DELTA SCORING FUNCTION REFERENCE

D. Atanasov

October 14, 2021

1 Introduction

This document describes the the set of functions and the way of their ussage for obtaining the DELTA SCORING approach for test evaluations.

2 Instalation

To install place the folder in the MATLAB path and rename it to +deltaScoring.

3 Ussage

Here a example of a simple ussage of the package.

Suppose the raw dichotomous item response id placed in varible itemScore.

To estimate the items delta by bootstrapping procedure

```
[ItemDelta, estimatedDeltaSE] = ...
        deltaScoring.estimate.itemDeltaBootstrap(
           itemScore);
```

is called. The resulted item deltas and the corresponding standard error of estimate are returned in variables ItemDelta, estimatedDeltaSE.

The classical person D-scores are calculated usind the response paterns in itemScore and already calculated item deltas.

```
personDscores = ...
         deltaScoring.scoring.dScore(ItemDelta,
            itemScore,opt.Dscore_method);
```

Here opt is a structure containing the options for he considered delta scoring model. It can be generated by

```
opt = deltaScoring.scoring.Options;
```

Here and after the default will be thw model RFM2. If a RFM3 model is aquared this can be stated in

```
opt.model = 3;
```

and the corresonding options should be passed to the functions.

The item properties location b and shape s can be obtained by

```
[params, CI, ~, Results] = ...
deltaScoring.estimate.logitDeltaFit(itemScore,
   personDscores,opt);
```

where *params* contains the matrix with corresponding parameters for any item in the test $[b, s]$. The first column corresponds to the location paramater b while the second represents the shape s. If the model is RMF3, the guessing parameter is in the third column.

The matrix *CI* is contains the 95% confidence interval of the estimated values. *Results* contains additional fitting parameters (for example MAD i available in *Results.MAD*).

The persons true scores can be calculated by

```
personTrueScores = ...
deltaScoring.scoring.trueScore(ItemDelta,
   ItemParameters,personDscores,opt);
```

and the SE

```
personTrueScoresSE = ...
deltaScoring.scoring.trueScoreSE(ItemDelta,
   ItemParameters,personDscores,opt);
```

A latent verssion of the location and shape parameters (together with their SE) can be obtained by

```
[LatentParams,LatentSE] =  deltaScoring.estimate.
   ML_RFM_params( itemScore, personDscores, opt);
```

The corresponding MAD is obtained by

```
LatentMAD = deltaScoring.item.MAD(LatentParams,
   Results.observedLogitDelta ,opt);
```

where *Results.observedLogitDelta* is calculated with *deltaScoring.estimate.logitDeltaFit* above and contains the proportion of observed correct answers for the values on the D-score scale.

The corresponding latent values of the person D-scores (and SE) can be obtained by

```
[latentScore, ~, latentScoreSE]= ...
deltaScoring.estimate.ML_RFM_scores( itemScore,
   LatentParams, opt)
```

The estimations of latent item parameters and person D-scores can be iterated untill a convergence is reached

```
OldLatentParams = zeros(size(LatentParams));
while max(abs(OldLatentParams - LatentParams)) > eps
        OldLatentParams = LatentParams;
        [LatentParams,LatentSE] =  ...
                deltaScoring.estimate.ML_RFM_params(
                   itemScore, personDscores, opt);
        [latentScore, ~, latentScoreSE]= ...
                deltaScoring.estimate.ML_RFM_scores(
                   itemScore,LatentParams, opt)
end
```

Equating of different tests can be reached by functions located in *deltaScoring.equating*. Here an example of equating of classical (nonlatent) parameters of the test will be presented. The equating is based of caclulation of two constants A and B which represents the change of scale for the test equationg. These constants can be calculated on the base of item deltas of the target test *targetDeltas*, item deltas (allready colculated above) and few common items between the two tests (stated in the variable *CommonItems*)

```
   [A,B] = deltaScoring.equating.constants(
      targetDeltas, ItemDelta, CommonItems);
   equatedItemDeltas = deltaScoring.equating.rescale
      (ItemDelta,A,B);
   equatedDscores = deltaScoring.scoring.dScore(
      ItemDelta,itemScore,opt.Dscore_method);
```

4 Function reference

deltaScoring.scoring.trueScoreSE