Detecting categorical perception in continuous discrimination data

Paul Boersma & Kateřina Chládková

University of Amsterdam

Amsterdam Center for Language and Communication ACLC

Interspeech 2010, Makuhari Japan, 27 September 2010

Categorical perception

identification

- same category = same label

discrimination

- same category = difficult discrimination

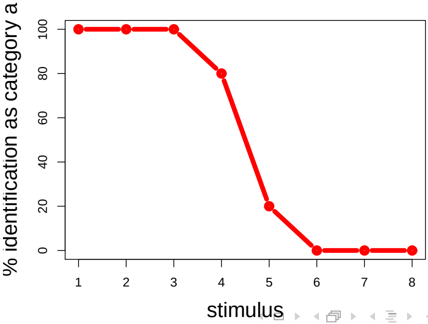
Categorical perception

identification

- same category = same label

discrimination

- same category = difficult discrimination



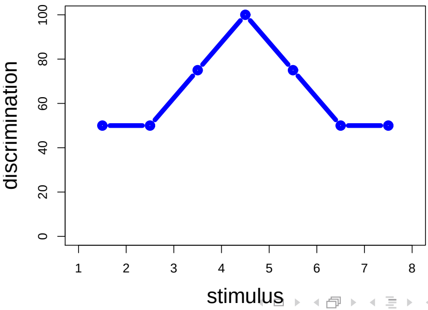
Categorical perception

identification

- same category = same label

discrimination

- same category = difficult discrimination



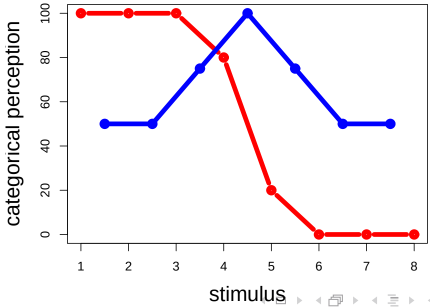
Categorical perception

identification

- same category = same label

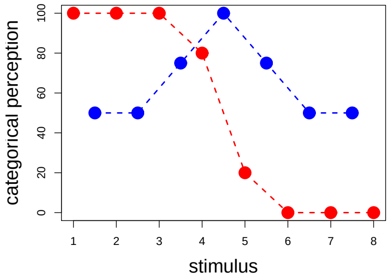
discrimination

- same category = difficult discrimination



A problem with previous studies

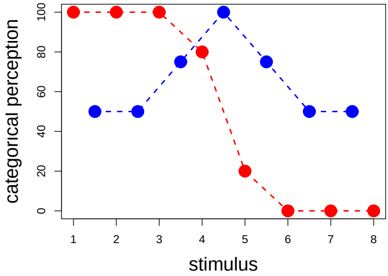
- small number of different stimuli
- repeated multiple times



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

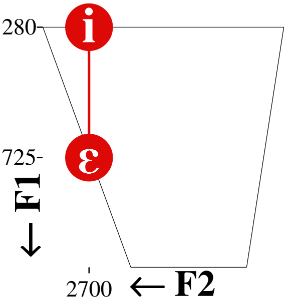
- remaining problem with Rogers & Davis:
  (indentification results: logistic regression,)
  discrimination results: non-continuous method of analysis

The aim of the present study

to provide a **continuous analysis method** for continuous discrimination data

Part of a larger experiment[1]

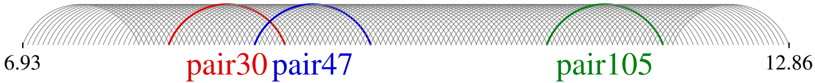
- vowel continuum between /i/ and /ɛ/
- discrimination along the F1 dimension



[1] Chládková & Benders (in prep.).

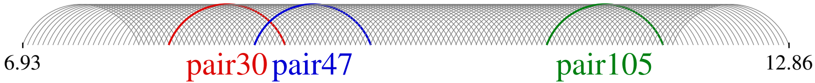
Stimuli

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

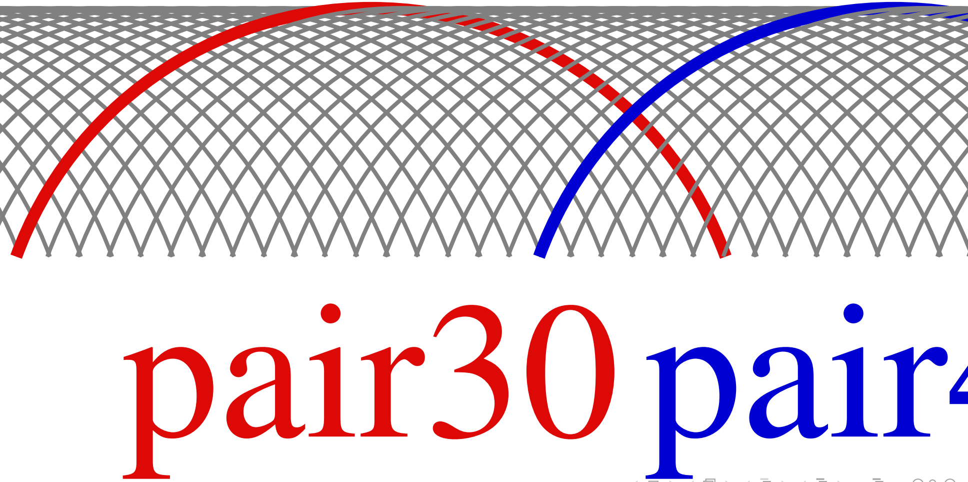


Stimuli

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



Stimuli



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

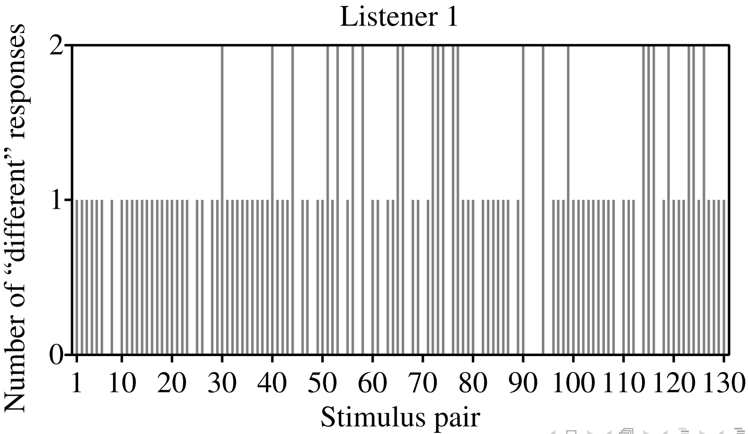
- **Participants:** 62 monolingual Czechs
- **Question:** How many categories do they have along the continuum?

Data: visual inspection

Visual inspection

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

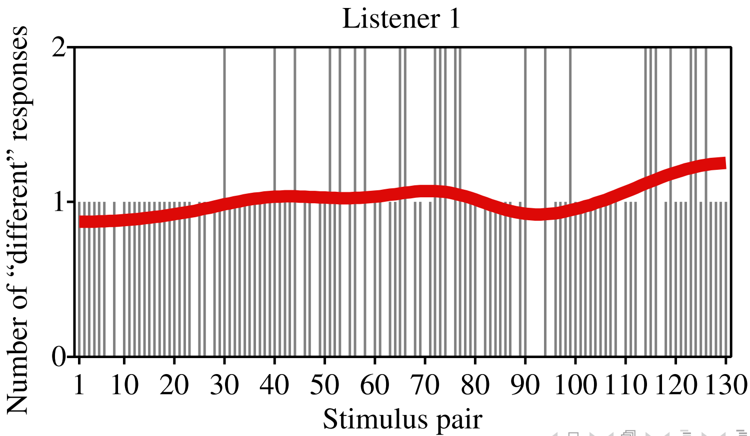
Smoothed data (convolution with a Gaussian): inspection possible



Visual inspection

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

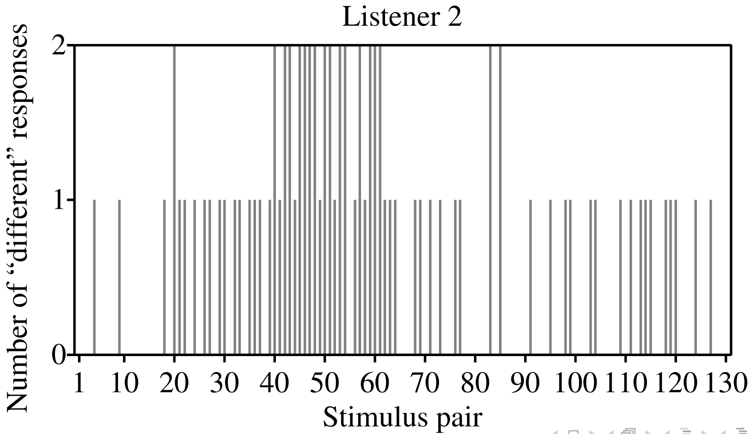
Smoothed data (convolution with a Gaussian): inspection possible



Visual inspection

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

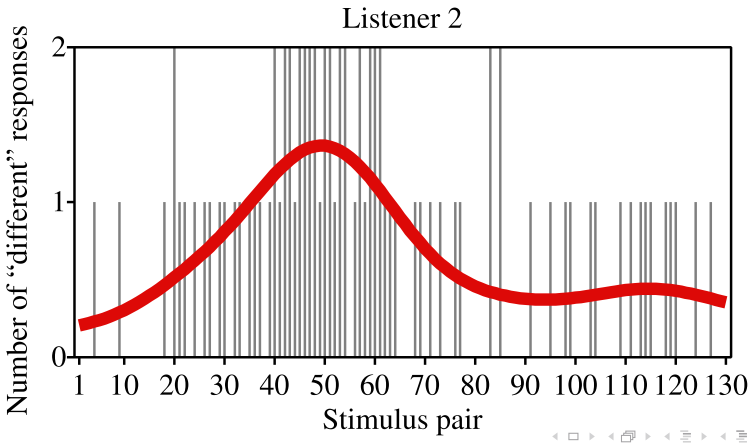
Smoothed data (convolution with a Gaussian): inspection possible



Visual inspection

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

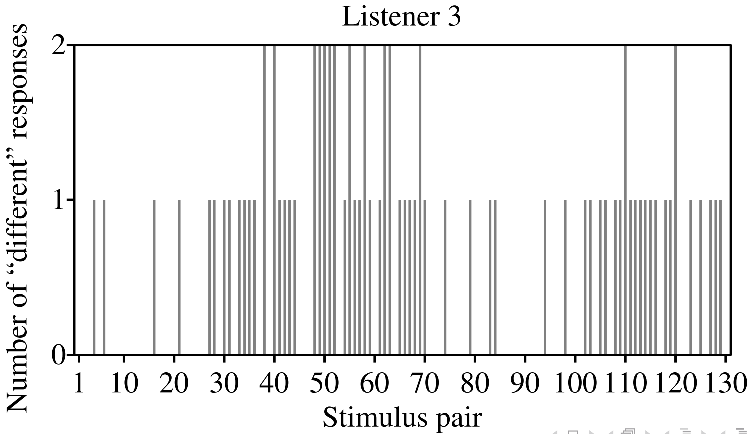
Smoothed data (convolution with a Gaussian): inspection possible



Visual inspection

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

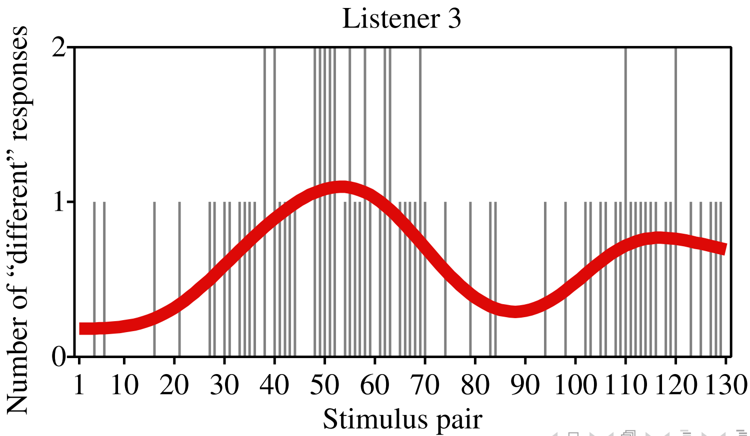
Smoothed data (convolution with a Gaussian): inspection possible



Visual inspection

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

Smoothed data (convolution with a Gaussian): inspection possible



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_{const}
  - 1 peak: p_{min}, p_{max}, μ, σ
  - 2 peaks: p_{min}, p_{1max}, μ_1, σ_1, p_{2max}, μ_2, σ_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_{const}
  - 1 peak: p_{min}, p_{max}, μ, σ
  - 2 peaks: p_{min}, p_{1max}, μ_1, σ_1, p_{2max}, μ_2, σ_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

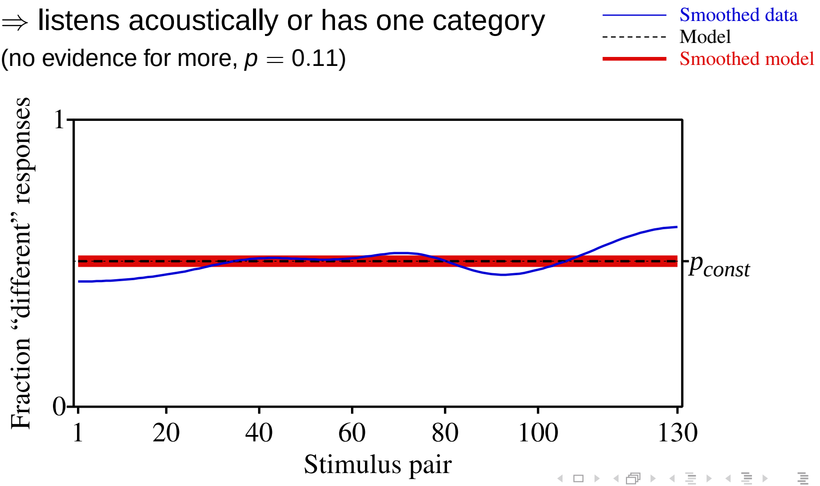
- per listener, **model** the data with **every possible number of discrimination peaks**
- estimate the **best value of the parameters** that define a model with *n* discrimination peaks
  - 0 peaks: p_{const}
  - 1 peak: p_{min}, p_{max}, μ, σ
  - 2 peaks: p_{min}, p_{1max}, μ_1, σ_1, p_{2max}, μ_2, σ_2
  - ...
- **find which model best fits the data**,
  i.e. test whether a model fits the data significantly better than the preceding simpler model

Model: listener 1

Best fit = 0 peaks

⇒ listens acoustically or has one category

(no evidence for more, $p = 0.11$)

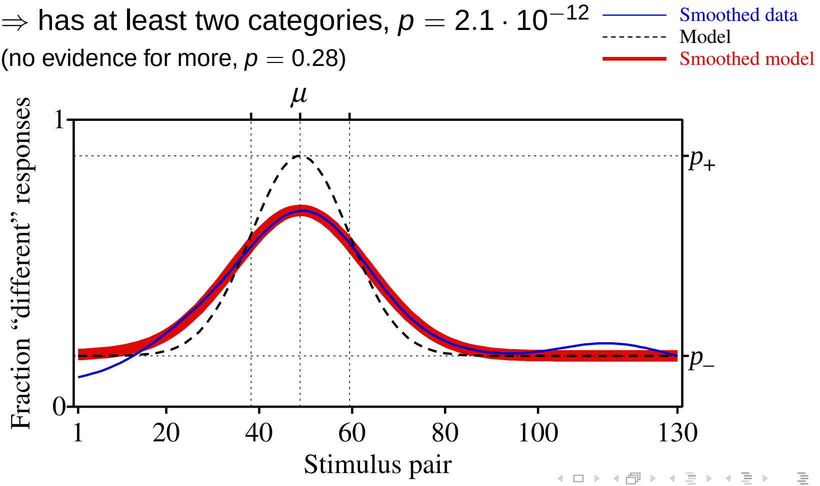


Model: listener 2

Best fit = 1 peak

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

(no evidence for more, $p = 0.28$)

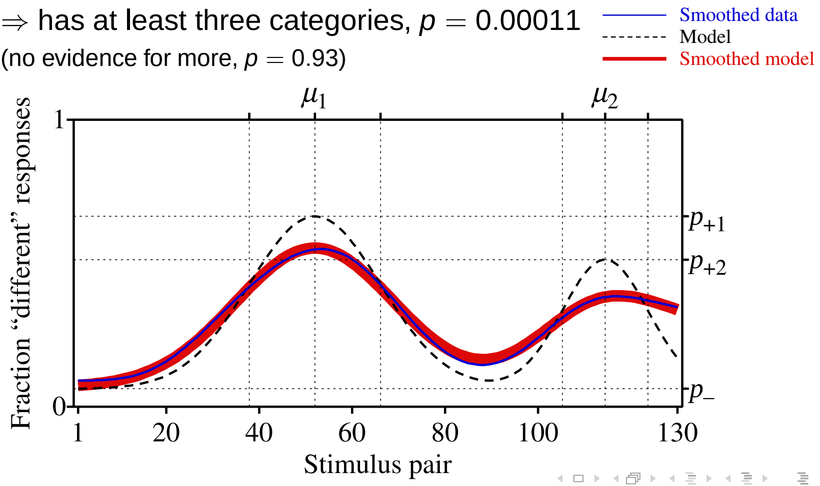


Model: listener 3

Best fit = 2 peaks

$\Rightarrow$ 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

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

Thank you.

Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks

parameters: p_{min}, p_{1max}, μ_1, σ_1, p_{2max}, μ_2, σ_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 best LL*

Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks

parameters: p_{min}, p_{1max}, μ_1, σ_1, p_{2max}, μ_2, σ_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 best LL*

Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks

parameters: p_{min}, p_{1max}, μ_1, σ_1, p_{2max}, μ_2, σ_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 best LL*

Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks

parameters: p_{min}, p_{1max}, μ_1, σ_1, p_{2max}, μ_2, σ_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 best LL*

Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks

parameters: p_{min}, p_{1max}, μ_1, σ_1, p_{2max}, μ_2, σ_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 best LL*

Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks

parameters: p_{min}, p_{1max}, μ_1, σ_1, p_{2max}, μ_2, σ_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 best LL*

Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks

parameters: p_{min}, p_{1max}, μ_1, σ_1, p_{2max}, μ_2, σ_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 best LL*

Maximum-likelihood fit: algorithm

→ the algorithm for 2 peaks

parameters: p_{min}, p_{1max}, μ_1, σ_1, p_{2max}, μ_2, σ_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 best LL*