Summer Institute on Bounded Rationality, Workshop 3 (June 9th, 2015):

Building Fast and Frugal Trees (FFTs)

Lael J. Schooler[1,3], Uwe Czienskowski[1], Hansjörg Neth[1,2] , Evaldas Jablonskis[1]
[1:] Max Planck Institute for Human Development, Berlin, Germany
[2:] Social Psychology and Decision Sciences, University of Konstanz, Germany
[3] Syracuse University

Fast and Frugal Trees (FFTs) are a quintessential family of simple heuristics that allow effective and efficient binary classification decisions and often perform remarkably well when compared to more complex methods. This half-day tutorial will familiarize participants with examples of FFTs and elucidate the theoretical link between FFTs and signal detection theory (SDT). A range of presentations, practical exercises and interactive tools will enable participants to construct and evaluate FFTs for different data sets.

Introduction

A *fast-and-frugal tree* (FFT) is a decision algorithm that sequentially checks up to m cues before making a binary classification decision. As there is at least one exit on any level (i.e., any cue yields a classification decision) a FFT has exactly $m + 1$ exits: one exit for each of the first $m - 1$ cues and two exits for the m^{th} and final cue. (See Martignon, Vitouch, Takezawa & Forster, 2003; Martignon, Katsikopoulos, & Woike, 2008; and Luan, Schooler, & Gigerenzer, 2011 for details).

This part of the tutorial will familiarize you with basic methods in constructing and assessing FFTs by exploring them with WebFFTBuilder (Jablonskis, Mihan, Czienskowski, 2015). To start using WebFFTBuilder go to:
http://www.dotwebresearch.net/divapps/fftree/

By the end of this part, you should be able to

- explore the predictive validity of individual cues by building minimal FFTs;
- combine cues to construct complex FFTs that are effective and efficient;
- evaluate the performance of a given FFT on different data sets.

We will first illustrate the functionality of WebFFTBuilder by re-constructing and evaluating a FFT that has been developed by Green & Mehr (1997).

The Green and Mehr (1997) Tree. A key property of FFTs is that they allow for rapid and robust decisions in real world settings, yet nevertheless have a simple and transparent structure. The tree developed by Green & Mehr (1997) exhibits these properties in an exemplary fashion (see Figure 1). By sequentially checking a maximum of three cues (whose meaning is described in more detail below) an emergency-room doctor can use this FFT to decide whether a patient concerned about a possible heart attack should be sent to the coronary care unit (CCU) or a regular nursing-bed. This is a life-and-death decision that has to be made under severe time-pressure. It is of vital importance that the decision algorithm does not assign a patient with a coronary infarction to a regular nursing bed (which would be considered a *miss*). However, as unnecessary treatments are dangerous and costly, a patient without heart disease should not be sent to the CCU (i.e., avoid *false alarms*).

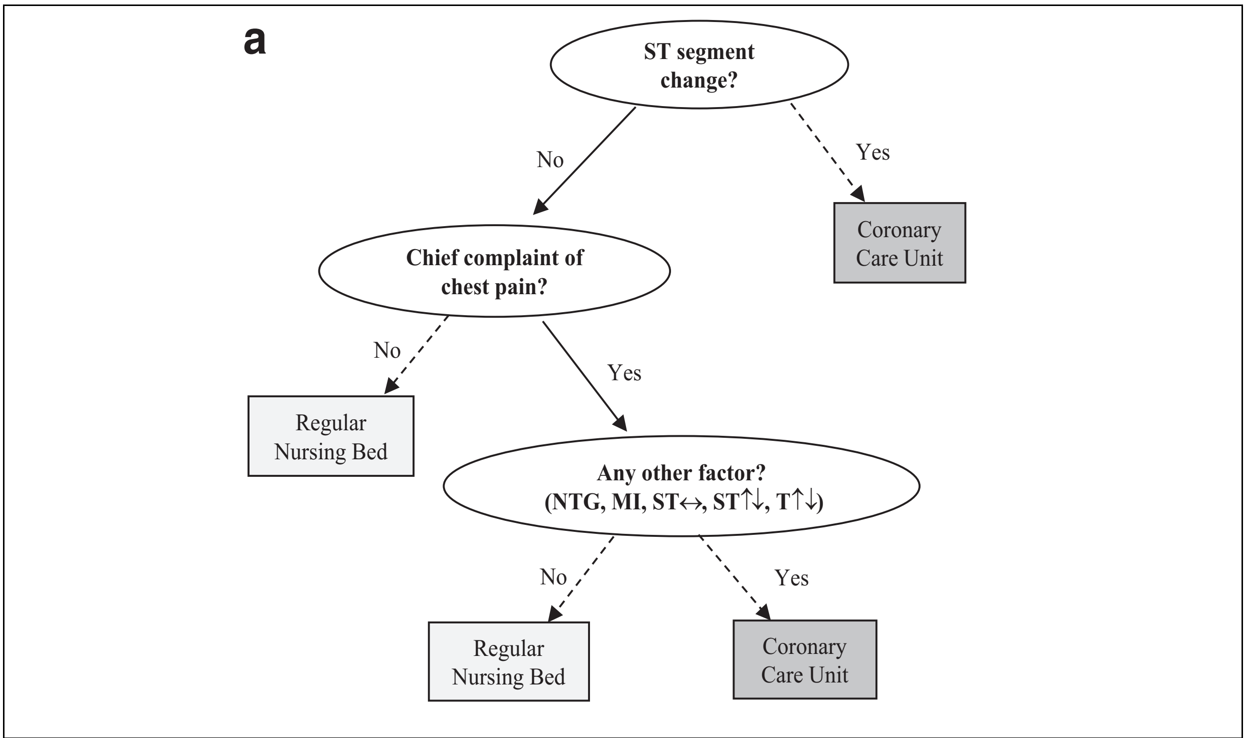


Figure 1: A fast-and-frugal tree designed by Green and Mehr (1997) to help emergency–room doctors to rapidly decide whether to send a patient with severe chest pain to the coronary care unit (CCU) or a regular nursing bed. The meaning of the cues is described in the text. (Image from Luan, Schooler, & Gigerenzer, 2011, Figure 4a).

A necessary precondition of building a FFT is to select a data set of cases that are to be classified. Suitable data sets must contain one or more *predictor* variables (so-called *cues*), which are either binary or are made binary by setting a threshold value (or *split* value), and a *criterion* variable, which indicates the true category membership of each case, i.e., whether the case is to be classified as a positive instance ("yes", criterion = 1) or as a negative instance ("no", criterion = 0).

Data set. The reconstructed data set of Green and Mehr (1997) is contained in the CSV file "data_CCU_FFTorg" and contains N = 89 cases. Each case is characterized by four values: A binary criterion value (0 indicating "no infarction", 1 indicating an "infarction", as measured by the presence or absence of a subsequent heart attack) and three binary cues, labeled as "ST", "CP", and "OC". "ST" refers to an altered segment in the ECG profile, "CP" notes whether or not the chief complaint is chest pain, and "OC" is a compound cue that notes the presence or absence of one or more other cues. All cues are coded in a binary fashion so that a positive cue value (of 1) indicates the presence and a negative value (of 0) indicates the absence of the respective cue.

Creating and Evaluating FFTs

WebFFTBuilder is organized into three basic sections:

A. *Blue Area* , after the dataset is loaded, it contains a range of predictor variables (cues) for a number of individual cases.

B. *White Empty Areas on the left and on the right are the locations to build two FFTs.*

C. *In the middle of the White Area are the tables of statistics, which* evaluates the current FFT by indicating the accuracy of the predictions for each case, classifying the predictions into different cases (as defined by signal detection theory, SDT), and computing some corresponding counts and summary statistics.

Exercise 0 **Constructing the Green & Mehr tree (1997)**

1. **Load** the CSV dataset "data_CCU_FFTorg" (choose from the list).
2. **Select** "infarction" as a Criterion (click the circle button. It will turn gray).
3. **Choose** cues
   (drag and drop to the White Area the three cues ST, CP, and OC in that order).

4. **Define** a test for the first selected cue (**ST**):
   The test being conducted on the ST values is:
   If (ST > .5), then: predict a *positive* criterion value "yes" (of 1);
   else: continue with the next cue.
   Result: Note the corresponding predictions for all cases (expand the cue):
   33 cases are predicted as "1"/"yes" (indicating an infarction);
   56 cases are deferred to the next cue.

5. **Define** a test for the second selected cue (**CP**):
   The test being conducted on the CP values is:
   If (CP < .5), then: predict a *negative* criterion value "no" (of 0);
   else: continue with the next cue.
   Result: Note the corresponding predictions for the 56 remaining cases (expand the cue):
   29 cases are predicted as "0" /"no" (indicating no infarction);
   27 cases are deferred to the next cue;
   (Note that of the original 33 cases have already been processed)

6. **Define** a test for the third selected cue (**OC**):
   The test being conducted on the OC values is:
   If (OC < .5), then: predict a *negative* criterion value "no" (of 0);
   else: predict a *positive* criterion value "yes" (of 1).

   Result: Of the remaining 27 cases 10 cases are predicted as "0"/"no" (no infarction) and 17 cases are predicted as "1"/"yes" (infarction).

7. **Evaluation** of classification outcomes:
   For a well-defined FFT, every case results in exactly one prediction (see *statistics in the expanded cues*). These predictions are evaluated in *statistics of the tree*:
   - *counts* the resulting predictions for all cases.
   - determines the accuracy of these predictions by comparing them with the true criterion values.
   - compares the prediction and criterion values and further categorizes correct and incorrect classifications into four possible cases:
     - *Hits*: prediction = 1 ("yes") and criterion = 1 ("yes").
     - *False alarms (FA)*: prediction = 1 ("yes") but criterion = 0 ("no").
     - *Misses*: prediction = 0 ("no") but criterion = 1 ("yes").
     - *Correct rejections (CR)*: prediction = 0 ("no") and criterion = 0 ("no").

   These classification outcomes are counted and summarized in the **2x2 contingency table** to the right, together with various indicators of the FFT's efficiency and effectiveness:

   - **Frugality**: The mean number of steps or decisions per case [here: 1.93].
   - **Accuracy**: the difference between *p*(hit) and *p*(FA) [here: 0.521], and the *d'*-value from Signal Detection Theory [2.603], corrected for a *p*(hit)=1.0].
   - **Bias**: the *c*-value from Signal Detection Theory [here -1.234]. Larger (positive) values indicate conservative FFTs, smaller (negative) values indicate liberal FFTs. Note this is a different measure of bias than the one introduced in the lecture.

8. **Changing cue properties:**
The test for the last cue (OC) currently is:

If (OC < .5), then: predict a *negative* criterion value "no" (of 0);
else: predict a *positive* criterion value "yes" (of 1).

(a) *Reversing the test direction*: What if positive values of the final cue (i.e., OC > .5) predicted a *negative* criterion value ("no")? Let's test this hypothesis by swaping "yes" and "no" in the expanded OC.

- Note that this also flips the values (e.g., from 10 predicted instances of "no" and 17 predicted instances of "yes" to 17 predictions of "no" and "10 predictions of "yes").
- **Accuracy**: Note that the overall accuracy of the FFT decreases:
  p(hit) – p(FA) [now: 0.462], *d'*-value [now 1.35].
- **Frugality**: As the change only affected the last cue, the mean number of steps or decisions per case remains the same [1.93].
- **Bias**: the *c*-value from Signal Detection Theory [here -0.436].

9. **Changing cue order:**

To change the *order* of cues, simply drag and drop the cue OC between ST and CP. Change the settings for the OC cue back to its original settings (see step 6). Note that this changes the number of cases that are classified on levels 2 and 3 (e.g., there are now 19 + 17 = 36 (instead of 27) cases that are classified by the 3rd cue). Correspondingly, the frugality measure increases from 1.93 to 2.03 steps on average (indicating slightly more classification steps on average). However, the classification outcomes (summarized in the contingency table) and all measures of prediction accuracy remain unaffected. This insensitivity to a change in cue order illustrates the *partial order invariance* rule of FFTs (see Luan et al., 2011, p. 324, for details).

10. **Truncating the FFT** by removing the 3rd cue:

[Change the order in the tree back to the original cue order: ST, CP, OC.]

Suppose you wanted to truncate the FFT by removing its 3rd cue (click X on the cue OC). Note that the resulting 2-cue FFT is more frugal than the original 3-cue FFT, but still displays decent performance [with an accuracy of p(hit) – p(FA) = 0.386 and d' = 2.261.

Exercise 1: Exploring exit structures (using the same dataset "data_CCU_FFTorg")

In general, a total of m binary cues allow for the construction of $m! \times 2^{m-1}$ different FFTs. Whereas the first term ($m!$) indicates the number of possible cue orders, the second term (2^{m-1}) indicates the number of different exit structures for a given cue order.

Above we have seen that the FFT proposed by Green and Mehr (1997) used three cues (ST, CP, OC, in that order) with a particular exit structure. Holding the order of these three cues constant still allows for $2^{3-1} = 4$ possible FFTs that differ in their exit structures. For instance, a patient with a high value on the initial ST cue (>.5, "yes") was immediately assigned to the "CCU" category (=1, "infarction" is "yes") by Green and Mehr (1997). Alternatively, it would have been just as plausible to assign a patient with a low value on the ST cue (< .5, "no") to a "regular nursing bed" (= 0, "infarction" is "no) and continue inspecting the second cue (CP) in case of a high value on the ST cue.

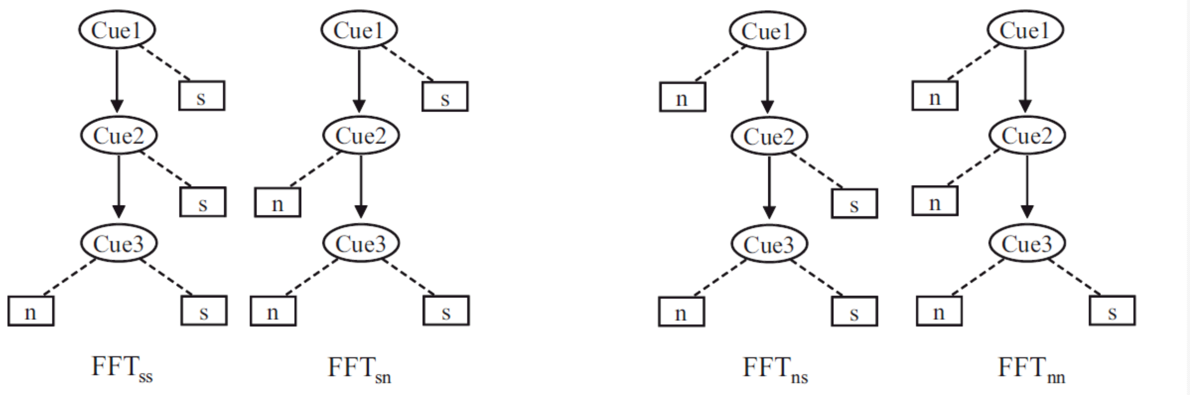
Luan et al. (2011) have shown that different exit structures are FFT's analogue to *criterion shifts* in SDT, i.e., changing the exit structure of a FFT affects the *bias* or *criterion values c* of the resulting FFT. (See Table 1 for the definition the bias index *c*.)

Your task in this exercise is to construct all four trees containing the same cue order (ST, CP, OC) but different *exit structures* (see Figure 2) and to inspect and compare their respective bias values *c*. How does a FFT's exit structure affect its *bias* or *criterion value c*?

Table 1: Definitions of SDT measures according to Luan et al. (2011, p. 319).

List of Commonly Used Signal-Detection Theory Terms, Their Definitions/Measurements, and Some Explanatory Notes

Term	Definition/measurement	Notes[a]
Hit	Respond "signal" given signal	Complementary to miss
False alarm (FA)	Respond "signal" given noise	Complementary to correct rejection (CR)
d'	zHit − zFA	zHit and zFA are the z scores of the hit rate and the FA rate, respectively; d' reflects the standardized distance between the signal (S) and noise (N) distributions
c	−0.5 × (zHit + zFA)	$c > 0$: *conservative* bias, making more noise than signal decisions relative to prior probabilities $c < 0$: *liberal* bias, making more signal than noise decisions relative to prior probabilities $c = 0$: *neutral* bias, making decisions consistent with prior probabilities



more liberal <---> **more conservative**

Figure 2: Four different exit structures with the same cue order. By convention, exits to the right are classified as signals (s), and exits to the left are classified as noise (n). (From Luan et al., 2011, p. 318.)

Solutions to Exercise 1

Exercise: **Demonstrating criterion shifts (bias).**

Exploring 4 different exit structures for same cue order (ST, CP, OC):

1 **ss:**

FFT Cues & settings: IF	**ST**	**CP**	**OC**	d' = 1.433	
direction:	**>**	**>**	<	corrected for	
split value:	**0.5**	**0.5**	0.5	c =	
THEN: prediction:	**1** ("yes")	**1** ("yes")	0 ("no")	-1.819	= most **liberal** FFT

2 **sn:**

FFT Cues & settings: IF	**ST**	**CP**	**OC**	d' = 2.603
direction:	**>**	**<**	<	corrected for
split value:	**0.5**	**0.5**	0.5	c =
THEN: prediction:	**1** ("yes")	**0** ("no")	0 ("no")	-1.234

3 **ns:**

FFT Cues & settings: IF	**ST**	**CP**	**OC**	d' = **1.81**
direction:	**<**	**>**	<	
split value:	**0.5**	**0.5**	0.5	c =
THEN: prediction:	**0** ("no")	**1** ("yes")	0 ("no")	-0.207

4 **nn:**

FFT Cues & settings: IF	**ST**	**CP**	**OC**	d' = **1.24**	
direction:	**<**	**<**	<		
split value:	**0.5**	**0.5**	0.5	c =	
THEN: prediction:	**0** ("no")	**0** ("no")	0 ("no")	**0.37**	= most **conservative** FFT

2 – Minimal FFTs: Exploring the Predictive Validity of Individual Cues

The CSV file "data_CMC_1cue" contains a new data set for which we will explore the predictive properties of individual cues. Essentially, we will look at a minimal FFT for each cue. Such FFTs are *minimal* as they contain only a single cue and immediately result into two exits.

Data set. The data used in this exercise was obtained from the *UCI Machine Learning Repository* (http://archive.ics.uci.edu/ml/) and is a subset of the 1987 *National Indonesia Contraceptive Prevalence Survey*. The N = 1473 cases are samples of married women who were either not pregnant or did not know if they were pregnant at the time of interview. Table 1 provides an overview of the variables contained in the data set.

Objective. Assume that you are working for an organization that aims to promote a better contraceptive method to those women who are using contraceptives, i.e., identifying a woman currently using contraceptives would be considered as a "hit". Thus, the goal of this task is to predict whether a woman is using contraception (with CMC = 0 indicating "non-use" vs. CMC = 1 indicating "use") based on her demographic and socio-economic characteristics.

Table 1: Overview of the criterion and cue variables in the CMC data set.

Criterion:			
	Contraceptives used	(binary)	0=no-use, 1=use *
Cues:			
1.	Wife's age	(numerical)	min=16, max=49
2.	Wife's education	(categorical)	1=low, 2, 3, 4=high
3.	Husband's education	(categorical)	1=low, 2, 3, 4=high
4.	N of children ever born	(numerical)	min=0, max=16
5.	Wife's religion	(binary)	0=non-Islam, 1=Islam
6.	Wife's now working?	(binary)	0=no, 1=yes *
7.	Standard-of-living index	(categorical)	1=low, 2, 3, 4=high
8.	Media exposure	(binary)	0=low, 1=high *

*: Cue values have been re-coded (inverted or binned) relative to original data set.

Exploring the cues contained in a data file by constructing minimal FFTs requires the following steps:

1) Selecting a Criterion (click the circle button on the first variable "cm_bin")
2) Selecting a cue to be explored (e.g., cue *X, click "V" to expand the cue*).
3) For every selected cue *X*, defining a test (e.g., "*X* > .5 = 1", "yes") that results in a prediction (i.e., reaches an exit with a classification decision).

   Defining such a test requires three sub-steps:
   a) setting the direction of the test (swap "yes" and "no");
   b) setting the split value (e.g., .5 in the case of binary cues) with the slider;

Exercise 2.1 (using the dataset "data_CMC_1cue")

1. Exploring a first **cue** (*n_child*):
   A variable that is likely to be related to the use or non-use of contraceptives is the number of children (*n_child*). Note that this cue is non-binary, but contains numeric values from 0 to 16.

   How could the cue *n_child* relate to the criterion of use or non-use of contraceptives? The first hypothesis we will investigate is whether married women without children are childless because they are using contraceptives. An appropriate test could be stated as:

   t1: If: (*n_child* < 1, split value "0.5"),
   then: predict a *positive* criterion value "yes" (of 1);
   else: predict a *negative* criterion value "no" (of 0).

   The results are spectacularly bad: The contingency table shows that this FFT predicts the use of contraceptives (i.e., having a high criterion value) for only 97 cases. Of those, merely 2 are *hits* (i.e., correctly predicted to use contraceptives), whereas the remaining 95 cases are *false alarms* (i.e., women erroneously predicted to use contraceptives). A large majority of 1376 cases in the data set were women with one or more children and were hence predicted to not use contraceptives (0). Of those, 534 were *correct rejections* (i.e., actually did not use contraceptives), whereas 842 were *misses* (i.e., did use contraceptives). A glance at the summary measures confirms the abysmal performance of these cue settings, with a $p(H) = .00$, $p(FA) = .15$, and $d' = -1.79$.

   (b) Whereas our first hypothesis concerning the *n_child* cue may have been incorrect, this result teaches us an important fact about the data set: Contrary to our initial hypothesis, women without children are *unlikely* to use contraceptives. This insight leads to a second hypothesis to be explored:

Perhaps married women in this data set are only using contraceptives *after* giving birth to one or more children? A minimal FFT that tests whether women with children (i.e., for which it is the case that *n_child* > 0) *are* using contraceptives

t2: If: (*n_child* > 0, split value 0.5),
then: predict a *positive* criterion value "yes" (of 1);
else: predict a *negative* criterion value "no" (of 0).

Please verify that p(hit) – p(FA) = .15, and d' = 1.79 in both cases.

A problem of all these FFTs is that they divide the data set into one subset of 97 cases (6.6% of all cases) and a second subset of 1376 cases (93.4%). To get two subsets of more similar sizes we need to increase the split point value. The test

t3: If: (*n_child* > 1, split value 1.5),
then: predict a *positive* criterion value "yes" (of 1);
else: predict a *negative* criterion value "no" (of 0).

yields $p(hit) - p(FA)$ = .22, and d' = 0.68. Incrementing the split point value to 2, 3 and beyond does not yield better results (in terms of d') despite successfully changing the subset sizes.

2. Exploring a second **cue** (*age*): - expand the cue.
How does the cue *age* relate to the use or non-use of contraceptives?
Entering different age values yields the following d'-values:
if *age* < 20 (split 19.5), predict 1 ("yes") d = -.13
if *age* < 30 (split 29.5), predict 1 ("yes") d = .03
if *age* < 40 (split 39.5), predict 1 ("yes") d'= .43
if *age* < 45 (split 44.5), predict 1 ("yes") d'= **.60** (the maximal d' for this cue)
if *age* < 46 (split 45.5), predict 1 ("yes") d' = .58

Exercise 2.2: Exploring minimal FFTs (using the spreadsheet "CMCv1_1cue")

The dataset "data_CMCv1_1cue" contains the data of a *village (village1)*, i.e., a random sample of n = 400 women of the full CMC data set.
Your task is to use this dataset to explore the predictive validity of individual cues by building minimal FFTs in an analogous fashion as demonstrated above. The goal of this exercise is to identify two or three cues that you think you'd like to combine when building a complex (multi-cue) FFT. Keep in mind that prediction accuracy will be measured in terms of p(hit)–p(FA).

Some explorations and solutions

Using cue *age* (*column O*):

if *age* < 20 (split 19.5), predict 1 ("yes")	*d'* = .04.	p(h)-p(FA) = .002
if *age* < 30, predict 1	*d'* = ___	p(h)-p(FA) = ___
if *age* < 35, predict 1	*d'* = ___	p(h)-p(FA) = ___
if *age* < 40, predict 1	*d'* = ___	p(h)-p(FA) = ___
if *age* < 43, predict 1	*d'* = ___	p(h)-p(FA) = ___
if *age* < 44, predict 1	*d'* = ___	p(h)-p(FA) = ___
if *age* < 45, predict 1	*d'* = ___	p(h)-p(FA) = ___

Using cue *education* (*column P*):

if *edu* > 1 (split 1.5), predict 1 ("yes")	*d'* = .74	p(h)-p(FA) = .127
if *edu* > 2, predict 1	*d'* = ___	p(h)-p(FA) = ___
if *edu* > 3, predict 1	*d'* = ___	p(h)-p(FA) = ___

Using cue *number of children* (*column R*):

if *n_child* > 0 (split 0.5), predict 1 ("yes")	*d'* = ___	p(h)-p(FA) = ___
if *n_child* > 1, predict 1	*d'* = .86	p(h)-p(FA) = .263
if *n_child* > 2, predict 1	*d'* = ___	p(h)-p(FA) = ___
if *n_child* > 3, predict 1	*d'* = ___	p(h)-p(FA) = ___
if *n_child* > 5, predict 1	*d'* = ___	p(h)-p(FA) = ___

Using cue *religion* (*column S*):

if *religion* < 0.5 (= non-Islam), predict ___	*d'* = ___	p(h)-p(FA) = ___

Using cue *working* (*column T*):

if *work* > 0.5, predict ___	*d'* = ___	p(h)-p(FA) = ___

Using cue *standard of living* (*column U*):

if *st_o_lv* > 1 (split 1.5), predict 1	*d'* = ___	p(h)-p(FA) = ___
if *st_o_lv* > 2, predict 1	*d'* = ___	p(h)-p(FA) = ___
if *st_o_lv* > 3, predict 1	*d'* = .27	p(h)-p(FA) = .106

Using cue *media exposure* (*column V*):

if *med_exp* > 0.5, predict ___	*d'* = ___	p(h)-p(FA) = ___

3 – Constructing complex FFTs

Exercise 3.1: Constructing complex FFTs (using the dataset "data_CMC_villages_FFT" and select the training sample).

It contains the data of *village1*, i.e., the same random sample of n = 400 women of the full CMC data set as the previous exercise. When constructing FFTs that contain a *sequence* of cues, we call those *multi-cue* or *complex FFTs*.

Your task is to use your knowledge about the predictive validity of individual cues (acquired in the previous exercise) to construct a FFT that is both efficient and effective, i.e., makes accurate predictions in as few decision steps as possible.

A decent 3-cue solution uses the following cues and cue settings:

FFT			
Cues & settings: IF:	n_child	age	edu
direction:	>	<	>
split value:	3.5	34.5	2.5
THEN: prediction:	1 ("yes")	1 ("yes")	1 ("yes")
Prediction counts:			
prediction (exit):	151	187	46
else (continue):	249	62	16

and yields the following result:

Summary:

mean N of steps (frugality):	
(3 cues)	**1.78**

Criterion:	**Prediction** Yes =1	No =0	
Yes =1	238	0	238
No =0	146	16	162
	384	16	**400**
	p(hit) = **.999**		p(hit)–p(FA) = **0.10**

d' =
1.735

Reducing false alarms. A potential problem with this FFT is that it predicts a positive criterion value (of 1) for nearly all cases (384 out of 400, i.e., 96%). Making almost exclusively positive predictions yields a high numbers of hits, but also leads to many false alarms. How can we reduce the number of false alarms? A simple solution is to flip the direction of one or more tests from "If *cue* > *x*, predict 1" to "If *cue* < *x*+1 predict 0". Semantically, it seems the same to say "If *age* > *20*, predict *drinks alcohol*" and "If *age* < *21*, predict *does not drink alcohol*". Nevertheless, the consequences of flipping the direction of a cue on a FFT can be profound, as this effectively means that the cases that are classified by this cue (i.e., reach the cue's exit) will no longer encounter any of the subsequent cues. Try this by changing the test of the first cue in the above FFT from "If *n_child* > 3 (split 3.5), predict 1 ("yes")" to "If *n_child* < 4 (3.5), predict 0 ("no")":

FFT			
Cues & settings: IF:	n_child	age	edu
direction:	<	<	>
split value:	3.5	34.5	2.5
THEN: prediction:	0 ("no")	1 ("yes")	1 ("yes")

Prediction counts:

prediction (exit):	249	60	51
else (continue):	151	91	40

The resulting FFT yields the following results:

Summary:

mean N of steps (frugality):	
(3 cues)	**1.61**

Criterion:	**Prediction** Yes =1	No =0	
Yes =1	87	151	238
No =0	24	138	162
	111	289	**400**
	p(hit) = **0.37**		p(hit)–p(FA) = **0.22**

d' =
0.70

Thus, we successfully achieved our goal of reducing the number of false alarms, at the cost of increasing the number of misses. Note the mixed effects of this change on classification accuracy: The measure *d'* has decreased, but p(hit)–p(FA) has increased.

Reducing misses. A weakness of this FFT is the high number of 151 *misses*, which is a direct consequence of our flipped prediction for the first cue (where 249 of the 400 cases are now classified as 0). To reduce this number, we can incrementally reduce the split value for the *n_child* cue from 3.5 to 2.5, 1.5, 0.5. Please try this and observe the effects on the predictions made by the first cue, the 2 x 2 contingency table and all summary measures. For instance, the test "If *n_child* < 1 (split 0.5), predict 0 ("no")" only classifies 24 cases on the first cue and yields the following solution:

FFT			
Cues & settings: IF:	n_child	age	Edu
direction:	<	<	>
split value:	0.5	34.5	2.5
THEN: prediction:	0 (“no”)	1 (“yes”)	1 (“yes”)

Prediction counts:

prediction (exit):	24	226	96
else (continue):	376	150	54

Summary:

mean N of steps (frugality):	
(3 cues)	**2.32**

Criterion:	**Prediction** Yes =1	No =0	
Yes =1	226	12	238
No =0	96	66	162
	322	78	**400**
	p(hit) = **0.95**	p(FA) = **0.59**	p(hit)–p(FA) = **0.36**

d' =
1.41

FFT Construction Contest. Try constructing the best FFT you can by using your knowledge about the predictive validity of individual cues—as acquired in Exercise 2.2—and sequentially checking multiple cues.

A good FFT is both efficient and effective, i.e., makes accurate predictions in as few decision steps as possible. However, to determine a winner, we will use prediction accuracy (as indicated by p(hit)–p(FA)) as the criterion of this contest.

Exercise 4: Predicting in an additional data sets (generalization, cross validation)

In all our explorations so far, we have been selecting cues and adjusting cue settings to *fit* minimal and complex FFTs to a given data set. However, the real goal of constructing a FFT is to *predict* the criterion value of cases that we have not encountered before.

To test how well our previous solutions generalize to another sample, click the button labeled testing. This will run your tree on a second village, a random sample of 400 cases that are non-overlapping with the training data used in Exercises 3.

To evaluate the same FFT across two samples do the following:

1) Assess the predictive validity (capacity to generalize) of the classification tree you developed in Exercise 3 by comparing outcome measures on the testing sample of 400 people to the outcomes on the previous training sample. How well did it do on the testing sample, relative to its capacity to fit the training sample?

References

Green, L., & Mehr, D. R. (1997). What alters physicians' decisions to admit to the coronary care unit? *Journal of Family Practice, 45*, 219–226.

Jablonskis, Mihan, Czienskowski (2015). WebFFTBuilder [Computer Software]. Url: http://www.dotwebresearch.net/divapps/fftree/; last access: 07/June/2015

Luan, S., Schooler, L. J., & Gigerenzer, G. (2011). A signal-detection analysis of fast-and-frugal trees. *Psychological Review, 118*, 316–338.

Martignon, L., Vitouch, O., Takezawa, M., & Forster, M. (2003). Naïve and yet enlightened: From natural frequencies to fast and frugal decision trees. In D. Hardman, & L. Macchi (Eds.), *Thinking: Psychological perspectives on reasoning, judgment, and decision making* (pp. 189–211). Chichester: John Wiley and Sons.

Martignon, L., Katsikopoulos, K. V., and Woike, J. K. (2008). Categorization with limited resources: A family of simple heuristics. *Journal of Mathematical Psychology, 52*, 352–361.