Detecting categorical perception in continuous discrimination data

Paul Boersma & Kateřina Chládková

University of Amsterdam
Categorical perception

**identification**
- same category = same label

**discrimination**
- same category = difficult discrimination
Categorical perception

**identification**
- same category = same label

**discrimination**
- same category = difficult discrimination

![Graph showing % identification as category a vs stimulus](image-url)
Categorical perception

**identification**
- same category = same label

**discrimination**
- same category = difficult discrimination
Categorical perception

Identification
- same category = same label

Discrimination
- same category = difficult discrimination

Graph showing categorical perception with continuous data.
A problem with previous studies

- small number of different stimuli
- repeated multiple times

Rogers & Davis (2009): such design increases listeners’ categorical bias
A problem with previous studies

- small number of different stimuli
- repeated multiple times

Rogers & Davis (2009): such design increases listeners’ categorical bias
The solution, and a remaining problem

- Rogers & Davis’ (2009) solution: **test categorical perception ‘continuously’**, i.e. on a densely-sampled phonetic continuum, without repetition

- remaining problem with Rogers & Davis: (indentification results: logistic regression,) **discrimination results: non-continuous method of analysis**
The aim of the present study

to provide a **continuous analysis method** for continuous discrimination data
vowel continuum between /i/ and /ɛ/

discrimination along the F1 dimension

---

1Chládková & Benders (in prep.).
Stimuli

- 260 different vowels = 130 stimulus pairs
- equal steps between 280 Hz and 725 Hz (6.93 erb and 12.86 erb)
- 260 different vowels = 130 stimulus pairs
- equal steps between 280 Hz and 725 Hz (6.93 erb and 12.86 erb)
Stimuli

pair30 pair47

Categorical perception with continuous data

Paul Boersma & Kateřina Chládková
Testing procedure

- AX task
- each of the 130 stimulus pairs included twice, i.e. $a - b$ in one trial, $b - a$ in the other trial
- the auditory F1 distance is always the same

- Participants: 62 monolingual Czechs
- Question: How many categories do they have along the continuum?
Testing procedure

- AX task
- each of the 130 stimulus pairs included twice, i.e. $a - b$ in one trial, $b - a$ in the other trial
- the auditory F1 distance is always the same

- **Participants:** 62 monolingual Czechs
- **Question:** How many categories do they have along the continuum?
Testing procedure

- AX task
- each of the 130 stimulus pairs included twice, i.e. $a - b$ in one trial, $b - a$ in the other trial
- the auditory F1 distance is always the same

- **Participants**: 62 monolingual Czechs
- **Question**: How many categories do they have along the continuum?
Data: visual inspection
Visual inspection

Raw data: max 2 ‘different’ responses / pair: peaks hard to find

Smoothed data (convolution with a Gaussian): inspection possible

Categorical perception with continuous data
Paul Boersma & Kateřina Chládková
Visual inspection

Raw data: max 2 ‘different’ responses / pair: peaks hard to find

Smoothed data (convolution with a Gaussian): inspection possible

![Graph showing number of “different” responses vs stimulus pair for Listener 1.](image)
Visual inspection

Raw data: max 2 ‘different’ responses / pair: peaks hard to find

Smoothed data (convolution with a Gaussian): inspection possible

![Graph showing number of "different" responses against stimulus pair for Listener 2.]
Raw data: max 2 ‘different’ responses / pair: peaks hard to find

Smoothed data (convolution with a Gaussian): inspection possible
Visual inspection

Raw data: max 2 ‘different’ responses / pair: peaks hard to find

Smoothed data (convolution with a Gaussian): inspection possible
Visual inspection

Raw data: max 2 ‘different’ responses / pair: peaks hard to find

**Smoothed data** (convolution with a Gaussian): inspection possible

![Graph showing number of "different" responses vs stimulus pair](image-url)
Data: analysis
Analysis: estimate the ‘best’ number of peaks

• per listener, **model** the data with **every possible number of discrimination peaks**

  • estimate the **best value of the parameters** that define a model with \( n \) discrimination peaks
    • 0 peaks: \( p_{\text{const}} \)
    • 1 peak: \( p_{\text{min}}, p_{\text{max}}, \mu, \sigma \)
    • 2 peaks: \( p_{\text{min}}, p_{1\text{max}}, \mu_1, \sigma_1, p_{2\text{max}}, \mu_2, \sigma_2 \)
    • ...

• **find which model best fits the data**,
  i.e. test whether a model fits the data significantly better than the preceding simpler model
Analysis: estimate the ‘best’ number of peaks

- per listener, **model** the data with **every possible number of discrimination peaks**

- estimate the **best value of the parameters** that define a model with \( n \) discrimination peaks
  - 0 peaks: \( p_{\text{const}} \)
  - 1 peak: \( p_{\text{min}}, p_{\text{max}}, \mu, \sigma \)
  - 2 peaks: \( p_{\text{min}}, p_{1\text{max}}, \mu_1, \sigma_1, p_{2\text{max}}, \mu_2, \sigma_2 \)
  - ...

- find which model best fits the data, i.e. test whether a model fits the data significantly better than the preceding simpler model
Analysis: estimate the ‘best’ number of peaks

- per listener, **model** the data with **every possible number of discrimination peaks**

- estimate the **best value of the parameters** that define a model with \( n \) discrimination peaks
  - 0 peaks: \( p_{\text{const}} \)
  - 1 peak: \( p_{\text{min}}, p_{\text{max}}, \mu, \sigma \)
  - 2 peaks: \( p_{\text{min}}, p_{1\text{max}}, \mu_1, \sigma_1, p_{2\text{max}}, \mu_2, \sigma_2 \)
  - ...

- **find which model best fits the data**, i.e. test whether a model fits the data significantly better than the preceding simpler model
Best fit = 0 peaks

⇒ listens acoustically or has one category

(no evidence for more, \( p = 0.11 \))
Best fit = 1 peak

⇒ has at least two categories, $p = 2.1 \cdot 10^{-12}$

(no evidence for more, $p = 0.28$)
Model: listener 3

Best fit = 2 peaks

⇒ has at least three categories, $p = 0.00011$

(no evidence for more, $p = 0.93$)
Conclusions

Method of analysis of continuous discrimination data

- finds the plausible (minimum) number of categories
- estimates location and crispness of category boundaries
- preserves the continuous nature of the data
Method of analysis of continuous discrimination data

- finds the plausible (minimum) number of categories
- estimates location and crispness of category boundaries
- preserves the continuous nature of the data
Thank you.
Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks
parameters: $p_{\text{min}}, p_{\text{1max}}, \mu_1, \sigma_1, p_{\text{2max}}, \mu_2, \sigma_2$

1. assign random values to the 7 parameters
2. randomly change the 7 parameters a little bit
3. check whether $LL$ improves (becomes less negative)
4. if $LL$ improves, keep the values of the parameters
5. repeat steps 2 - 4 1000 times
6. repeat steps 1 - 5 100 times
7. keep the parameters that give the best $LL$
Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks

parameters: \( p_{min}, p_{1max}, \mu_1, \sigma_1, p_{2max}, \mu_2, \sigma_2 \)

1. assign random values to the 7 parameters
2. randomly change the 7 parameters a little bit
3. check whether \( LL \) improves (becomes less negative)
4. if \( LL \) improves, keep the values of the parameters
5. repeat steps 2 - 4 1000 times
6. repeat steps 1 - 5 100 times
7. keep the parameters that give the best \( LL \)
Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks
parameters: $p_{min}$, $p_{1max}$, $\mu_1$, $\sigma_1$, $p_{2max}$, $\mu_2$, $\sigma_2$

1. assign random values to the 7 parameters
2. randomly change the 7 parameters a little bit
3. check whether $LL$ improves (becomes less negative)
4. if $LL$ improves, keep the values of the parameters
5. repeat steps 2 - 4 1000 times
6. repeat steps 1 - 5 100 times
7. keep the parameters that give the best $LL$
Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks

parameters: $p_{\text{min}}, p_{1\text{max}}, \mu_1, \sigma_1, p_{2\text{max}}, \mu_2, \sigma_2$

1. assign random values to the 7 parameters
2. randomly change the 7 parameters a little bit
3. check whether $LL$ improves (becomes less negative)
4. if $LL$ improves, keep the values of the parameters
5. repeat steps 2 - 4 1000 times
6. repeat steps 1 - 5 100 times
7. keep the parameters that give the best $LL$
Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks
parameters: \( p_{\text{min}}, p_{1\text{max}}, \mu_1, \sigma_1, p_{2\text{max}}, \mu_2, \sigma_2 \)

1. assign random values to the 7 parameters
2. randomly change the 7 parameters a little bit
3. check whether \( LL \) improves (becomes less negative)
4. if \( LL \) improves, keep the values of the parameters
5. repeat steps 2 - 4 1000 times
6. repeat steps 1 - 5 100 times
7. keep the parameters that give the best \( LL \)
Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks
parameters: $p_{\text{min}}, p_{1\text{max}}, \mu_1, \sigma_1, p_{2\text{max}}, \mu_2, \sigma_2$

1. assign random values to the 7 parameters
2. randomly change the 7 parameters a little bit
3. check whether $LL$ improves (becomes less negative)
4. if $LL$ improves, keep the values of the parameters
5. repeat steps 2 - 4 1000 times
6. repeat steps 1 - 5 100 times
7. keep the parameters that give the best $LL$
Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks
parameters: \( p_{\text{min}}, p_{1\text{max}}, \mu_1, \sigma_1, p_{2\text{max}}, \mu_2, \sigma_2 \)

1. assign random values to the 7 parameters
2. randomly change the 7 parameters a little bit
3. check whether \( LL \) improves (becomes less negative)
4. if \( LL \) improves, keep the values of the parameters
5. repeat steps 2-4 1000 times
6. repeat steps 1-5 100 times
7. keep the parameters that give the best \( LL \)
The algorithm for 2 peaks

Parameters: $p_{\text{min}}, p_{1\text{max}}, \mu_1, \sigma_1, p_{2\text{max}}, \mu_2, \sigma_2$

1. Assign random values to the 7 parameters
2. Randomly change the 7 parameters a little bit
3. Check whether $LL$ improves (becomes less negative)
4. If $LL$ improves, keep the values of the parameters
5. Repeat steps 2 - 4 1000 times
6. Repeat steps 1 - 5 100 times
7. Keep the parameters that give the best $LL$