Probabilistic Model: Linear Regression with Pairwise Observations

We simulate a situation where we are interested in learning a function and there is an agent which can give us preferences over pairwise evaluations of the function, but the preferences are generated through a biased mechanism, perhaps due to the limited cognitive abilities of the agent. More specifically, we consider learning a linear regression function $f(x) = w^{\mathrm{T}}x$, with regression weights w, where we have a few direct (y_i, x_i) observations of the regression and a set of pairwise observations of whether $f(x_i)$ is larger or smaller than $f(x_j)$, generated using the TTB heuristic.

The main task is to learn the posterior distribution of w, $p(w \mid \mathcal{D}, \mathcal{H})$, where $\mathcal{D} = \{(y_i, x_i); i = 1, \ldots, N\}$ and $\mathcal{H} = \{h_{ij} = (f^*(x_i) > f^*(x_j)); (i,j) \in \mathcal{P}\}$, where $\mathcal{P}$ is a set of pairs, where pairwise preference observations are available, and the superscript * reminds us that the observations are biased. We use a Gaussian linear regression with a Gaussian prior on w:

$$p(y_i \mid x_i, w) = \mathrm{N}(y_i \mid w^{\mathrm{T}}x_i, \sigma^2), \qquad i = 1, \ldots, N,$$
$$p(w) = \mathrm{N}(w \mid 0, \tau^2),$$

where we fix $\sigma^2 = 1$ and $\tau^2 = 1$ and generate a simulated dataset $\mathcal{D}$ by sampling from the true model. The posterior distribution given $\mathcal{D}$ is $p(w \mid \mathcal{D}) \propto p(w) \prod_i p(y_i \mid x_i, w)$ (which is a multivariate Gaussian distribution).

To simulate an agent generating the pairwise observations, we first form a grid of points in the covariate space, denoting the set of points X_G. We then form all pairwise comparisons between these points given the true function f (or equivalently the true weights w) and use them as a training set to learn a TTB model. The observations $\mathcal{H}$ are then predictions from this TTB at a subset of the grid points. This simulates an agent who knows the true function, but can only access it through the heuristic TTB model. (We further included a threshold such that two covariate values must be further apart than it for them to discriminate in the TTB model.)

To use the pairwise observations to learn about w, we extend the model to include $\mathcal{H}$ through the probabilistic TTB model, similar to what was used to generate the observations:

$$\prod_{(i,j)\in\mathcal{P}} p(h_{ij} \mid x_i, x_j, \varepsilon, g, d) p(\varepsilon, g, d \mid X_G, w),$$

where the latter term is the posterior of a TTB model, where all pairwise comparisons generated by using w to predict the function values in the grid X_G are used as the training data. We further marginalize the TTB parameters to get the factor $p(\mathcal{H} \mid X_{\mathcal{H}}, X_G, w)$, where $X_{\mathcal{H}}$ denotes the set of pairs of points for which we have pairwise observations. That is, the posterior of w is now

$$\begin{aligned} p(w \mid \mathcal{D}, \mathcal{H}) &\propto p(w) \prod_i p(y_i \mid x_i, w) p(\mathcal{H} \mid X_{\mathcal{H}}, X_G, w) \\ &= p(w) \prod_i p(y_i \mid x_i, w) \times \\ &\sum_{g,d} \int_0^{\frac{1}{2}} \prod_{(i,j)\in\mathcal{P}} p(h_{ij} \mid x_i, x_j, \varepsilon, g, d) p(\varepsilon, g, d \mid X_G, w) d\varepsilon. \end{aligned}$$

In other words, the likelihood term $p(\mathcal{H} \mid X_{\mathcal{H}}, X_G, w)$ models how well the observed pairwise preferences $\mathcal{H}$ are concordant with a probabilistic TTB model induced by w on the points X_G. The regions with values of w with high concordance will get higher posterior mass. Exact computation of the probabilistic TTB model by enumeration is used in this experiment.

For comparison, we also formulate a model that includes the pairwise observations, but assumes them unbiased. For this we use the flip-noise likelihood

$$\begin{aligned} p(h_{ij} \mid x_i, x_j, w, \kappa) = {} & I((x_i - x_j)^{\mathrm{T}} w > 0) \, \mathrm{Ber}(h_{ij} \mid 1 - \kappa) \\ & + I((x_i - x_j)^{\mathrm{T}} w < 0) \, \mathrm{Ber}(h_{ij} \mid \kappa) \\ & + I((x_i - x_j)^{\mathrm{T}} w = 0) \, \mathrm{Ber}(h_{ij} \mid \frac{1}{2}) \end{aligned}$$

with uniform prior on the flip-noise parameter $\kappa \in (0, \frac{1}{2})$. The product of these terms over the observations in $\mathcal{H}$ is marginalized over κ to get the factor $p(\mathcal{H} \mid X_{\mathcal{H}}, w)$ for this alternative model.

Figure 4 shows the posterior densities for w computed over a grid of points for a case with 2 training data points in $\mathcal{D}$ and $\mathcal{H}$ containing all pairwise comparisons of a randomly selected set of 10 points x_i from a generated grid X_G, with true w being $(1, 0.8)$. The TTB-generated pairwise observations help to concentrate the posterior density. However, if the biased nature of the observations is not accounted for, the main bulk of posterior mass misses the true value. For comparison, the figure also shows the result if we would have unbiased pairwise observations.

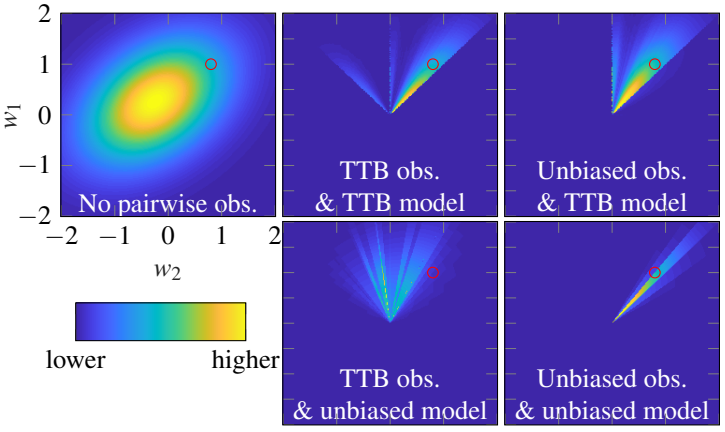

Figure 4: Posterior densities $p(w \mid \mathcal{D}, \mathcal{H})$ computed over a grid of (w_1, w_2) values and scaled such that the highest value in each is 1 (that is, the subfigures are not on the same scale). The true w is indicated by a red circle.

Discussion

We formulated a probabilistic, likelihood-based model of the Take The Best (TTB) heuristic. The decision strategy, that is, the cue search order, directions, and the decision uncertainty (or noise), arises from the posited model structure by

conditioning on the data and using probabilistic inference (Bayes theorem). The learning performance of the model compared favourably to classic TTB, ABC-TTB, and logistic regression. This indicates, together with fast convergence of the MCMC computation to good configurations of cue order and directions, that the probabilistic TTB provides a computationally frugal formulation of the Take The Best decision strategy (although this particular computational strategy is not uniquely positioned; any that implements Bayesian inference would suffice). We also presented an extension to learning cue discrimination thresholds for continuous-valued cues. This moves the TTB heuristic towards a compensatory model, similar to the evidence accumulation model of Lee & Cummins (2004) that interpolates between non-compensatory (single cue) and compensatory (multi-cue) decision making by learning a stopping rule (evidence threshold). Indeed, the extension considerably increased the decision accuracy in the mileage dataset, the only dataset where the compensatory logistic regression outperformed the TTB models.

Our model states that cue order results from rational adaptation, where the agent learns the optimal ordering of cues, given the task environment (cues and choices). This means that the principle of cue ordering is cognitively bounded, or computational, rationality. Another principle that has been suggested is fluency, where the memory retrieval time or visual saliency determines the cue order (Dimov & Link, 2017). The principle of cognitively bounded rationality does not exclude the latter possibility. In fact, its analysis of task behaviour that arises from the constraints of the environment *and* the cognitive architecture readily accepts such constraints as memory or vision. Above, we assumed perfect and immediate recall of cues, but it is entirely possible to extend the model to account for memory fluency. The resulting agent would – again, using rational adaptation – adjust the cue order based on how readily they are available for recall. Future work should add such constraints to the model, and investigate how the choice strategies change as their function.

We further demonstrated the benefit of formulating a probabilistic heuristic model by including it as a component in a function learning task to model preferential feedback from a biased agent. We believe that formulating joint probabilistic models of machine learning tasks and cognitive user models will be important, for example, in expert knowledge elicitation and interactive, human-in-the-loop machine learning (Daee et al., 2017, 2018). Notably, the probabilistic formalism allows naturally capturing our evolving uncertainty about the user's decision strategy (e.g., cue search order) in interaction and propagate it to other parts of the system. This highlights a role for cognitive science in the next generation of machine learning systems that interact and learn together with human experts and end-users. Cognitive user models can be used to increase the performance of the systems and allow for more natural and efficient human–computer interaction. Important future work is to evaluate the presented approach with human decision data and interaction.

Source code for the probabilistic TTB and the experiments is available at `https://github.com/to-mi/pttb`.

Acknowledgments

This work was financially supported by the Academy of Finland (Finnish Center of Excellence in Computational Inference Research COIN, and Grants 295503, 294238, 292334, 284642, and 310947).

References

Bröder, A. (2000). Assessing the empirical validity of the "take-the-best" heuristic as a model of human probabilistic inference. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *26*(5), 1332.

Daee, P., Peltola, T., Soare, M., & Kaski, S. (2017). Knowledge elicitation via sequential probabilistic inference for high-dimensional prediction. *Machine Learning*, *106*(9-10), 1599–1620.

Daee, P., Peltola, T., Vehtari, A., & Kaski, S. (2018). User modelling for avoiding overfitting in interactive knowledge elicitation for prediction. In *Proceedings of the 23rd international conference on Intelligent User Interfaces, IUI2018*.

Dimov, C. M., & Link, D. (2017). Do people order cues by retrieval fluency when making probabilistic inferences? *Journal of Behavioral Decision Making*.

Gershman, S. J., Horvitz, E. J., & Tenenbaum, J. B. (2015). Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, *349*(6245), 273–278.

Gigerenzer, G., & Goldstein, D. G. (1996). Reasoning the fast and frugal way: models of bounded rationality. *Psychological Review*, *103*(4), 650.

Heck, D. W., Hilbig, B. E., & Moshagen, M. (2017). From information processing to decisions: Formalizing and comparing psychologically plausible choice models. *Cognitive psychology*, *96*, 26–40.

Howes, A., Lewis, R. L., & Vera, A. (2009). Rational adaptation under task and processing constraints: implications for testing theories of cognition and action. *Psychological Review*, *116*(4), 717.

Lee, M. D. (2016). Bayesian outcome-based strategy classification. *Behavior Research Methods*, *48*(1), 29–41.

Lee, M. D., & Cummins, T. D. (2004). Evidence accumulation in decision making: Unifying the "take the best" and the "rational" models. *Psychonomic Bulletin & Review*, *11*(2), 343–352.

Schulz, E., Speekenbrink, M., & Meder, B. (2016). Simple trees in complex forests: Growing Take The Best by approximate Bayesian computation. *arXiv preprint arXiv:1605.01598*.

Simon, H. A. (1956). Rational choice and the structure of the environment. *Psychological Review*, *63*(2), 129.