`LIBTVA` — User Guide

Mads Dyrholm
Center for Visual Cognition
University of Copenhagen
Denmark

June 28, 2012

Contents

Chapter 1

Introduction

1.1 About this document

Sufficient knowledge of TVA is assumed as we will not go into describing TVA as such. Please note that this document is under heavy construction, *suggestions are welcome!*

1.2 System requirements

We believe the following systems are supported:

1. Matlab 2009b or later 32/64bit on Windows XP 32/64bit.
2. Matlab 2010b or later 32/64bit on Ubuntu Linux 32/64bit.
3. Matlab 2010a or later 32/64bit on Ubuntu Linux 32/64bit.
4. Matlab 2010a or later 32/64bit on Mac OS X 32/64bit.

1.3 Obtaining and installing the software

1. Obtain the latest `LIBTVA` software by visiting `http://zappa.psy.ku.dk/libtva` and downloading the appropriate `LIBTVA` `zip`-file.
2. Unzip the `LIBTVA` `zip`-file to extract the `libtva` folder.
3. Make sure that the path to the extracted folder gets added to the Matlab search path. This can be done by clicking the File-menu in the Matlab window, then clicking “Set Path...”, then clicking “Add folder...”, and then selecting the `libtva` folder which was extracted from the `zip`-file.
4. Test the installation by running `tvaversion` like this

   ```
   >> tvaversion
   ```

   If the toolbox is correctly installed then you should see a message similar to the following

   ```
   LIBTVA version: pre-Release.
   Engine status: Engine is sane.
   ```

   Note that if you get the above message, but with an engine which is *NOT* sane, then your platform might not be supported by the version you have installed. This means that some of the toolbox functions which access the low level `LIBTVA` engine will not run.

Chapter 2

Tutorial

2.1 Your first fit with `LIBTVA`

2.1.1 Loading your data

`LIBTVA` is backwards compatible with the dataformat used with the WinTVAfit software by Søren Kyllingsbæk [2], but note however that *the old dataformat does not contain information about whether a trial is masked or unmasked.* Assuming that your tutorial datafile is found in 'datafile.dat' you may load the data into a Matlab variable by running the following command

```
>> tvadata = tvaloader('datafile.dat');
```

If the datafile is in a different directory, then the path to the datafile must be included in the command above. If the command is successful, you will see a message similar to

```
Found 5 columns in the file.
```

and you can then proceed to the next step with the data now loaded in the Matlab variable `tvadata`.

2.1.2 Inspecting your data

To inspect the data, run for example `TVAREPORT` like this

```
>> tvareport(tvadata);
```

This will print a table of the stimulus exposure durations per conditions among other useful information.

2.1.3 Performing a simple fit

To do a simple fit, run `TVAFIT` on the data

```
>> [theta,tvamodel,tvadata] = tvafit(tvadata);
```

This will fit using some defaults.

2.1.4 Inspecting your simple fit

See the fit and errorbars with

```
>> tvareport(tvadata,tvamodel,theta);
```

2.2 Mixing masked and un-masked trials

2.2.1 Method one

The `TVALOADER` has been equipped with a fix for un-masked conditions. Use

```
>> tvadata = tvaloader('filename','STD', uc );
```

where `uc` is a list of condition numbers that are to be assumed un-masked, for example if conditions 5 and 6 are to be assumed un-masked then the command would be as follows

```
>> tvadata = tvaloader('filename','STD', [5 6] );
```

Note that trials are assumed masked unless otherwise defined.

2.2.2 Method two

Alternatively, to define un-masked conditions after the data has been loaded use `TVAPATCHER` like this

```
>> tvadata = tvapatcher(tvadata,{[5 6],'unmask',1});
```

where [5 6] is to be replaced by a list of condition numbers that are to be assumed un-masked.

2.2.3 Proceeding to fit

After you have done either of the above methods you can proceed to fit exactly as before.

2.3 Fitting multiple files and exporting group spreadsheet

Put the following in a text-file (.m) and adjust according to your needs

2.3.1 fitmygroup.m

```
%% Where is the data, and what files to fit...
datpath = '/home/mads/professor/Copenhagen/data/behavioral/randi/level2'
datfiles = {'Singlet204out.dat',
    'Singlet205out.dat'};

%% Put the condition numbers of un-masked trials.
unmaskedcond = [ ];  % set f.eks. to [5 6]

% This is used to name output files (see XX in the comments below).
uddyb = 'fit1000';

%% prepare variables for the following call to tvafit
%% [theta,tvamodel,tvadata,df,bic] = tvafit(D,L,M,C,P,S,A)
L = 1; % number of v values if applicaple, or number of w values is applicaple.
M = 'FREE'; % K-model. Could be 'TRAD' or 'FREE' or 'BINOSTACK' (unpublished)
C = 'EXP'; % Curve. Could be 'EXP' or 'EX-GAUSSIAN'.
P = 0; % Estimate lapses? - P=0 means no, P=1 means yes.
S = 0; % If applicaple (not if v above), estimate C or s? - 0 means C, 1 means 1 s, 2 means 2 s etc.
A = 1; % If applicable (distractors?), A=1 means one alpha for all, A=2 means individual alpha.

if ~isempty(unmaskedcond) && strcmp(C,'EX-GAUSSIAN')
  error('EX-Gaussian curve is not supported when unmasked trials. (YET!)');
end

%% The script will do the rest:
%%    Each fit will be saved in the data path as datafile_XX.mat
%%    And, finally a spreadsheet named report_XX.txt will be written to the datpath.
```

```
for subj = 1:length(datfiles)
  datfil = datfiles{subj};
  filename = fullfile(datpath,datfil);
  [tvadata,columns] = tvaloader(filename,'STD', unmaskedcond);
  [theta,tvamodel,tvadata,df,bic] = tvafit(tvadata,L,M,C,P,S,A);
  [dummy,filepre,fileext] = fileparts(datfil);
  tail = ['_' uddyb '.mat'];
  fitfilename = fullfile(datpath,[filepre tail]);
  save(fitfilename,'theta','tvamodel','tvadata','df','bic');
end

reportfilename =  fullfile(datpath,['report_' uddyb '.txt']);
fprintf('Rapporten bliver gemt i %s\n',reportfilename);

tvaexport(reportfilename,'DIR',fullfile(datpath,['*' tail]));
```

2.3.2 Group analysis with R, SPSS, Excel etc...

The written file 'report_XXX.txt' can be imported into your favorite statistics program.

2.3.3 Interpretation of the output

Error-bars

The software computes parametric error-bars using a Gaussian approximation to the multivariate parameter pdf. Parameters K and σ_0 are excluded from the computation of error-bars, and any correlations with these parameters are therefore neglected.

Chapter 3

Known issues

- Fitting with the ex-Gaussian engine does not support un-masked trials yet. The Hessian entry for (μ, u_0) needs to be derived and implemented first. So, if you have un-masked trials then use only the exponential curve.
- un-masked trial indicator needs to be incorporated into the data file format so that the user does not have to specify un-masked conditions when using the toolbox.
- Supported systems may vary.

Chapter 4

The `LIBTVA` Matlab function reference

This chapter contains an extract of the help text which is provided with each Matlab function of the `LIBTVA` toolbox. The information here is thus equivalent to typing

```
>> help XXXXXX
```

where `XXXXXX` can be the name of a Matlab function. Not all functions have very elaborate help text, but this will be expanded over time. Again, suggestions are welcome!

4.1 `tvabic`

```
 TVABIC

  Synopsis
  ========

  bpp = tvabic(data)
  bpz = tvabic(data,dim)
  bic = tvabic(data,dim,ll)

  -- Author: Mads Dyrholm --
     Center for Visual Cognition, University of Copenhagen.
     2009 - September 2010

  Purpose
  =======

  Compute the Swartz criterion/Bayes information criterion
  to account for parameter count.

  Inputs
  ======

  data - Scalar or TVADATA struct. The criterion depends on the
  number of observations: A scalar can be given to directly give
  the number of observations; or if TVADATA struct is given, then
  the length is taken as the number of trials.

  dim - Number of parameters to account for.

  ll - The log likelihood (see TVACOST or TVADIAG).

  Outputs
  =======
  bpp - Penalization per parameter.

  bpz - Penalization, i.e. dim*bpp.

  bic - The penalized log likelihood, i.e. ll minus bpz.
```

4.2 tvaboot

```
TVABOOT

 Synopsis
 ========

 BOOTSTATS = tvaboot(N,tvadata,varargin)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    January 2011

 Purpose
 =======

 Bootstrap TVAFIT.

 Inputs
 ======

 N - Number of bootstrap samples.

 tvadata - Data, same input as to TVAFIT.

 varargin - Same inputs as to TVAFIT.

 Outputs
 =======

 BOOTSTATS - Each row of this matrix is the bootstrap
 sample estimates according to TVAFITVEC.
```

4.3 tvacheckdatafile

```
TVACHECKDATAFILE

 Synopsis
 ========

 status = tvacheckdatafile(datafile)

 -- Author: Rasmus Berg Palm --
    Center for Visual Cognition, University of Copenhagen.
    October 2011

 Purpose
 =======

 Checks if datafile is sane. Throws exception if not.

 Inputs
 ======

 datafile - A string containing the data filename.

 Outputs
 =======

 status - 'OK' if no errors found.
```

4.4 tvacheckgrad

```
TVACHECKGRAD

 Synopsis
 ========

 [maxJ, err, index] = tvacheckgrad(theta,tvamodel,tvadata)
 [maxJ, err, index] = tvacheckgrad(theta,tvamodel,tvadata,theta_fix)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    April 2010

 Purpose
 =======

 Useful for numerical checking of gradient computations. This
 is intended for debugging purposes.
```

4.5 tvaconditions

```
TVACONDITIONS

 Synopsis
 ========

 1. for PR and WR data

 [cc,tt,nn,errors,intrusions,   ...
  reported,correported,targets, ...
  distractors,scorepdf,maxtargets] = tvaconditions(tvadata)

 2. for CD data

 [cc,tt,nn,tp,fp,tn,fn,dk] = tvaconditions(tvadata);

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - May 2010
 Purpose
 =======

 Summarize stimulus and subject behavior and
 performance per condition.

 Inputs
 ======

 tvadata - Dataset (see also TVALOADER).
```

4.6 tvacost

```
TVACOST

 Synopsis
 ========

 [nll,g,H] = tvacost(theta,theta_fix,tvamodel,tvadata)
 [nll,g,H] = tvacost(theta,theta_fix,tvamodel,tvadata,vcf)
 [nll,g,H] = tvacost(theta,theta_fix,tvamodel,tvadata,vcf,shutup)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - July 2011

 Purpose
 =======

 Compute the cost function, gradient, and Hessian,
 for optimization purposes.

 Inputs
 ======

 theta - Parameter vector. Note: TVACOST only returns gradient
 and Hessian for what is in theta, hence, theta should not
 be stripped if one is optimizing guessing constants.
 (see also TVASTRIPTHETA, TVAFLEXCHAIN)

 theta_fix - Fixed parameters, can be []. (see also TVAFIXER)

 tvamodel - Model struct. (see also TVAINIT)

 tvadata - Dataset. (see also TVALOADER)

 vcf - (Optional) This must be set to 1 when TVACOST is used
 for optimization (default). When set to 0, the gradient
 and Hessian outputs are derived in the human domain.

 shutup - (Optional) Set this to 1 to disable the default
 behavior of writing the cost function to the screen.

 Outputs
```

```
=======

nll - Negative log likelihood.

g - Gradient vector i.e. d_nll/d_theta.

H - Hessian matrix of second order derivatives d^2_nll/d_theta^2
```

4.7 tvadatacat

```
TVADATACAT

 Synopsis
 ========

 [tvadata,pshift,cshift] = tvadatacat(tvadata1,tvadata2)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    November 2011

 Purpose
 =======

 Concatenate two TVA datasets.

 Inputs
 ======

 Outputs
 =======
```

4.8 tvadeal

```
TVADEAL

 Synopsis
 ========

 [alpha,w,C,s,v,u0,chdetgm,mu] = tvadeal(tvamodel,theta)
 [alpha,w,C,s,v,u0,chdetgm,mu] = tvadeal(tvamodel,theta,theta_fix)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - May 2010

 Purpose
 =======

 Deal the elements of theta and theta_fix as according to
 the tvamodel.

 Inputs
 ======

 tvamodel - Model struct. (see also TVAINIT)

 theta - Parameter vector. (see also TVAFLEXCHAIN)

 theta_fix - Fixed parameters, can be []. (see also TVAFIXER)
```

4.9 tvadiag

```
TVADIAG

 Synopsis
 ========

 [humtheta,sigma,ll,pM, ...
  p,normg,H,g,U,rcondU] = tvadiag(theta,tvamodel,tvadata)
 [humtheta,sigma,ll,pM, ...
  p,normg,H,g,U,rcondU] = tvadiag(theta,tvamodel,tvadata,theta_fix)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - February 2011

 Purpose
 =======

 Compute human domain error bars and other diagnostics.

 Inputs
 ======

 theta - Parameter vector. (see also TVAFLEXCHAIN)

 tvamodel - Model struct. (see also TVAINIT)

 tvadata - Dataset. (see also TVALOADER)

 theta_fix - (Optional) Fixed parameters. Can be [] for
 no fixed parameter. (see also TVAFIXER)

 Outputs
 =======

 humtheta - Human readable theta combined with theta_fix.
 (see also TVAHUMAN)

 sigma - Error bars for humtheta. Elements corresponding
 to theta_fix will be set to NaN.

 ll - Log likelihood.

 pM - Bayes model posterior, estimated via Laplace
 integration approximation.
 p - Flag which is 0 whenever Hessian is positive definite.
 Note! When p>0 then H is not positive definite and the
 error bars and model posterior can not be trusted.

 normg - The inf-norm of the gradient vector in optimizer
 domain. This should be small in combination with p=0 to
 indicate the the model is fitted to an optimum.

 H - Hessian matrix, in human domain.

 g - gradient vector, in open/optimizer domain.

 U - The Cholesky factor of H.

 rcondU - Reciprocal 1-norm condition number of U.
```

4.10 tvaexport

```
TVAEXPORT

 Synopsis
 ========

 tvaexport(filename,op,par1,par2,...,op,...)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - November 2010

 Purpose
 =======

 Export fits to text file suitable for importing into
 statistical software such as R, SPSS, etc.
 The exported file comes with a header, followed by a
 line per fit.

 Inputs
 ======

 filename - The file to create/overwrite. TVAEXPORT Will
 prompt if the file already exists.

 op - String indicating which option the following arguments
 will pertain to. Options are divided in two groups:

  Group 1. How to pass the data and fits to TVAEXPORT.

   op     | Description
  --------+----------------------------------------
   'DIR'  | par1 is the dir argument and the export
          | will be performed on the files that
          | match the dir argument. Each file must
          | have a tvadata, tvamodel, theta,
         | and optionally theta_fix variable.
   'GIVE' | Give all the fits as five cell arrays of
         | identical dimension: par1 is a cell
         | array with par1{#} being TVADATA for
         | fit no. #. Similarly, par2 holds TVAMODEL,
         | par3 holds THETA, par4 holds the id
         | strings, and par5 holds THETA_FIX.
         | As an option, if par5 is empty then
         | theta_fix is assumed [].

  Group 2. What to put in the exported report.

   op     | Description
  --------+--------------------------------------
  'STL'   | Single-trial info will be exported.
  'LPR'   | Fitted parameters will be exported.

  The default for Group 2 is LPR.
```

4.11 tvaexportsingletrial

```
TVAEXPORTSINGLETRIAL

 Synopsis
 ========

 tvaexportsingletrial(filename,id,tvadata,tvamodel,theta)
 tvaexportsingletrial(filename,id,tvadata,tvamodel,theta,theta_fix)

 [1] Dyrholm, M., et al. (2011) "Single Trial Inference on Visual
 Attention", AIP Conference Proceedings, Volume 1371, pp. 37-43

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    December 2010

 Purpose
 =======

 Export single trial variables.
```

4.12 tvafit

```
TVAFIT

 Synopsis
 ========

  [theta,tvamodel,tvadata,df,bic] = tvafit(D)
  [theta,tvamodel,tvadata,df,bic] = tvafit(D,L)
  [theta,tvamodel,tvadata,df,bic] = tvafit(D,L,M)
  [theta,tvamodel,tvadata,df,bic] = tvafit(D,L,M,C)
  [theta,tvamodel,tvadata,df,bic] = tvafit(D,L,M,C,P)
  [theta,tvamodel,tvadata,df,bic] = tvafit(D,L,M,C,P,S)
```

```
[theta,tvamodel,tvadata,df,bic] = tvafit(D,L,M,C,P,S,A)

-- Author: Mads Dyrholm --
   Center for Visual Cognition, University of Copenhagen.
   2009 - May 2011

Purpose
=======

TVA fitting made easy.

Inputs
======

(Note that any of the inputs can be set to [] for
default value)

D - a FILENAME string or TVADATA struct. If a string is
given, then the file will be loaded using TVALOADER.

L - Scalar or matrix. Either the number of hazard rates,
the number of attentional weights, or the inflator matrix.
Default value is 1. See also TVALINK.

M - String indicating the K model. Possible values are
'FREE','BINOSTACK','TRAD'. Default value is 'FREE'.

C - String indicating the shape of the curve. Possible
values are 'EXP' and 'EX-GAUSSIAN'. Default value is
'EXP'.

P - Scalar, flag whether lapse probability should be
estimated. Set to 1 for yes. Default value is 0.

S - Scalar, indicating split processing capacity. Set
equal to the number of s values to estimate. Default value
is 0 which means that C is estimated instead of s.

A - Scalar, flag whether alpha is a scalar or whether
distractors get their own weight. Set to 1 for scalar
alpha, set different to enable individual distractor
weights. Default value is 1.

Outputs
=======

df - Degrees of freedom used.

bic - Bayes Information Criterion, see also TVABIC.

tvamodel.M - Set to M.

tvamodel.Curve - Set to C.
```

4.13 tvafitvec

```
TVAFITVEC

 Synopsis
 ========

 v = tvafitvec(varargin)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    January 2011

 Purpose
 =======

 Call TVAFIT and prepare vector for resampling.

 Inputs
 ======

 varargin - Same inputs as TVAFIT.

 Outputs
 =======

 v - Vector of parameters, [theta; tvamodel.K; tvamodel.s0].
```

4.14 tvafixer

```
TVAFIXER

 Synopsis
 ========

 1.                 v = tvafixer(s,x)
 2. [theta,theta_fix] = tvafixer(theta,theta_fix,idx,tvamodel)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - April 2010

 Purpose
 =======

 Useful for fixing individual parameters:

 1. Transform a value 'x' to so that it can be put in theta
 or theta_fix manually.
 2. Fix a set of parameters in theta. The fixed parameters
 are moved from theta to theta_fix.

 Inputs
 ======

 s - A character, can be '+' or 'u'. Set it to chose
 the transformation of x to v.

 x - A number to be transformed. Can be a vector, then all
 elements will be subject to the same transformation.

 theta - Parameter vector.

 theta_fix - Vector of fixed parameters.

 idx - Parameter indices to fix. The indices are relative
 to the combination of theta and theta_fix. That is, if
 theta_fix is [] then idx is relative to theta. Otherwise
 idx is relative to theta_fix.

 tvamodel - Model struct. (see also TVAINIT)

 Outputs
 =======

 v - Transformed version of x. If s is '+' then the
 exponential function is used. If s is 'u' then the
 logistic function is used.

 theta - New theta, with fewer parameters than on input.

 theta_fix - New theta_fix containing the fixed parameters.
```

4.15 tvaflexchain

```
TVAFLEXCHAIN

 Synopsis
 ========

 [PE,g,H]  = tvaflexchain(theta,tvamodel,tvatrial,logistidx,expidx)
 [PE,g,H]  = tvaflexchain(theta,tvamodel,tvatrial,logistidx,expidx,vcf)
 [v,taumu,w] = tvaflexchain(theta,tvamodel,tvatrial,logistidx,expidx,2)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - September 2011

 Purpose
 =======

 Compute trial response probability, the gradient, and the
 Hessian, for various factorizations of the TVA model.

 Inputs
 ======

 theta - Vector of parameters. The elements and order depends
 on the '.facstr' field of tvamodel. The tables below gives
 the relationship between valid characters of facstr and theta.
 The elements must live in open domain on input (see TVAFIXER).
```

```
Theta must be stripped from guessing constants on input
(see TVASTRIPTHETA).

  character | Corresponding TVA parameter.
  ----------+----------------------------------------
   'a'      | Efficiency of selection, alpha.
   'v'      | Hazard rate.
   'w'      | Attentional weight.
   'C'      | Processing capacity.
   's'      | Split processing capacity.
   'u'      | u0 or t0. This must always be present.
   'm'      | Mu.

With these characters the user can define which factorisation
of the TVA model to use. The following table gives examples
of various facstr and their corresponding theta.

  facstr    | theta
  ----------+----------------------------------------
   'vu'     | [v values; t0]
   'wCu'    | [w values without w_1; C; t0]
   'awCu'   | [alpha; w values without w_1; C; t0]
   'awsu'   | [alpha; w values without w_1; s values; t0]
   'awCum'  | [alpha; w values without w_1; C; t0; mu]
   'awsum'  | [alpha; w values without w_1; s values; t0; mu]

Note that the order of the elements of theta is independent
of the order of the characters of facstr. The order of the
elements of theta following the first table above. Note
that facstr must always contain a 'u'.

tvatrial - A single trial, i.e. a single cell from a
TVADATA array. (see also TVALOADER)

vcf - Flag whether you want the output g and H to represent
theta in open domain or in human domain. Set to 1 for open
domain, set to 0 for human domain. Default is 1.

Outputs
=======

PE - Trial response probability.

g - Gradient array. The element order follows the
ordering of theta.

H - Hessian matrix. The element order follows the
ordering of theta.
```

4.16 tvagetopts

```
TVAGETOPTS

 Synopsis
 ========
 opts = tvagetopts('datafile 1', 'datafile 2', ...)

 -- Author: Rasmus Berg Palm --
    Center for Visual Cognition, University of Copenhagen.
    May 2011

 Purpose
 =======

 Returns intersect of possible options for datafiles to be used in TVAFIT. If only one datafile is sup

 Inputs
 ======

 datafile(s) - string filename(s)

 Outputs
 =======

 opts - struct with possible options for TVAFIT. The first element of the options are the default valu

 Example
 =======
 opts = tvagetopts('betula.dat', 'betula_bad.dat')

 opts =

    mode: 'attentional weights'
       L: 1
       M: {'FREE'  'TRAD'  'BINOSTACK'}
       C: {'EXP'  'EX-GAUSSIAN'}
       P: {[0]  [1]}
       S: [0 1]
       A: 1
```

4.17 tvaheat

```
TVAHEAT

 Synopsis
 ========

 [theta,tvamodel] = tvaheat(theta,tvamodel,tvadata,s0list)
 [theta,tvamodel] = tvaheat(theta,tvamodel,tvadata,s0list,theta_fix)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - May 2010

 Purpose
 =======

 Heat up the t0 model and pick the best one using log likelihood.
```

```
Inputs
======

theta - Parameter vector. (see also TVAFLEXCHAIN)

tvamodel - Model struct. (see also TVAINIT)

tvadata - Dataset. (see also TVALOADER)

s0list - A vector with the STD[t0] to test.

theta_fix - Vector of fixed parameters. (see also TVAFIXER)

Outputs
=======

theta - New fittet theta assuming ex-Gaussian processing.
(see also TVAPUTT)

tvamodel - New updated model struct with '.s0' field set to
STD[t0].
```

4.18 tvaheatb

```
TVAHEAT

  Synopsis
  ========

  [theta,tvamodel] = tvaheatb(theta,tvamodel,tvadata,s0list)
  [theta,tvamodel] = tvaheatb(theta,tvamodel,tvadata,s0list,theta_fix)

  -- Author: Mads Dyrholm --
     Center for Visual Cognition, University of Copenhagen.
     2009 - May 2010

  Purpose
  =======

  Heat up the t0 model and pick the best one using Bayes,
  see also TVADIAG.

  Inputs
  ======

  theta - Parameter vector. (see also TVAFLEXCHAIN)

  tvamodel - Model struct. (see also TVAINIT)

  tvadata - Dataset. (see also TVALOADER)

  s0list - A vector with the STD[t0] to test.

  theta_fix - Vector of fixed parameters. (see also TVAFIXER)
```

```
Outputs
=======

theta - New fittet theta assuming ex-Gaussian processing.
(see also TVAPUTT)

tvamodel - New updated model struct with '.s0' field set to
STD[t0].
```

4.19 tvaheater

```
TVAHEATER

 Synopsis
 ========

 [theta,tvamodel] = tvaheater(theta,tvamodel,tvadata)
 [theta,tvamodel] = tvaheater(theta,tvamodel,tvadata,theta_fix)
 [theta,tvamodel] = tvaheater(theta,tvamodel,tvadata,theta_fix,crit)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - June 2010

 Purpose
 =======

 Estimate sigma0, with option to switch criterion.

 Inputs
 ======

 crit - Integer, 1:loglik, 2:bayes. Default is 1.
```

4.20 tvahuman

```
TVAHUMAN

 Synopsis
 ========

 [humtheta,logistidx,expidx] = tvahuman(theta,tvamodel)
 [humtheta,logistidx,expidx] = tvahuman(theta,logistidx,expidx)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - July 2011

 Purpose
 =======

 Convert to human readable. See TVAFIXER for the inverse.

 Inputs
 ======

 theta - Parameter vector. (see also TVAFLEXCHAIN)

 tvamodel - Model struct. (see also TVAINIT)

 logistidx - Indices of theta that are to be consideret
 subject to a unitary constraint.

 expidx - Indices of theta that are to be consideret
 subject to a positive constraint.

 Outputs
 =======

 humtheta - Human readable version of theta.

 logistidx - Indices of theta that are to be consideret
 subject to a unitary constraint.

 expidx - Indices of theta that are to be consideret
 subject to a positive constraint.
```

4.21 tvainit

```
TVAINIT

 Synopsis
 ========

 [theta,tvamodel] = tvainit(tvadata)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - Aug 2011

 Purpose
 =======

 Ballpark TVA parameters and model configuration to
 get a quick and dirty fit to the data. To do more precise
 fitting one should use a more elaborate function such as
 TVAFIT.

 Inputs
 ======

 tvadata - Dataset. (see also TVALOADER)

 Outputs
 =======

 theta - Parameter vector. (see also TVAFLEXCHAIN)

 tvamodel - Model struct which can be used for fitting
```

```
with given dataset. (see also TVASCULPT)
```

4.22 tvaisalphaexpanded

```
TVAISALPHAEXPANDED

 Synopsis
 ========

 b = tvaisalphaexpanded(tvamodel)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    July 2010

 Purpose
 =======

 Check whether alpha is position dependent or not.
```

4.23 tvaiscddata

```
TVAISCDDATA

 Synopsis
 ========

 b = tvaiscddata(tvadata)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - May 2010

 Purpose
 =======

 Check for Change Detection trials in the dataset.
```

4.24 tvaisprdata

```
TVAISPRDATA

 Synopsis
 ========

 b = tvaisprdata(tvadata)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - May 2010

 Purpose
 =======

 Check for Partial Report trials in the dataset.
```

4.25 tvaiswrdata

```
TVAISWRDATA

 Synopsis
 ========

 b = tvaiswrdata(tvadata)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - May 2010

 Purpose
 =======

 Check for Whole Report trials in the dataset.
```

4.26 tvak

```
TVAK

 Synopsis
 ========

 K = tvak(tvamodel)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - May 2010

 Purpose
 =======

 Compute K as E[K|K>0].

 Outputs
 =======

 K - Estimated VSTM capacity. The K=0 probability
 is not included in the computation.
```

4.27 tvalapse

```
TVALAPSE

 Synopsis
 ========
 [tvamodel,theta] = tvalapse(tvamodel,theta,tvadata)
 [tvamodel,theta] = tvalapse(tvamodel,theta,tvadata,theta_fix)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - May 2010

 Purpose
 =======

 Estimate the probability of attentional lapse.

 Outputs
 =======

 tvamodel - The K=0 probability of the model will
 be updated to represent the lapse probability.
```

4.28 tvalink

```
TVALINK

 Synopsis
 ========

 1. b = tvalink(all)
 2. L = tvalink(all,some)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - May 2010

 Purpose
 =======

 Set up inflator matrices for linking elements to TVA parameters.

 Inputs
 ======

 all - integer, how many positions.

 some - integer, how many parameters.

 Outputs
 =======

 1. b - A vector of possible 'some' argument.

 2. L - inflator matrix.
```

4.29 tvalinkbuilder

```
TVALINKBUILDER

 Synopsis
 ========

 L = tvalinkbuilder(L1,L2,L3,...)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    June 2011

 Purpose
 =======

 Set up inflator matrices for linking positions to TVA parameters.

 Inputs
 ======

 Ln -

 Outputs
 =======

 L - inflator matrix.
```

4.30 tvaloader

```
TVALOADER

 Synopsis
 ========

 [tvadata,columns] = tvaloader(datafile)
 [tvadata,columns] = tvaloader(datafile,what)
 [tvadata,columns] = tvaloader(datafile,what,opts)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - November 2011

 Purpose
 =======

 Load data from file.

 Inputs
 ======

 datafile - A string containing the data filename.

 what - A string, optional, not case sensitive,
 possible values: 'STD','CDTAS'. Default is STD.
```

```
  opts - Depends on 'what' as following table

   what  | Description of opts
  -------+-----------------------------------------
   'STD' | Vector with un-masked condition numbers.

  Outputs
  =======

  tvadata - struct
```

4.31 tvalogistic

```
 TVALOGISTIC

  Synopsis
  ========

  px = tvalogistic(x)

  -- Author: Mads Dyrholm --
     Center for Visual Cognition, University of Copenhagen.
     2009 - June 2010

  Purpose
  =======

  Compute the logistic function.
```

4.32 tvalpr

```
 TVALPR

  Synopsis
  ========

  tvalpr(filename,id,tvadata,tvamodel,theta)
  tvalpr(filename,id,tvadata,tvamodel,theta,theta_fix)

  -- Author: Mads Dyrholm --
     Center for Visual Cognition, University of Copenhagen.
     2010 - February 2011

  Purpose
  =======

  The TVA line printer.

  Inputs
  ======

  filename - A string with the name of the file to write to.
  If filename is empty then STDOUT will be used for output.
```

```
id - A string which will be put in the first column of
the line. If ID is empty, then a header line will be written.
```

4.33 tvalprx

```
TVALPRX

 Synopsis
 ========

 tvalprx(fid,cmd,varargin)
 tvalprx(fid,'PRED',tvadata,tvamodel,theta,theta_fix,cc,title)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    October 2010

 Purpose
 =======

 Line printer extensions.
```

4.34 tvam

```
TVAM

 Synopsis
 ========

        m = tvam([]      ,str,par1,par2,...)
 tvamodel = tvam(tvamodel,str,par1,par2,...)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - May 2010

 Purpose
 =======

 Function for setting the VSTM capacity histogram and
 lapse probability.

 Inputs
 ======

 str - A string indicating which pdf to use and
 the meaning of par1 etc. Valid strings are:

   'TRAD' - (par1)=(mu)

   'BINOSTACK' - (par1,par2,par2)=(Kalways,Kmaybe,Pmaybe). The
   mean of this is Kalways+Kmaybe*Pmaybe.

   'm' - (par1)=(m)
```

```
    'lapse' - (par1)=(Plapse)  , requires tvamodel given on input.

  Outputs
  =======

  If tvamodel is [] then the pdf is returned. Otherwise the cdf is
  returned at tvamodel.K
```

4.35 tvamcmcstep

```
 TVAMCMCSTEP

  Synopsis
  ========

  [a,lnpZ]           = tvamcmcstep(lnpX,lnpY);
  [a,lnpZ,Z1,Z2,...] = tvamcmcstep(lnpX,lnpY,X1,Y1,X2,Y2,...)
  [a,lnpZ,Z1,Z2,...] = tvamcmcstep(lnpX,lnpY,X1,Y1,X2,Y2,...)

  -- Author: Mads Dyrholm --
     Center for Visual Cognition, University of Copenhagen.
     February 2010

  Purpose
  =======

  Metropolis-Hastings step.

  Inputs
  ======

  lnpY - log of proposed probability

  Outputs
  =======

  a - accept flag. 1 if accept, 0 if reject.

  *Z* - Equals *Y* if accept, equals *X* if reject. nb: reject proposed if nan or complex
```

4.36 tvamuk

```
 TVAMUK

  Synopsis
  ========

  [muk,m,idx] = tvamuk(tvamodel)

  -- Author: Mads Dyrholm --
     Center for Visual Cognition, University of Copenhagen.
     2009 - May 2010
  Purpose
  =======

  Compute E[K], and m.

  Inputs
  ======

  tvamodel -

  Outputs
  =======

  muk - E[K]

  m - m, [Plapse k_histo]

  idx - indexes non-zero elements of m
```

4.37 tvanumlocations

```
TVANUMLOCATIONS

 Synopsis
 ========

 [N,e] = tvanumlocations(tvadata)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    July 2010

 Purpose
 =======

 Return the number of locations in the data.

 Outputs
 =======

 N - The number of locations in trial no. 1.

 e - Set to 1 if some other trial deviates from N.
 Set to 0 otherwise.
```

4.38 tvapatcher

```
TVAPATCHER

 Synopsis
 ========

 tvadata = tvapatcher(tvadata,{c1,p1,v1},{c2,p2,v2},...)
```

```
-- Author: Mads Dyrholm --
   Center for Visual Cognition, University of Copenhagen.
   2009 - April 2010

Purpose
=======

Patch data according to condition numbers.

Inputs
======

c* - Condition set.

p* - Field name to be set in tvadata.

v* - Field value to be set in tvadata.
```

4.39 tvapenc

```
TVAPENC

 Synopsis
 ========

 [PE,gv,Hvv,gt,Hvt,Htt] = tvapenc(tau,v,K,x)
 [PE,gv,Hvv,gt,Hvt,Htt] = tvapenc(tau,v,K,x,s0)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen
    2009 - January 2010

 Purpose
 =======

 This routine provides a safe wrapper to the TVACOREPENC
 routine which calls the libTVA kernel algorithm for
 computing the probability that an item is encoded into
 VSTM.

 Inputs
 ======

 tau - Expected effective stimulus duration, t-E[t0].

 v - Vector of v values.

 K - Scalar, or vector indicating the VSTM capacity CDF.
 If K is a scalar then the integer mix model is used,
 otherwise K must be the VSTM capacity CDF over
 [0,length(v)-1].

 x - Probed item scalar in [1,length(v)].

 s0 - Scalar, optional, assumed deviation on t0. Specifying this
 option s0>0.0 will enable the ex-gaussian model. If this option
 is not specified, or set to 0.0 then exponential processing
 will be used (i.e. t0=E[t0]).

 Outputs
 =======

 PE - Encoding probability.

 g* - Gradient arrays. gv is dPE/dv, gt is dPE/dE[t0].

 H** - Hessian matrices. Hvv is ddPE/dvdv. Hvt is the
 vector ddPE/dvdE[t0], Htt is the scalar ddPE/dE[t0]dE[t0].
```

4.40 tvaplot

```
TVAPLOT

 Synopsis
 ========

 [tt,oo,pp,cc] = tvaplot(tvadata,tvamodel,theta)
 [tt,oo,pp,cc] = tvaplot(tvadata,tvamodel,theta,theta_fix)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    February 2012

 Purpose
 =======

 Prepare for plotting purposes.

 Outputs
 =======

 tt - List of exposure average durations per condition.

 oo - List of observed performances, per condition.

 pp - List of predicted performances, per condition.

 cc - List of conditions.
```

4.41 tvappar

```
TVAPPAR

 Synopsis
 ========

 [PP,gv,Hvv,gt,Hvt,Htt] = tvappar(tau,v,K,R,T,s0)

 -- Author: Mads Dyrholm --
```

```
    Center for Visual Cognition, University of Copenhagen
    February 2010

 Purpose
 =======

 This routine provides a safe wrapper to the TVACOREPPAR
 routine which calls the libTVA kernel algorithm for
 computing the partial report probability for the report R.

 Inputs
 ======

 tau - Expected effective stimulus duration.

 v - Vector of v values.

 K - Scalar, or vector indicating the VSTM capacity CDF.
 If K is a scalar then the integer mix model is used,
 otherwise K must be the VSTM capacity CDF over
 [0,length(v)-1].

 R - Vector, report, each element in [1,length(v)].

 T - Vector, tagets, each element in [1,length(v)].

 s0 - Scalar, optional, assumed deviation on t0. Specifying this
 option s0>0.0 will enable the ex-gaussian model. If this option
 is not specified, or set to 0.0 then exponential processing
 will be used.

 Outputs
 =======

 PP - Report probability.

 g* - Gradient arrays. gv is dPP/dv, gt is dPP/dt0.

 H** - Hessian matrices. Hvv is ddPP/dvdv. Hvt is the
 vector ddPP/dvdt0, Htt is the scalar ddPP/dt0dt0.
```

4.42 tvaprddistwrap

```
TVAPRDDISTWRAP

 Synopsis
 ========

 [Pd,Ed] = tvaprddistwrap(tau,v,K,R,T,s0)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen
    July 2010

 Purpose
```

```
  =======

  This routine provides a safe wrapper to the TVAPARDDIST
  routine which calls the libTVA kernel algorithm for
  computing ...

  Inputs
  ======

  (see also TVAPPARWRAP)

  Outputs
  =======

  Pd(i) - The probability that (i-1) distractors were
  encoded.

  Ed - The expected number of distractors encoded.
```

4.43 tvapredict

```
 TVAPREDICT

  Synopsis
  ========

  S = tvapredict(tvadata,tvamodel,theta,theta_fix)

  -- Author: Mads Dyrholm --
     Center for Visual Cognition, University of Copenhagen
     October 2010

  [1] Dyrholm, M., et al. (2011) "Single Trial Inference on Visual
  Attention", AIP Conference Proceedings, Volume 1371, pp. 37-43

  Purpose
  =======

  Compute predictions...

  Outputs
  =======

  S{n}.PE - Individual encoding probabilities.

  S{n}.Es - Expected VSTM load.
```

4.44 tvaputt

```
 TVAPUTT

  Synopsis
  ========
  [theta,info] = tvaputt(theta,tvamodel,tvadata)
  [theta,info] = tvaputt(theta,tvamodel,tvadata,theta_fix)
  [theta,info] = tvaputt(theta,tvamodel,tvadata,theta_fix,opts)

  -- Author: Mads Dyrholm --
     Center for Visual Cognition, University of Copenhagen.
     2009 - September 2011

  Purpose
  =======

  Fit TVA parameters with fixed K and sigma0.

  Inputs
  ======

  opts - [maxit] , default [100]
```

4.45 tvapwho

```
TVAPWHO

 Synopsis
 ========

 [PW,gv,Hvv,gt,Hvt,Htt] = tvapwho(tau,v,K,R,s0)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen
    January 2010

 Purpose
 =======

 This routine provides a safe wrapper to the TVACOREPWHO
 routine which calls the libTVA kernel algorithm for
 computing the whole report probability for the report R.

 Inputs
 ======

 tau - Expected effective stimulus duration.

 v - Vector of v values.

 K - Scalar, or vector indicating the VSTM capacity CDF.
 If K is a scalar then the integer mix model is used,
 otherwise K must be the VSTM capacity CDF over
 [0,length(v)-1].

 R - Vector, report, each element in [1,length(v)].

 s0 - Scalar, optional, assumed deviation on t0. Specifying this
 option s0>0.0 will enable the ex-gaussian model. If this option
 is not specified, or set to 0.0 then exponential processing
```

```
  will be used.

  Outputs
  =======

  PW - Encoding probability.

  g* - Gradient arrays. gv is dPW/dv, gt is dPW/dt0.

  H** - Hessian matrices. Hvv is ddPW/dvdv. Hvt is the
  vector ddPW/dvdt0, Htt is the scalar ddPW/dt0dt0.
```

4.46 tvareport

```
 TVAREPORT

  Synopsis
  ========

  tvareport(tvadata)
  tvareport(tvadata,tvamodel,theta)
  tvareport(tvadata,tvamodel,theta,theta_fix)

  -- Author: Mads Dyrholm --
     Center for Visual Cognition, University of Copenhagen.
     2009 - June 2011

  Purpose
  =======

  Display a report about the data, model, and fit.
```

4.47 tvarsq

```
 TVARSQ

  Synopsis
  ========

  [Rsq,oo,pp] = tvarsq(tvadata,tvamodel,theta)
  [Rsq,oo,pp] = tvarsq(tvadata,tvamodel,theta,theta_fix)

  -- Author: Mads Dyrholm --
     Center for Visual Cognition, University of Copenhagen.
     February 2012

  Purpose
  =======

  Compute R-squared on the mean scores per condition.

  oo - Observed score, per condition.

  pp - Predicted score, per condition.
```

4.48 tvasavefit

```
TVASAVEFIT
tvasavefit(fitfile,tvadata,tvamodel,theta)
tvasavefit(fitfile,tvadata,tvamodel,theta,theta_fix)
```

4.49 tvasculpt

```
TVASCULPT

 Synopsis
 ========

 [theta,tvamodel,theta_fix] = tvasculpt(theta,tvamodel,theta_fix,op)
 [theta,tvamodel,theta_fix] = tvasculpt(theta,tvamodel,theta_fix,op,par1,par2,...)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - May 2011

 Purpose
 =======

 Re-configure an existing tvamodel and theta pair.

 Inputs
 ======

 op - String indicating the operation to perform according
 to the following table.

  op string | Description
  ----------+-------------------------------------------
    'C2s'   | Go from model with C to model with s.
            | par1 is TVADATA, par2 is the
            | number of s values to use. par can be sinfl.
    'wrep'  | Repeat w values. par1 - Repfactor.
    'a2w'   | Go from scalar alpha to indiv. alpha.

 (Note that theta_fix must be empty for some of these operations
 to be successful due to the use of TVASETVAL)

 Outputs
 =======

 The outputs represent the re-configured model and parameters.
```

4.50 tvaselect

```
TVASELECT
```

```
 Synopsis
 ========

 idx = tvaselect(tvadata,cc)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    September 2011

 Purpose
 =======

 Subset of TVADATA based on condition numbers.

 Inputs
 ======

 cc - List of condition numbers.
```

4.51 tvasetval

```
TVASETVAL

 Synopsis
 ========

 [theta,theta_fix] = tvasetval(theta,tvamodel,theta_fix,chr,value)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - July 2010

 Purpose
 =======

 Set a parameter in the theta / theta_fix pair.

 Inputs
 ======

 chr - Can be ’a’,’w’,’C’,’s’,’v’,’u’,’t’,’m’ to indicate
 which parameter to set.

 value - The value. (Note that value must have the correct
 dimension, see for example TVADEAL, unless theta_fix is
 empty).

 Outputs
 =======

 The outputs represent the updated parameters.
```

4.52 tvashave

```
TVASHAVE

 Synopsis
 ========

 [theta,tvamodel,change] = tvashave(theta,tvamodel,tvadata)
 [theta,tvamodel,change] = tvashave(theta,tvamodel,tvadata,theta_fix,opts)
 [theta,tvamodel,change] = tvashave(theta,tvamodel,tvadata,theta_fix,opts,puttopts)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    February 2010

 Purpose
 =======

 Shave K.

 Inputs
 ======

 opts - [blade, maxit, epsilon], default [0.5 20 0.01];
 epsilon is compared to the largest element of m subject to change
 after each iteration.

 puttopts - see TVAPUTT

 Outputs
 =======
```

4.53 tvasmoothplot

```
TVASMOOTHPLOT

 Synopsis
 ========

 [ED,PMS] = tvasmoothplot(tt,tvamodel,theta,tvadata)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    January 2012

 Purpose
 =======

 Input
 =====

 tt - Masked exposure durations for the plot. To plot
 unmasked plot simply add mu to this.

 Output
```

```
  ======

  PMS - Predicted Mean Score.
```

4.54 tvasniff

```
TVASNIFF

  Synopsis
  ========

  facstr = tvasniff(tvadata)

  -- Author: Mads Dyrholm --
     Center for Visual Cognition, University of Copenhagen.
     May 2011

  Purpose
  =======

  Sniff the data and return a reasonable facstr, see
  also TVAFLEXCHAIN.
```

4.55 tvaspectral

```
TVASPECTRAL

  Synopsis
  ========

  [m1,m2] = tvaspectral(tvadata)

  -- Author: Mads Dyrholm --
     Center for Visual Cognition, University of Copenhagen
     October 2010

  Purpose
  =======

  Check the spectral properties of the paradigm.

  Outputs
  =======

  m1 - The number of disjoint stim networks.

  m2 -
```

4.56 tvastriptheta

```
TVASTRIPTHETA
```

```
 Synopsis
 ========

 [theta,tvamodel,strip,humstrip] = tvastriptheta(theta,tvamodel)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - July 2011

 Purpose
 =======

 take gm from theta and put it in the tvamodel.
 tvamodel.*gm is updated.
```

4.57 tvasweepbinostack

```
TVASWEEPBINOSTACK

 Synopsis
 ========

 [theta,tvamodel,Ka,Km,Pm] = tvasweepbinostack(theta,tvamodel,tvadata)
 [theta,tvamodel,Ka,Km,Pm] = tvasweepbinostack(theta,tvamodel,tvadata,theta_fix)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - April 2010

 Purpose
 =======

 TVA fitting with BINOSTACK K model.
```

4.58 tvasweeptrad

```
TVASWEEPTRAD

 Synopsis
 ========

 [theta,tvamodel] = tvasweeptrad(theta,tvamodel,tvadata)
 [theta,tvamodel] = tvasweeptrad(theta,tvamodel,tvadata,theta_fix)
 [theta,tvamodel] = tvasweeptrad(theta,tvamodel,tvadata,theta_fix,limits)
 [theta,tvamodel] = tvasweeptrad(theta,tvamodel,tvadata,theta_fix,limits,resolutionlist)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    April 2010

 Purpose
 =======

 TVA fitting with TRAD K model.
```

4.59 tvathetacombine

```
TVATHETACOMBINE

 Synopsis
 ========

 [theta,nanfix,notnanfix] = tvathetacombine(theta_adapt,theta_fix)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - July 2010

 Purpose
 =======

 Combine theta and theta_fix. The inverse of TVATHETACOMBUST.

 Outputs
 =======

 nanfix - The nan indices of theta_fix.

 notnanfix - The not-nan indices of theta_fix.
```

4.60 tvathetacombust

```
TVATHETACOMBUST

 Synopsis
 ========

 [theta_adapt,theta_fix] = tvathetacombust(theta,nanfix)

 -- Author: Mads Dyrholm --
    Center for Visual Cognition, University of Copenhagen.
    2009 - May 2010

 Purpose
 =======

 The inverse of TVATHETACOMBINE.
```

4.61 tvaversion

```
TVAVERSION

 Synopsis
 ========

 tvaversion
```

```
  -- Author: Mads Dyrholm --
     Center for Visual Cognition, University of Copenhagen.
     July 2010

  Purpose
  =======

  Print version and engine status to the screen.
```

4.62 tvaxrddist

```
 TVAXRDDIST

  Synopsis
  ========

  [Pd,Ed,Pi] = tvaxrddist(tau,v,K,R,T,s0)

  -- Author: Mads Dyrholm --
     Center for Visual Cognition, University of Copenhagen
     July 2010

  [1] Dyrholm, M., et al. (2011) "Single Trial Inference on Visual
  Attention", AIP Conference Proceedings, Volume 1371, pp. 37-43

  Purpose
  =======

  For PR and WR.
  This routine provides a safe wrapper to the TVACOREXRDDIST
  routine which calls the libTVA kernel algorithm for
  computing ... and to TVACOREXRDPENC which ...

  Inputs
  ======

  (see also TVAPPAR)

  Outputs
  =======

  Pd(i) - The probability that (i-1) distractors were
  encoded.

  Ed - The expected number of distractors encoded.

  Pi - Encoding probability of distractor i.
```

4.63 tvaxrinfer

```
 TVAXRINFER

  Synopsis
```

```
========

S = tvaxrinfer(tvadata,tvamodel,theta,theta_fix)

-- Author: Mads Dyrholm --
   Center for Visual Cognition, University of Copenhagen
   October 2010

[1] Dyrholm, M., et al. (2011) "Single Trial Inference on Visual
Attention", AIP Conference Proceedings, Volume 1371, pp. 37-43

Purpose
=======

For PR and WR compute distractor encoding probabilities
for individual trials, given observed values.
See also TVAXRDDIST.

Output
======

S{n}.Pi - The probability that element "i" is encoded
(Reported targets are set to 1 and non-reported targets
are set to 0).

S{n}.Ed - The expected number of distractors encoded.

S{n}.Pd - A vector with the number of distractors
encoded pdf.

S{n}.Es - Inferred VSTM load.
```

Bibliography

[1] Dyrholm, M., et al. (2011). "Generalizing parametric models by introducing trial-by-trial parameter variability: The case of TVA", *Journal of Mathematical Psychology*, 10.1016/j.jmp.2011.08.005

[2] Kyllingsbæk, S., "Modeling Visual Attention", *Behavior Research Methods*, Volume 38, Issue 1, pp. 123–133 (2006)

[3] Dyrholm, M., et al. (2011). "Single Trial Inference on Visual Attention", *AIP Conference Proceeding*, Volume 1371, pp. 37–43