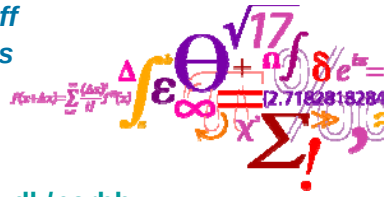
Similarity testing and replicated data (and sensR)

Per Bruun Brockhoff
Professor, Statistics
DTU, Copenhagen
August 17 2015



http://www.staff.dtu.dk/perbb

sensR - Part 2

1. Analysing similarity test data.

2. Planning similarity tests
(power and sample size for similarity testing)

3. Analysing replicated difference test data

The three levels of interpretation
Figure 7.1 (book)

Level 0	Level 1	Level 2
Sensory Scale	Proportion of Correct answers in test of A vs B	Proportion of Discriminators in population
A — B, δ Intensity difference	p_c	$p_d = \frac{p_c - 1/2}{1/2}$ $p_d = \frac{p_c - 1/3}{2/3}$

Analysing Similarity test data

1. Aim: proof that products are (sufficiently) similar!
2. Traditionally, ”Power approach”:
   1. Claim ”similarity” if NOT different
   2. ”Acceptance” of difference test hypothesis of no difference.

3. Better:
   1. Equivalence tests and/or Confidence limits.

Equivalence tests

1. Interchanges the roles of the hypotheses:

H0: Products are NOT similar
H1: Products are similar

Example with specified level of similarity:

$(p_{d0} = 0.25)$

$$H_0 : p_d \geq 0.25 \Leftrightarrow p_c \geq 0.50$$
$$H_A : p_d < 0.25 \Leftrightarrow p_c < 0.50$$

Equivalence tests

For such one-tailed situations this is equivalent to:

- Specify the wanted degree of similarity
- Claim similarity (with 95% confidence) IF the 90% upper (2-tailed) confidence limit is within this specification

Can be used on level 0, level 1 or level 2
BUT: ”*discrim*” requires the Pd0 as input!

Doing the equivalence test

Does 40 out of 100 in a triangle prove a 0.25 Pd-equivalence?

discrim(40,100,pd0=0.25,conf.level=0.90, method="triangle“,test="similarity")

Similarity and difference tests

SAME data may provide non-significant results for BOTH difference and similarity tests:

```
discrim(40, 100, pd0=0.2, conf.level=0.90,
                    method="triangle", test="similarity")

discrim(40, 100, conf.level=0.90, method = "triangle")
```


Similarity and difference tests

SAME data may provide significant results for BOTH difference and similarity tests:

```
discrim(80, 200, pd0=0.25, conf.level=0.90,
                    method="triangle", test="similarity")

discrim(80, 200, conf.level=0.90, method = "triangle")
```


Doing the equivalence test

Does 35 out of 100 in a 3AFC prove a 0.5 d-prime equivalence?

```
mypd0=rescale(d.prime=0.5)$pd

discrim(40,100,pd0=mypd0,conf.level=0.90,
method="3AFC",test="similarity")
```


sensR - Part 2

1. Analysing similarity test data.

2. Planning similarity tests
(power and sample size for similarity testing)

3. Analysing replicated difference test data

Power of Similarity tests

ISO triangle standard does (a version of) the following:

```
discrimPwr(pdA=0,pd0=0.25, sample.size=100, alpha=0.05,pGuess =1/3,
test="similarity")
```

So assumes the true pd to be zero! (pdA=0)

(only best case scenario)

Power of Similarity tests

But what if pdA>0, e.g. pdA=0.2:

(ISO triangle standard ignores this) sensR does the job:

```
discrimPwr(pdA=.2,pd0=0.25, sample.size=100, alpha=0.05,pGuess=1/3,
test="similarity")
```


Explanation of Power of Similarity tests

The critical value of the test can be found:

(The upper limit of being able to prove similarity)

```
findcr(sample.size=100, alpha = .05, p0 = 1/3, pd0 = 0.25,
test = c("similarity"))

#Power:
pbinom(41,100,1/3)
```


Power of Similarity tests

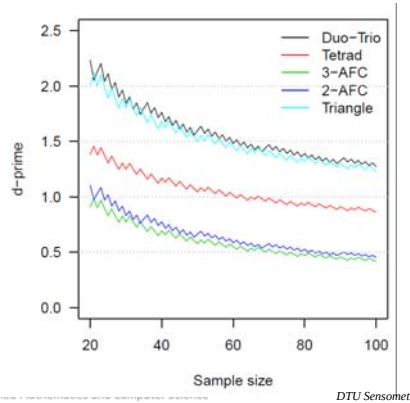
What if similarity is defined on d-prime scale, e.g. d.prime0=0.5:

```
d.primePwr(d.primeA=0,d.prime0=0.5,
sample.size=100, alpha=0.05,method="triangle",
test="similarity")
```

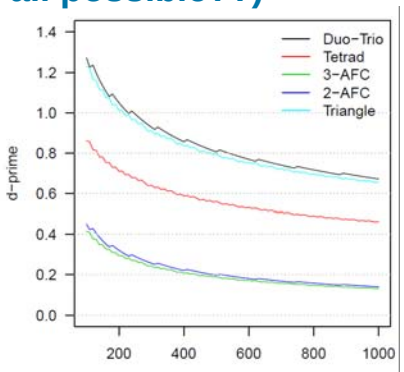
Sample Size for Similarity tests

```
# First on pd-scale with pd0=0.25, best case:
discrimSS(pdA=0,pd0=0.25, target.power=0.9, alpha=0.05,pGuess =1/3,
        test="similarity")

# Still on pd-scale with pd0=0.25, "bad" case:
discrimSS(pdA=.2,pd0=0.25 , target.power=0.9, alpha=0.05, pGuess=1/3,
        test="similarity")

# Then on d.prime-scale with d.prime0=0.5, best case:
d.primeSS(d.primeA=0, d.prime0=0.5, target.power=0.9, alpha=0.05,
method="triangle", test="similarity")

# Then on d.prime-scale with d.prime0=0.5, "bad" case:
d.primeSS(d.primeA=0.4, d.prime0=0.5, target.power=0.9, alpha=0.05,
method="triangle",  test="similarity")
```


Smallest reasonable equivalence margin (What is at all possible??)



Smallest reasonable equivalence margin (What is at all possible??)



sensR - Part 2

1. Analysing similarity test data.

2. Planning similarity tests
(power and sample size for similarity testing)

3. Analysing replicated difference test data

Replicated situations

n individuals each performed k tests

- **Individuals may be different**
- **There may be heterogeneity**
- **The nk observations are not independent**

Actually: the naive (pooled) difference hypothesis test is NOT wrong
BUT: NOT enough for extracting complete information

Replicated traingle tests

Example, n=15, k=12:
2,2,3,3,4,4,4,4,4,4,4,4,5,6,10,11

Naive analysis: sensR:

```
discrim(70,180,method="triangle")
```

Confidence limits generally NOT OK
Test for detecting product difference may NOT be the strongest possible!

Replicated traingle tests How do they come out??

sensR: **Simulation:**

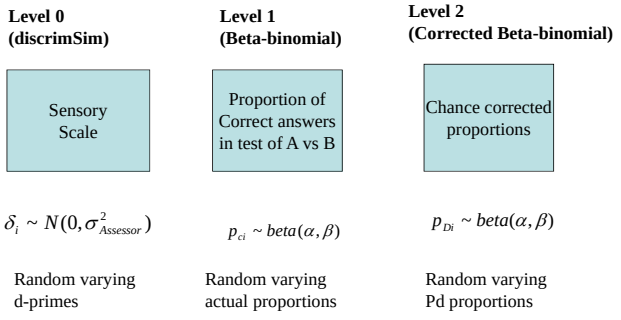
```
# First a case where individuals are "clones" - they are NOT really different,
# AND there is NO product difference:
discrimSim(20, replicates = 12, d.prime = 0, method = "triangle", sd.indiv=0)

# Next a case where individuals are "clones" - they are NOT really different,
# but there IS a product difference, d.prime=2:
discrimSim(20, replicates = 12, d.prime = 2, method = "triangle", sd.indiv=0)

# Finally a case where individuals arereally different, (sd.indiv=2)
# AND there IS a product difference, d.prime=2:
discrimSim(20, replicates = 12, d.prime = 2, method = "triangle", sd.indiv=2)
```


Replicated traingle tests

TWO aspects are now in play:
1. **Average level of difference**
2. **Variability of individual differences**
   1. **"*Usual*" variability**
   2. **"*EXCESS*" variability ("*over-dispersion*")**

Different possible approaches to cope with this! (all: random individuals)

Figure 7.4



Replicated difference tests in light of the three overall analysis levels.

Replicated triangle tests

sensR: **Table 7.4 analysis: (corrected beta-binomial)**

```
> summary(betabin(X2,method="triangle"))

Chance-corrected beta-binomial model for the triangle protocol
with 95 percent confidence intervals

          Estimate   Std. Error  Lower      Upper
mu        0.09779917 0.06599972 0.0000000 0.2271562
gamma     0.62516874 0.20607101 0.2212770 1.0000000
pc        0.39853278 0.04399981 0.3333333 0.4847708
pd        0.09779917 0.06599972 0.0000000 0.2271562
d-prime   0.86877203 0.31203861 0.0000000 1.3855903

log-likelihood:  -30.98173
LR-test of over-dispersion, G^2: 13.05338 df: 1 p-value: 0.0003027369
LR-test of association, G^2: 15.49198 df: 2 p-value: 0.0004324741
```


Replicated triangle tests

sensR: **Table 7.4 analysis: (corrected beta-binomial)**

```
> summary(betabin(X2,method="triangle"))

Chance-corrected beta-binomial model for the triangle protocol
with 95 percent confidence intervals

          Estimate   Std. Error  Lower      Upper
mu        0.09779917 0.06599972 0.0000000 0.2271562
gamma     0.62516874 0.20607101 0.2212770 1.0000000
pc        0.39853278 0.04399981 0.3333333 0.4847708
pd        0.09779917 0.06599972 0.0000000 0.2271562
d-prime   0.86877203 0.31203861 0.0000000 1.3855903
```

Proper Confidence Intervals!

```
log-likelihood:  -30.98173
LR-test of over-dispersion, G^2: 13.05338 df: 1 p-value: 0.0003027369
LR-test of association, G^2: 15.49198 df: 2 p-value: 0.0004324741
```


Replicated triangle tests

sensR: **Table 7.4 analysis: (corrected beta-binomial)**

```
> summary(betabin(X2,method="triangle"))

Chance-corrected beta-binomial model for the triangle protocol
with 95 percent confidence intervals

          Estimate   Std. Error  Lower      Upper
mu        0.09779917 0.06599972 0.0000000 0.2271562
gamma     0.62516874 0.20607101 0.2212770 1.0000000
pc        0.39853278 0.04399981 0.3333333 0.4847708
pd        0.09779917 0.06599972 0.0000000 0.2271562
d-prime   0.86877203 0.31203861 0.0000000 1.3855903
```

Excess varability measure AND test!

```
log-likelihood:  -30.98173
LR-test of over-dispersion, G^2: 13.05338 df: 1 p-value: 0.0003027369
LR-test of association, G^2: 15.49198 df: 2 p-value: 0.0004324741
```

Replicated triangle tests

sensR: **Table 7.4 analysis: (corrected beta-binomial)**

```
> summary(betabin(X2,method="triangle"))

Chance-corrected beta-binomial model for the triangle protocol
with 95 percent confidence intervals

           Estimate   Std. Error  Lower     Upper
mu         0.09779917 0.06599972 0.0000000 0.2271562
gamma      0.62516874 0.20607101 0.2212770 1.0000000
pc         0.39853278 0.04399981 0.3333333 0.4847708
pd         0.09779917 0.06599972 0.0000000 0.2271562
d-prime    0.86877203 0.31203861 0.0000000 1.3855903

log-likelihood:  -30.98173
LR-test of over-dispersion, G^2: 13.05338 df: 1 p-value: 0.0003027369
LR-test of association, G^2: 15.49198 df: 2 p-value: 0.0004324741
```

JOINT Product difference test!

Replicated Difference test

Joint product difference test:

H0: No mean effect AND no excess variability
HA: The products are different

sensR - Part 2

1. Analysing similarity test data.

2. Planning similarity tests
(power and sample size for similarity testing)

3. Analysing replicated difference test data