18 Heuristic decision-making

18.1 Take-the-best

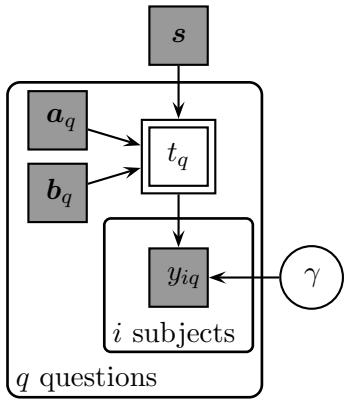
The take-the-best (TTB) model of decision-making (Gigerenzer & Goldstein, 1996) is a simple but influential account of how people choose between two stimuli on some criterion, and a good example of the general class of heuristic decision-making models (e.g., Gigerenzer & Todd, 1999; Gigerenzer & Gaissmaier, 2011; Payne, Bettman, & Johnson, 1990). TTB addresses decision tasks like "which of Frankfurt or Munich has the larger population?", "which of a catfish and a herring is more fertile?", and "which of these two professors has the higher salary?".

TTB assumes that all stimuli are represented in terms of the presence or absence of a common set of cues. In the well-studied German cities data set, this means cities are represented in terms of nine cues, including whether or not they have an international airport, whether they have hosted the Olympics, and whether they have a football team in the Bundesliga. Associated with each cue in TTB is a "cue validity." This validity measures the proportion of times that, for those pairs of stimuli where one has the cue and the other does not, the cue belongs to the stimulus that has the greater criterion value. For example, the cue "Is the city the national capital?" is highly valid because the capital city, Berlin, is also the most populous city.

The TTB model assumes that, when people decide which is the larger of two cities, they search the cues from highest to lowest validity, stopping as soon as a cue is found that one city has but the other does not. At this point, TTB says simply that people choose the city that has the cue. If all of the cues are exhausted, TTB assumes people guess.

The data we consider involve 20 subjects choosing between pairs of stimuli for the same 30 questions.[1] On each trial, a subject could search cues to find out whether or not the two stimuli had that cue. At any point during this search, the subject could choose one of the stimuli, on the basis of the available information. Thus, the experiment measures how people search for information, when they decide to stop that search, and what decision they make. We focus solely on the decision data collected during the experiment, with the goal of modeling the decisions made, and inferring from those decisions something about how people searched, and how they terminated their search.

[1] We thank Ben Newell for sharing these data.



$$t_q \leftarrow \text{TTB}_{\boldsymbol{s}}(\boldsymbol{a}_q, \boldsymbol{b}_q)$$
$$\gamma \sim \text{Uniform}(0.5, 1)$$
$$y_{iq} \sim \begin{cases} \text{Bernoulli}(\gamma) & \text{if } t_q = a \\ \text{Bernoulli}(1-\gamma) & \text{if } t_q = b \\ \text{Bernoulli}(0.5) & \text{otherwise} \end{cases}$$

Fig. 18.1 Graphical model for the take-the-best model of heuristic decision-making.

A graphical model for implementing TTB is shown in Figure 18.1. The decision made by the ith subject on the qth question is $y_{iq} = 1$ if the first stimulus (stimulus 'a') is chosen, and $y_{ij} = 0$ if the second stimulus (stimulus 'b') is chosen. The cues for the stimuli in the jth question are given by the vectors $\boldsymbol{a}_q$ and $\boldsymbol{b}_q$, and the validity-based search order is given by $\boldsymbol{s}$.

To make the inherently deterministic TTB model a probabilistic model, a "responding rate" or "accuracy of execution" parameter γ is included, so that TTB decisions are followed with probability γ (Rieskamp, 2008). Thus, the TTB decision for the qth question is given by t_q, taking binary values corresponding to the choices "a" or "b." We can write this $t_q \leftarrow \text{TTB}_{\boldsymbol{s}}(\boldsymbol{a}_q, \boldsymbol{b}_q)$. The observed human decisions are distributed as

$$y_{iq} \sim \begin{cases} \text{Bernoulli}(\gamma) & \text{if } t_q = a \\ \text{Bernoulli}(1-\gamma) & \text{if } t_q = b \\ \text{Bernoulli}(0.5) & \text{otherwise.} \end{cases}$$

The probability TTB is followed is expected to be high, and certainly should not be below 0.5, which would reverse the decisions. Thus, the prior $\gamma \sim \text{Uniform}(0.5, 1)$ is used.

The script `TTB.txt` implements the graphical model in WinBUGS:

```
# Take The Best
model{
  # Data
  for (q in 1:nq){
    for (i in 1:ns){
      y[i,q] ~ dbern(ttb[t[q]])
      ypred[i,q] ~ dbern(ttb[t[q]])
    }
  }
  # TTB Model For Each Question
  for (q in 1:nq){
    # Add Cue Contributions To Mimic TTB Decision
```

```
    for (j in 1:nc){
      tmp1[q,j] <- (m[p[q,1],j]-m[p[q,2],j])*pow(2,s[j]-1)
    }
    # Find if Cue Favors First, Second, or Neither Stimulus
    tmp2[q] <- sum(tmp1[q,1:nc])
    tmp3[q] <- -1*step(-tmp2[q])+step(tmp2[q])
    t[q] <- tmp3[q]+2
  }
  # Cue Search Order Follows Validities
  for (j in 1:nc){
    s[j] <- rank(v[1:nc],j)
  }
  # Choose TTB Decision With Probability Gamma, or Guess
  ttb[1] <- 1-gamma
  ttb[2] <- 0.5
  ttb[3] <- gamma
  # Priors
  gamma ~ dunif(0.5,1)
}
```

The script simulates the TTB model by comparing all of the cues, but applying non-compensatory weights (the `pow(2,s[j]-1)` multiple) so that the first discriminating cue determines the decision. Note that the script determines the TTB validity-based search order from cues using the `rank` function.

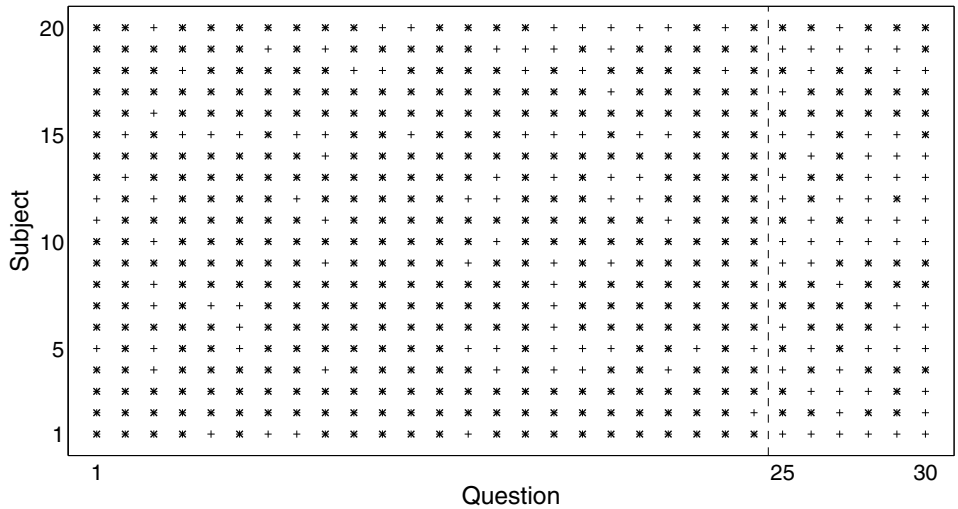


Fig. 18.2 Results of applying the probabilistic TTB model to the decision data.

The code `TTB.m` or `TTB.R` applies the model to the simulated data. Figure 18.2 summarizes the results, showing the agreement between subject decisions, and those produced by the TTB model. The `x` marker indicates that a subject chose stimulus "a," while the absence of a `x` marker indicates that they chose "b." A `+` marker is used to indicate the probability of the model choosing stimulus 'a." A full-size `+` marker means the model chooses "a" with probability one, while the complete

absence of a + marker means the model choses "b" with probability one. Between these two extremes, the size of the + marker represents the probability of "a" being chosen by the model. Thus, agreement between the data and model is visually evident in a * marker, which is the combination of x marker and + marker, showing that both the subject and model chose "a," or the absence of any markers, showing that both chose "b."

The questions in Figure 18.2 are ordered, so that the first 24 are those in which the TTB model, and a more exhaustive model that examines all of the cues, lead to the same decision. The final 6 questions, however—separated by a broken line from the first 24—involve stimulus pairs where TTB makes a different decision. In the presentation of results, but not the presentation to subjects in the experiment, the stimulus pairs are also coded so that choice "a" always corresponds to the TTB choice.

Figure 18.2 shows, as expected, that the model always favors the TTB option. Especially for the first 24 questions, the subjects also often choose the TTB option, and the expected posterior agreement between the model and data over all decisions is calculated by the code as 63%. This is a measure of the descriptive adequacy of the model, and is often called "correspondence" in the judgment and decision-making literature (Dunwoody, 2009).

Exercise

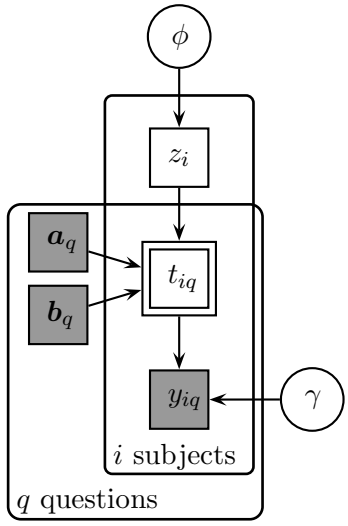
Exercise 18.1.1 What is the posterior expectation of γ, and how can it be interpreted?

18.2 Stopping

A strong assumption of the TTB heuristic is that one discriminating cue is sufficient to trigger a decision. There is debate about this stopping rule, and heuristics that rely on one reason are often contrasted against an alternative model that assumes all of the cues are searched (e.g., Gigerenzer & Goldstein, 1996; Katsikopoulos, Schooler, & Hertwig, 2010).

A common comparison (e.g., Bergert & Nosofsky, 2007; Lee & Cummins, 2004) is between TTB and a model often called the Weighted ADDitive (WADD) model, which sums the evidence for both decision alternatives over all available cues, and chooses the one with the greatest evidence. The evidence provided by a discriminating cue can be determined from its validity, and is naturally measured on the log-odds scale as $x_k = \log \frac{v_k}{1-v_k}$. Thus, the WADD decision heuristic sums all the evidence values for cues that discriminate in favor of each alternative, and chooses the one with the most evidence.

A basic question these data were designed to address is whether subjects consistently used TTB or WADD and, if so, what proportion used each. This is naturally



$$
\begin{aligned}
\phi &\sim \text{Uniform}(0,1) \\
z_i &\sim \text{Bernoulli}(\phi) \\
\gamma &\sim \text{Uniform}(0.5,1) \\
t_{iq} &\leftarrow \begin{cases} \text{TTB}\,(\boldsymbol{a}_q, \boldsymbol{b}_q) & \text{if } z_i = 1 \\ \text{WADD}\,(\boldsymbol{a}_q, \boldsymbol{b}_q) & \text{if } z_i = 0 \end{cases} \\
y_{iq} &\sim \begin{cases} \text{Bernoulli}(\gamma) & \text{if } t_{iq} = a \\ \text{Bernoulli}(1-\gamma) & \text{if } t_{iq} = b \\ \text{Bernoulli}(0.5) & \text{otherwise} \end{cases}
\end{aligned}
$$

Fig. 18.3 Graphical model for a latent mixture of TTB and WADD stopping rules.

addressed using a latent-mixture model, where each subject either uses TTB or WADD to make decisions on every question. Figure 18.3 shows a graphical model that implements this approach. The z_i parameter functions as an indicator variable, with $z_i = 1$ if the ith subject uses TTB, and $z_i = 0$ if they use WADD. This indicator variable is distributed according to a base-rate of TTB subjects in the population, so that $z_i \sim \text{Bernoulli}\,(\phi)$. The deterministic node θ_{ij} for the ith subject is then given by

$$
t_{iq} \leftarrow \begin{cases} \text{TTB}\,(\boldsymbol{a}_q, \boldsymbol{b}_q) & \text{if } z_i = 1 \\ \text{WADD}\,(\boldsymbol{a}_q, \boldsymbol{b}_q) & \text{if } z_i = 0 \end{cases}
$$

with, as before

$$
y_{iq} \sim \begin{cases} \text{Bernoulli}(\gamma) & \text{if } t_{iq} = a \\ \text{Bernoulli}(1-\gamma) & \text{if } t_{iq} = b \\ \text{Bernoulli}(0.5) & \text{otherwise.} \end{cases}
$$

The script `Stop.txt` implements the graphical model in WinBUGS. The basic approach is to find the decision both TTB and WADD makes for each question, and use the indicator `z[i]` to determine how the ith subject makes their decisions:

```
# Stop
model{
  # Data
  for (i in 1:ns){
    for (q in 1:nq){
      y[i,q] ~ dbern(dec[t[i,q,z1[i]]])
      ypred[i,q] ~ dbern(dec[t[i,q,z1[i]]])
    }
  }
  # TTB Decision
  for (i in 1:ns){
```

```
    for (q in 1:nq){
      # Add Cue Contributions To Mimic TTB Decision
      for (j in 1:nc){
        tmp1[i,q,j] <- (m[p[q,1],j]-m[p[q,2],j])*pow(2,s[j]-1)
      }
      # Find if Cue Favors First, Second, or Neither Stimulus
      tmp2[i,q] <- sum(tmp1[i,q,1:nc])
      tmp3[i,q] <- -1*step(-tmp2[i,q])+step(tmp2[i,q])
      t[i,q,1] <- tmp3[i,q]+2
    }
  }
  # WADD Decision
  for (i in 1:ns){
    for (q in 1:nq){
      for (j in 1:nc){
        tmp4[i,q,j] <- (m[p[q,1],j]-m[p[q,2],j])*x[j]
      }
      # Find if Cue Favors First, Second, or Neither Stimulus
      tmp5[i,q] <- sum(tmp4[i,q,1:nc])
      tmp6[i,q] <- -1*step(-tmp5[i,q])+step(tmp5[i,q])
      t[i,q,2] <- tmp6[i,q]+2
    }
  }
  # Follow Decision With Probability Gamma, or Guess
  dec[1] <- 1-gamma
  dec[2] <- 0.5
  dec[3] <- gamma
  # Cue Search Order Follows Validities
  for (j in 1:nc){
    stmp[j] <- nc-j+1
    s[j] <- rank(v[1:nc],stmp[j])
  }
  # TTB and WADD Subjects in Latent Mixture
  for (i in 1:ns){
    z[i] ~ dbern(phi)
    z1[i] <- z[i]+1
  }
  # Priors
  gamma ~ dunif(0.5,1)
  phi ~ dbeta(1,1)
}
```

The code `Stop.m` or `Stop.R` applies the model to the same simulated data. Figure 18.4 summarizes the results, and calculates an improved 68% correspondence with the human decisions. Most of the improvement comes from the final 6 questions, where TTB and WADD make different decisions. Those people—like subjects 17, 18, 19, and 20—who did not make TTB-consistent decisions for the last 6 questions are better modeled.

Figure 18.5 highlights this point by showing the posterior expectation for the latent-mixture indicator variable for each subject. At least half the subjects—including subjects 17, 18, 19 and 20—are inferred, with high certainly, to belong to the WADD group.

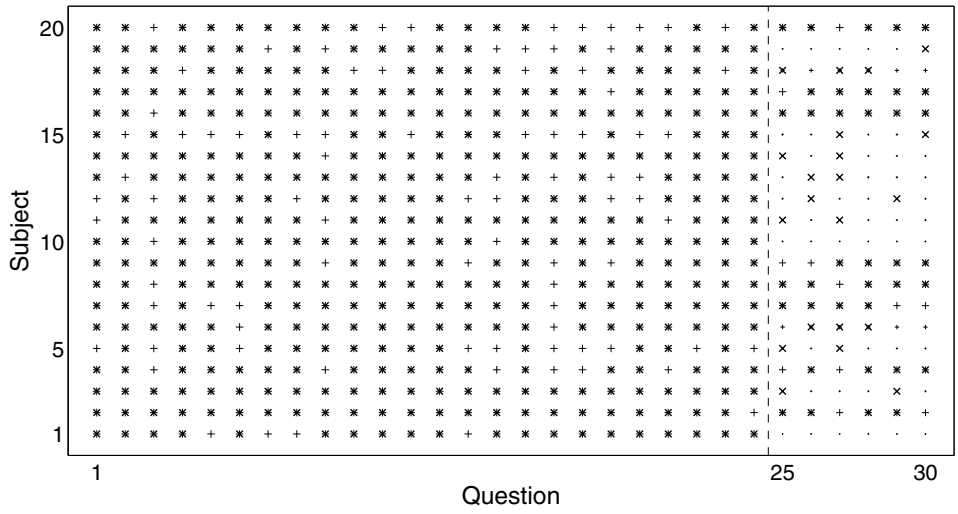


Fig. 18.4 Results of applying the latent mixture of the TTB and WADD models to the decision data.

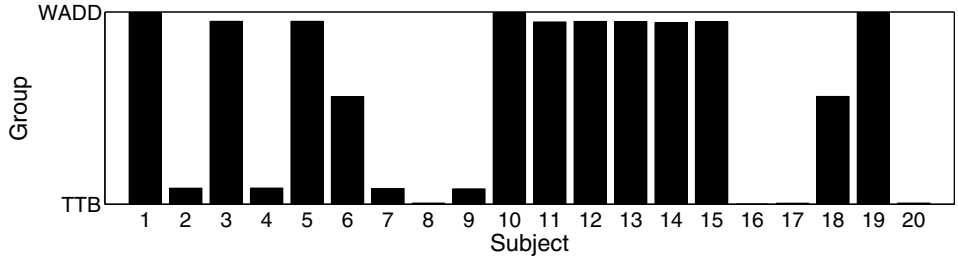


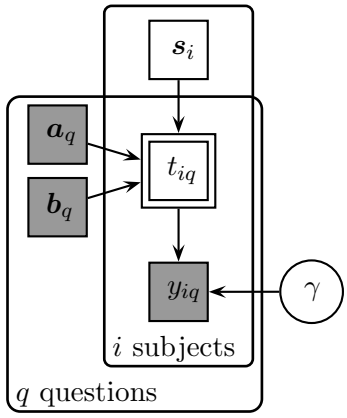
Fig. 18.5 Inferences about the use of the TTB and WADD models by each subject.

Exercise

Exercise 18.2.1 Plot and interpret the posterior distribution of ϕ. Approximate and interpret the Bayes factor comparing $\mathcal{H}_0 : \phi = 0$ versus $\mathcal{H}_1 : \phi \neq 0$.

18.3 Searching

Another way to extend the initial TTB account of the decision data is to preserve the one-reason stopping rule, but allow for flexibility in the search order. Most simply, it would be possible to consider a model in which everybody used a search order that did not follow cue validities. But, given the obvious individual differences in the data themselves, a more interesting model allows different people to have different search orders.



$$
\begin{aligned}
s_i &\sim \text{Uniform}((1,\ldots,9),\ldots,(9,\ldots,1)) \\
t_{iq} &\leftarrow \text{TTB}_{s_i}(a_q, b_q) \\
\gamma &\sim \text{Uniform}(0.5, 1) \\
y_{iq} &\sim \begin{cases} \text{Bernoulli}(\gamma) & \text{if } t_{iq} = a \\ \text{Bernoulli}(1-\gamma) & \text{if } t_{iq} = b \\ \text{Bernoulli}(0.5) & \text{otherwise} \end{cases}
\end{aligned}
$$

Fig. 18.6 Graphical model for individual differences in search orders.

The graphical model shown in Figure 18.6 implements this idea. The model allows each subject to have their own search order s_i, and treats these orders as latent parameters to be inferred from the data. Since the search orders are now model parameters, they must be given priors. In the graphical model, each of the $9! = 362,880$ search orders is made equally likely. That is, there is a uniform prior over all possible values of the search order parameters.

The script `Search.txt` implements the graphical model in WinBUGS:

```
# Individual Search Orders
model{
  # Data
  for (i in 1:ns){
    for (q in 1:nq){
      y[i,q] ~ dbern(ttb[t[i,q]])
      ypred[i,q] ~ dbern(ttb[t[i,q]])
    }
  }
  # One Reason Model, With Different Search Order Per Subject
  for (i in 1:ns){
    for (q in 1:nq){
      for (j in 1:nc){
        tmp1[i,q,j] <- (m[p[q,1],j]-m[p[q,2],j])*pow(2,s[i,j]-1)
      }
      # Find if Cue Favors First, Second, or Neither Stimulus
      tmp2[i,q] <- sum(tmp1[i,q,1:nc])
      tmp3[i,q] <- -1*step(-tmp2[i,q])+step(tmp2[i,q])
      t[i,q] <- tmp3[i,q]+2
    }
    # Cue Search Order From Ranking stmp
    for (j in 1:nc){
      s[i,j] <- rank(stmp[i,1:nc],j)
      stmp[i,j] ~ dnorm(0,1)I(0,)
    }
  }
  ttb[1] <- 1-gamma
  ttb[2] <- 0.5
  ttb[3] <- gamma
```

```
  # Prior
  gamma ~ dunif(0.5,1)
}
```

The generation of search orders has to be implemented in a low-level way, since WinBUGS has no standard distribution for placing a prior over orders. One approach would be to have a categorical variable that could take 9! values, each corresponding to a search order. Doing inference on this variable, however, would be hampered by the lack of structure in the relationships between the values. If two search orders are similar (say they have the same order, except for two neighboring cues, whose order in search are flipped) it would help if they had similar (i.e., neighboring) categorical numbers. But it is obviously impossible to map all the structure relationships between the orders onto the number line.

A better approach is to create an underlying continuous scale, and have each cue take a value on that scale. A search order can then be given simply by the order of those values, and changing the value for a cue in sampling-based inference would only change the order slightly, if at all. Thus, the `Search.txt` script uses underlying continuous positive variables for each of the cues, each with a Gaussian$(0, .001)$ prior.

The code `Search.m` or `Search.R` applies the model to the decision data. Figure 18.7 summarizes the new results, and calculates the correspondence to be 73%. The results show that this model is able to fit patterns in individual subject behavior that neither TTB, nor a mixture of TTB and WADD, could accommodate. Question 17 provides a good example, since about half the subjects choose one alternative, but half choose the other. This question requires choosing between an alternative that is defined by the presence of cue 2, and an alternative that is defined by the presence of cue 6. That is, both alternatives have only one cue, and it is cue 2 for one alternative, and cue 6 for the other. The first of these alternatives is chosen by TTB, since cue 2 has greater validity than cue 6. Thus, one explanation for the evident individual differences across subjects for this question is that some subjects search for cue 2 before 6, while others search for cue 6 before cue 2.

This speculation can be pursued by considering the posterior distribution over search orders inferred by the model. Conceptually, the posterior for an individual subject's search order is a distribution over the 9! possible orders. One way to display this is to list the orders in decreasing order of mass, as estimated from the number of times each is sampled. The code `Search.m` or `Search.R` displays how many unique orders were sampled, and then details the 10 specific orders with the highest posterior mass estimates.

To make this concrete, we consider the search orders inferred for subjects 12 and 13, who make different decisions for question 17. The summary of the posterior for the search patterns of subject 12 gives the following 10 orders.

```
Subject 12
There are 3709 search orders sampled in the posterior.
Order=(2  6  9  7  1  5  3  4  8), Estimated Mass=0.0013
```

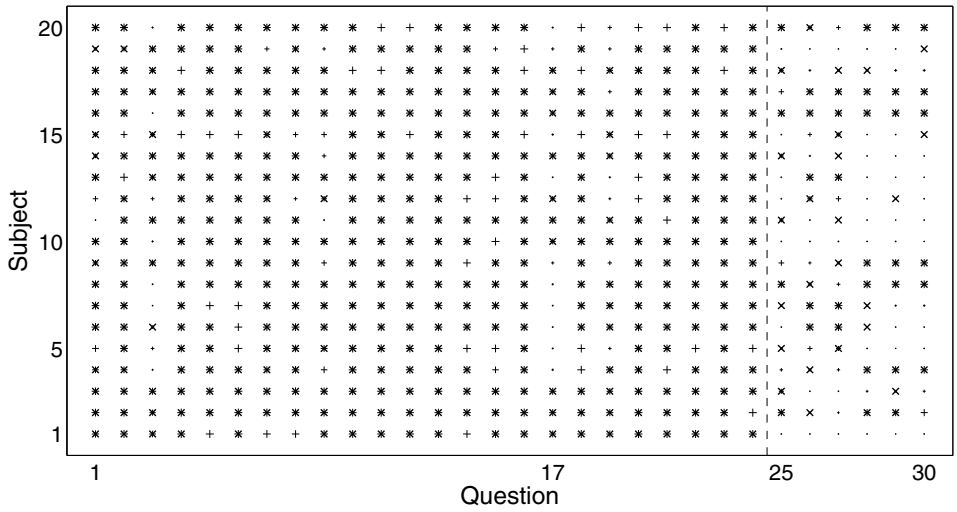


Fig. 18.7 Results of applying the individual search order model to the decision data.

```
Order=(2  6  5  9  7  1  8  3  4), Estimated Mass=0.0010
Order=(5  2  6  9  1  8  4  7  3), Estimated Mass=0.0010
Order=(2  6  5  9  1  8  7  3  4), Estimated Mass=0.0010
Order=(2  6  9  7  1  5  3  8  4), Estimated Mass=0.0010
Order=(2  6  9  5  1  3  4  7  8), Estimated Mass=0.0010
Order=(2  6  9  1  5  7  3  4  8), Estimated Mass=0.0010
Order=(2  6  5  1  3  9  7  4  8), Estimated Mass=0.0010
Order=(2  6  1  9  5  4  8  3  7), Estimated Mass=0.0010
Order=(2  1  6  9  7  5  4  8  3), Estimated Mass=0.0010
```

It is clear that, despite the uncertainty about the search order expressed by this posterior distribution, cue 2 is searched before cue 6. The summary of the posterior for the search patterns of subject 13 gives the following 10 orders.

```
Subject 13
There are 3350 search orders sampled in the posterior.
Order=(5  6  7  1  8  4  3  2  9), Estimated Mass=0.0018
Order=(5  4  6  1  7  3  2  8  9), Estimated Mass=0.0015
Order=(5  6  4  3  2  7  1  8  9), Estimated Mass=0.0013
Order=(5  6  7  8  1  9  4  3  2), Estimated Mass=0.0013
Order=(5  6  7  9  1  3  4  8  2), Estimated Mass=0.0013
Order=(5  6  4  7  1  9  2  8  3), Estimated Mass=0.0013
Order=(5  4  6  1  3  7  8  9  2), Estimated Mass=0.0013
Order=(5  6  4  8  7  1  2  9  3), Estimated Mass=0.0010
Order=(5  6  4  8  9  1  2  3  7), Estimated Mass=0.0010
Order=(5  4  6  9  7  1  2  3  8), Estimated Mass=0.0010
```

It seems clear for this subject that cue 6 is searched before cue 2. The difference in the order subjects 12 and 13 provides an explanation for their different decisions on question 17.

Exercises

Exercise 18.3.1 Look through the search order posterior distribution summaries for all of the subjects. How would you characterize the uncertainty they represent? Think about how many search orders are sampled, how many could be sampled, and how similar those sampled are to one another.

Exercise 18.3.2 Do you expect to be able to make inferences about the full search order if a subject is using a one-reason stopping rule like TTB? What consequences for analysis does this issue have?

Exercise 18.3.3 Are the inferred search orders for all of the subjects, and not just subjects 12 and 13, consistent with the individual differences observed in the answers to question 17?

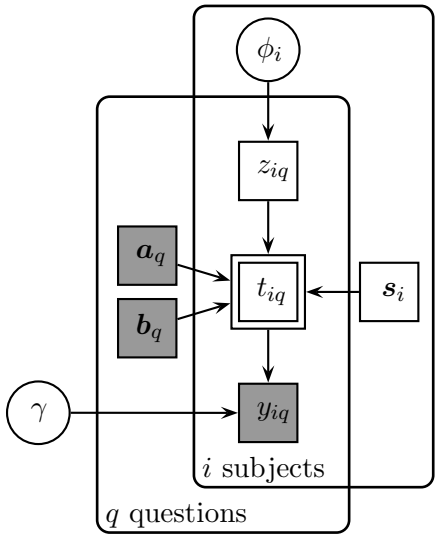
18.4 Searching and stopping

The previous two models extended the original TTB model in two ways, to try and account for individual differences in the decisions. One model incorporated different stopping rules as the source of individual differences, while the other incorporates different search orders. The natural theoretical progression is to consider a model that allows for both different stopping rules and search orders.

Figure 18.8 presents a graphical model that combines searching and stopping. It is almost the logical combination of the stopping graphical model in Figure 18.3 and the searching graphical model in Figure 18.6, with one small adjustment. The stopping model considered each subject to use either the TTB or WADD stopping rule for all of their decisions, and inferred a base-rate over subjects. The model in Figure 18.8, however, assumes there is a base-rate ϕ_i over all of the questions with which the ith subject uses the WADD rather than the TTB stopping rule. Thus, there is an indicator z_{iq} corresponding to the stopping rule used by the ith subject on the qth question. Of course, if the WADD stopping rule is used, the search order does not matter, since all of the cues are examined. But, when the TTB one-reason stopping rule is used, the search order $\boldsymbol{s}_i$ for the ith subject influences the decision.

The script `SearchStop.txt` implements the graphical model in WinBUGS:

```
# Search and Stop
model{
  # Data
  for (i in 1:ns){
    for (q in 1:nq){
      y[i,q] ~ dbern(dec[t[i,q,z1[i,q]]])
      ypred[i,q] ~ dbern(dec[t[i,q,z1[i,q]]])
    }
```



$$
\begin{aligned}
\phi_i &\sim \text{Uniform}(0,1) \\
z_{iq} &\sim \text{Bernoulli}(\phi_i) \\
\boldsymbol{s}_i &\sim \text{Uniform}((1,\ldots,9),\ldots,(9,\ldots,1)) \\
\gamma &\sim \text{Uniform}(0.5,1) \\
t_{iq} &\leftarrow \begin{cases} \text{TTB}_{\boldsymbol{s}_i}(\boldsymbol{a}_q,\boldsymbol{b}_q) & \text{if } z_{iq}=1 \\ \text{WADD}(\boldsymbol{a}_q,\boldsymbol{b}_q) & \text{if } z_{iq}=0 \end{cases} \\
y_{iq} &\sim \begin{cases} \text{Bernoulli}(\gamma) & \text{if } t_{iq}=a \\ \text{Bernoulli}(1-\gamma) & \text{if } t_{iq}=b \\ \text{Bernoulli}(0.5) & \text{otherwise} \end{cases}
\end{aligned}
$$

Fig. 18.8 Graphical model for inferring search orders and stopping rules.

```
}
# TTB Decision
for (i in 1:ns){
  for (q in 1:nq){
    for (j in 1:nc){
      tmp1[i,q,j] <- (m[p[q,1],j]-m[p[q,2],j])*pow(2,s[i,j]-1)
    }
    tmp2[i,q] <- sum(tmp1[i,q,1:nc])
    tmp3[i,q] <- -1*step(-tmp2[i,q])+step(tmp2[i,q])
    t[i,q,1] <- tmp3[i,q]+2
  }
}
# WADD Decision
for (i in 1:ns){
  for (q in 1:nq){
    for (j in 1:nc){
      tmp4[i,q,j] <- (m[p[q,1],j]-m[p[q,2],j])*x[j]
    }
    # Find if Cue Favors First, Second, or Neither Stimulus
    tmp5[i,q] <- sum(tmp4[i,q,1:nc])
    tmp6[i,q] <- -1*step(-tmp5[i,q])+step(tmp5[i,q])
    t[i,q,2] <- tmp6[i,q]+2
  }
}
# Follow Decision With Probability Gamma, or Guess
dec[1] <- 1-gamma
dec[2] <- 0.5
dec[3] <- gamma
# Cue Search Order From Ranking stmp
for (i in 1:ns){
  for (j in 1:nc){
    s[i,j] <- rank(stmp[i,1:nc],j)
```

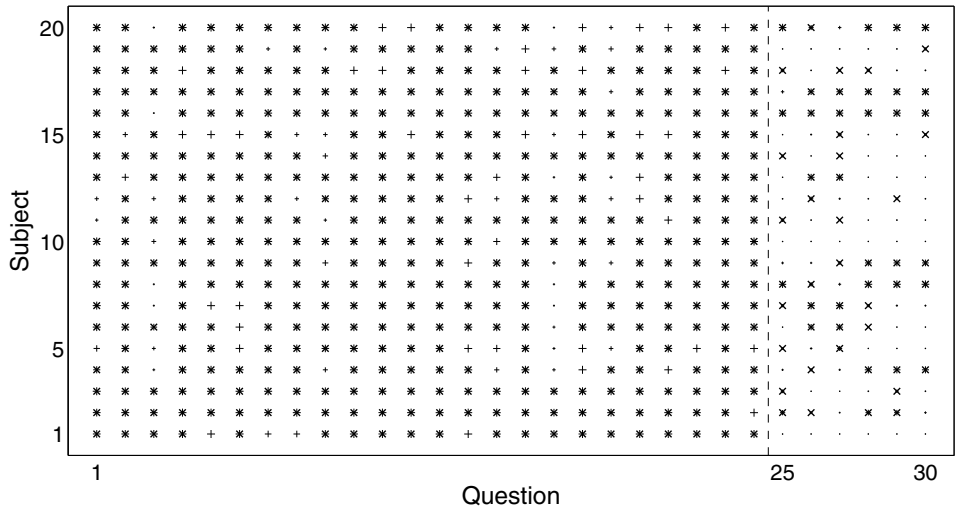


Fig. 18.9 Results of applying the flexible searching and stopping model to the decision data.

```
      stmp[i,j] ~ dnorm(0,1)I(0,)
    }
  }
  # TTB and WADD Rate Per Subject
  for (i in 1:ns){
    phi[i] ~ dbeta(1,1)
    for (q in 1:nq){
      z[i,q] ~ dbern(phi[i])
      z1[i,q] <- z[i,q]+1
    }
  }
  gamma ~ dunif(0.5,1)
}
```

The code `SearchStop.m` or `SearchStop.R` applies the model to the decision data. Figure 18.9 summarizes the results, and calculates the correspondence to be 74%. The incorporation of individual differences into the stopping rule does not appear to have had much impact on the posterior predicted decisions made by the search model in Figure 18.7.

Exercises

Exercise 18.4.1 How do the posterior expectations of the ϕ_i parameters for the searching and stopping model in Figure 18.8 compare to the posterior expectations of the z_i parameters for the stopping model in Figure 18.3? Interpret the similarities and differences.

Exercise 18.4.2 In what sense does the current model incorporate, and not incorporate, structured individual differences over searching and stopping? Suggest hierarchical extensions to the model in Figure 18.8 that could add structure to the individual differences in searching and stopping parameters.