Computing the Fundamental Frequency Variation Spectrum in Conversational Spoken Dialogue Systems

Kornel Laskowski[a,b],
Matthias Wölfel[b], Mattias Heldner[c] & Jens Edlund[c]

[a]CMU, Pittsburgh PA, USA
[b]UKA(TH), Karlsruhe, Germany
[c]KTH, Stockholm, Sweden

2 July, 2008

Fundamental Frequency (F0) Variation (FFV)

- how does F0 vary in time?
- FFV: ongoing work, building on ICASSP 2008 and Speech Prosody 2008
- OUR ULTIMATE GOAL: ability to automatically learn prosodic sequences characterizing various phenomena

Canonical Measurement of F0 Variation

1. estimate frame-level autocorrelation
2. find local maxima
3. identify best maximum via dynamic programming across multiple frames
4. median filter maxima across multiple frames
5. syllabify speech via ASR or landmark detection
6. fit linear model acros multiple frames in same syllable
7. estimate speaker's baseline pitch across multiple frames
8. normalize out baseline

Canonical Measurement of F0 Variation

1. estimate frame-level autocorrelation
2. find local maxima
3. identify best maximum via dynamic programming across multiple frames
4. median filter maxima across multiple frames
5. syllabify speech via ASR or landmark detection
6. fit linear model acros multiple frames in same syllable
7. estimate speaker's baseline pitch across multiple frames
8. normalize out baseline

Canonical Measurement of F0 Variation

1. estimate frame-level autocorrelation
2. find local maxima
3. identify best maximum via dynamic programming across multiple frames
4. median filter maxima across multiple frames
5. syllabify speech via ASR or landmark detection
6. fit linear model acros multiple frames in same syllable
7. estimate speaker's baseline pitch across multiple frames
8. normalize out baseline

Canonical Measurement of F0 Variation

1. estimate frame-level autocorrelation
2. find local maxima
3. identify best maximum via dynamic programming across multiple frames
4. median filter maxima across multiple frames
5. syllabify speech via ASR or landmark detection
6. fit linear model acros multiple frames in same syllable
7. estimate speaker's baseline pitch across multiple frames
8. normalize out baseline

Canonical Measurement of F0 Variation

1. estimate frame-level autocorrelation
2. find local maxima
3. identify best maximum via dynamic programming across multiple frames
4. median filter maxima across multiple frames
5. syllabify speech via ASR or landmark detection
6. fit linear model acros multiple frames in same syllable
7. estimate speaker's baseline pitch across multiple frames
8. normalize out baseline

Canonical Measurement of F0 Variation

1. estimate frame-level autocorrelation
2. find local maxima
3. identify best maximum via dynamic programming across multiple frames
4. median filter maxima across multiple frames
5. syllabify speech via ASR or landmark detection
6. fit linear model acros multiple frames in same syllable
7. estimate speaker's baseline pitch across multiple frames
8. normalize out baseline

Canonical Measurement of F0 Variation

1. estimate frame-level autocorrelation
2. find local maxima
3. identify best maximum via dynamic programming across multiple frames
4. median filter maxima across multiple frames
5. syllabify speech via ASR or landmark detection
6. fit linear model acros multiple frames in same syllable
7. estimate speaker's baseline pitch across multiple frames
8. normalize out baseline

Canonical Measurement of F0 Variation

1. estimate frame-level autocorrelation
2. find local maxima
3. identify best maximum via dynamic programming across multiple frames
4. median filter maxima across multiple frames
5. syllabify speech via ASR or landmark detection
6. fit linear model acros multiple frames in same syllable
7. estimate speaker's baseline pitch across multiple frames
8. normalize out baseline

Wish List

Would like

- a representation which is:
  - continuous: not undefined in unvoiced regions
  - instantaneous: no long-distance constraints
  - distributed: vector-valued rather than scalar-valued
  - sparse: minimally redundant
- and which:
- FFV appears to satisfy all these constraints/requirements

Wish List

Would like

- a representation which is:
  - continuous: not undefined in unvoiced regions
  - instantaneous: no long-distance constraints
  - distributed: vector-valued rather than scalar-valued
  - sparse: minimally redundant
- and which:

- FFV appears to satisfy all these constraints/requirements

Wish List

Would like

- a representation which is:
  - continuous: not undefined in unvoiced regions
  - instantaneous: no long-distance constraints
  - distributed: vector-valued rather than scalar-valued
  - sparse: minimally redundant
- and which:
- FFV appears to satisfy all these constraints/requirements

Wish List

Would like

- a representation which is:
  - continuous: not undefined in unvoiced regions
  - instantaneous: no long-distance constraints
  - distributed: vector-valued rather than scalar-valued
  - sparse: minimally redundant
- and which:
  - exhibits speaker-independence: no normalization necessary
  - enjoys perceptual relevance: variation in octaves per time
  - lends itself to a wealth of ASR HMM modeling techniques
- FFV appears to satisfy all these constraints/requirements

Wish List

Would like

- a representation which is:
  - continuous: not undefined in unvoiced regions
  - instantaneous: no long-distance constraints
  - distributed: vector-valued rather than scalar-valued
  - sparse: minimally redundant
- and which:
  - exhibits speaker-independence: no normalization necessary
  - enjoys perceptual relevance: variation in octaves per time
  - lends itself to a wealth of ASR HMM modeling techniques
- FFV appears to satisfy all these constraints/requirements

Wish List

Would like

- a representation which is:
  - continuous: not undefined in unvoiced regions
  - instantaneous: no long-distance constraints
  - distributed: vector-valued rather than scalar-valued
  - sparse: minimally redundant
- and which:
  - exhibits speaker-independence: no normalization necessary
  - enjoys perceptual relevance: variation in octaves per time
  - lends itself to a wealth of ASR HMM modeling techniques
- FFV appears to satisfy all these constraints/requirements

Wish List

Would like

- a representation which is:
  - continuous: not undefined in unvoiced regions
  - instantaneous: no long-distance constraints
  - distributed: vector-valued rather than scalar-valued
  - sparse: minimally redundant
- and which:
  - exhibits speaker-independence: no normalization necessary
  - enjoys perceptual relevance: variation in octaves per time
  - lends itself to a wealth of ASR HMM modeling techniques
- FFV appears to satisfy all these constraints/requirements

Wish List

Would like

- a representation which is:
  - continuous: not undefined in unvoiced regions
  - instantaneous: no long-distance constraints
  - distributed: vector-valued rather than scalar-valued
  - sparse: minimally redundant
- and which:
  - exhibits speaker-independence: no normalization necessary
  - enjoys perceptual relevance: variation in octaves per time
  - lends itself to a wealth of ASR HMM modeling techniques
- FFV appears to satisfy all these constraints/requirements

Wish List

Would like

- a representation which is:
  - continuous: not undefined in unvoiced regions
  - instantaneous: no long-distance constraints
  - distributed: vector-valued rather than scalar-valued
  - sparse: minimally redundant
- and which:
  - exhibits speaker-independence: no normalization necessary
  - enjoys perceptual relevance: variation in octaves per time
  - lends itself to a wealth of ASR HMM modeling techniques
- FFV appears to satisfy all these constraints/requirements

Applications in Speech Technology

- identification of places to use back-channel feedback
- classification of rhetorical relations
- interpretation of discourse markers
- dialogue act tagging
- identification of speech repairs

- here, **prediction of speaker change** in conversational spoken dialogue systems

Applications in Speech Technology

- identification of places to use back-channel feedback
- classification of rhetorical relations
- interpretation of discourse markers
- dialogue act tagging
- identification of speech repairs

- here, **prediction of speaker change** in conversational spoken dialogue systems

Applications in Speech Technology

- identification of places to use back-channel feedback
- classification of rhetorical relations
- interpretation of discourse markers
- dialogue act tagging
- identification of speech repairs

- here, **prediction of speaker change** in conversational spoken dialogue systems

Applications in Speech Technology

- identification of places to use back-channel feedback
- classification of rhetorical relations
- interpretation of discourse markers
- dialogue act tagging
- identification of speech repairs

- here, **prediction of speaker change** in conversational spoken dialogue systems

Applications in Speech Technology

- identification of places to use back-channel feedback
- classification of rhetorical relations
- interpretation of discourse markers
- dialogue act tagging
- identification of speech repairs

- here, **prediction of speaker change** in conversational spoken dialogue systems

Applications in Speech Technology

- identification of places to use back-channel feedback
- classification of rhetorical relations
- interpretation of discourse markers
- dialogue act tagging
- identification of speech repairs

- here, **prediction of speaker change** in conversational spoken dialogue systems

Applications in Speech Technology

- identification of places to use back-channel feedback
- classification of rhetorical relations
- interpretation of discourse markers
- dialogue act tagging
- identification of speech repairs

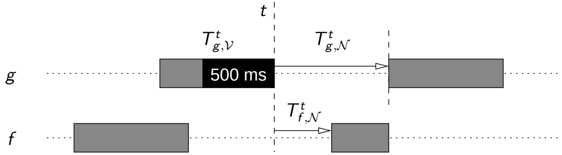
- here, **prediction of speaker change** in conversational spoken dialogue systems

Outline

1. Introduction & Motivation
2. Speaker-Change Prediction
3. Windowing Experiments
4. Conclusions

Speaker-Change Prediction in Dialogue Systems

- in other words: is the speaker finished?
- study how *humans* behave, towards humans
- learn from what actually happens: no need to label data

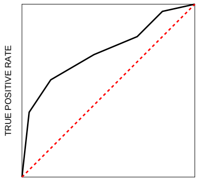


$$L_t = \begin{cases} \mathrm{SC} & \text{if } T^t_{f,\mathcal{N}} - T^t_{g,\mathcal{N}} < 0 \\ \neg\mathrm{SC}, & \text{otherwise} \end{cases} \quad (1)$$

Assessing Performance

- receiver operating characteristic (ROC) curves: true vs false positive rate
- performance of random guessing: line of no discrimination
- discrimination: area A below the ROC curve, $0 \leq A \leq 1$
- in this work: area A between the ROC curve and the *line of no discrimination*, $0 \leq A \leq \frac{1}{2}$

Assessing Performance



- receiver operating characteristic (ROC) curves: true vs false positive rate
- performance of random guessing: line of no discrimination
- discrimination: area A below the ROC curve, $0 \leq A \leq 1$
- in this work: area A between the ROC curve and the *line of no discrimination*, $0 \leq A \leq \frac{1}{2}$

Assessing Performance

- receiver operating characteristic (ROC) curves: true vs false positive rate
- performance of random guessing: line of no discrimination
- discrimination: area A below the ROC curve, $0 \leq A \leq 1$
- in this work: area A between the ROC curve and the *line of no discrimination*, $0 \leq A \leq \frac{1}{2}$

Assessing Performance

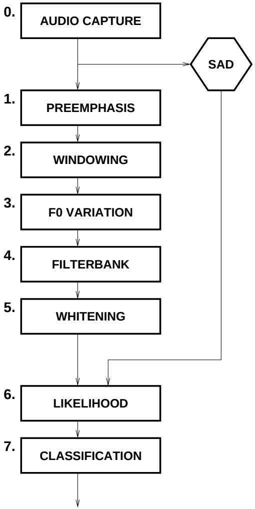
- receiver operating characteristic (ROC) curves: true vs false positive rate
- performance of random guessing: line of no discrimination
- discrimination: area A below the ROC curve, $0 \leq A \leq 1$
- in this work: area A between the ROC curve and the *line of no discrimination*, $0 \leq A \leq \frac{1}{2}$

Data

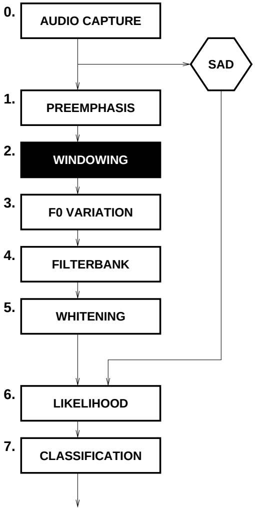
- interactive human-human dialogues
- Swedish Map Task Corpus:

Data Set	Duration (mn:ss)	Dialogue role g		
		speakers	# EOTs	# SCs
DevSet	77:40	F4,F5,M2,M3	480	222
EvalSet	60:39	F1,F2,F3,M1	317	149

System Architecture

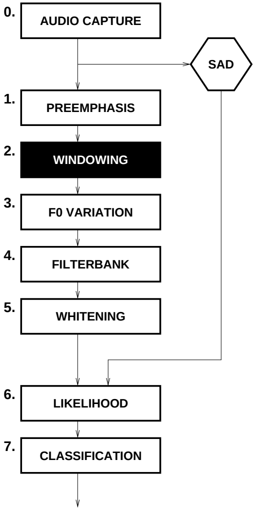


Step 2: Windowing (& FFT Computation)

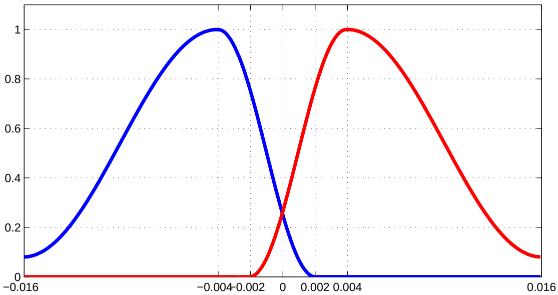


1. Spectral estimation over left and right portions of analysis frame.

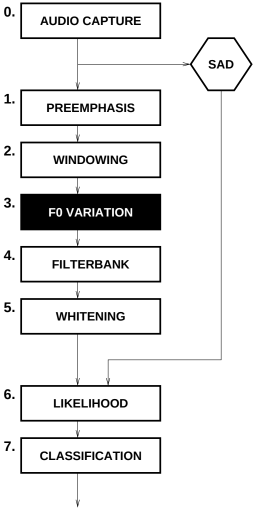
Step 2: Windowing (& FFT Computation)



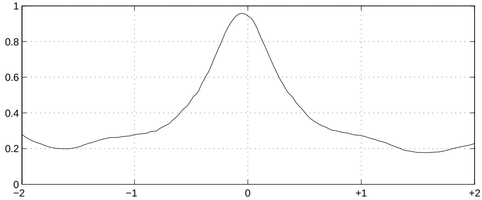
1. Spectral estimation over left and right portions of analysis frame.



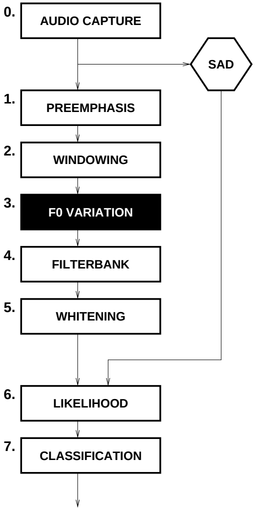
Step 3: F0 Variation (FFV) Computation



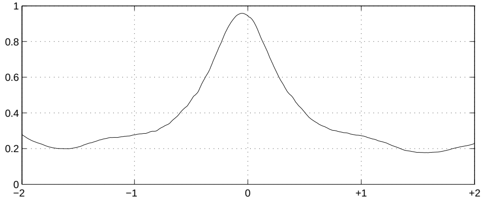
1. Dilate left FFT, dot product with right FFT; & vice versa. (ICASSP'2008)
2. Maximum over resulting spectrum represents change in octaves per second.



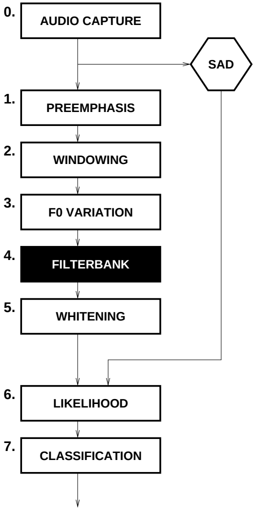
Step 3: F0 Variation (FFV) Computation



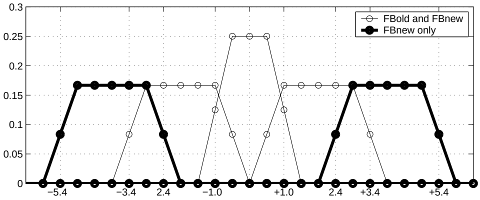
1. Dilate left FFT, dot product with right FFT; & vice versa. (ICASSP'2008)
2. Maximum over resulting spectrum represents change in octaves per second.



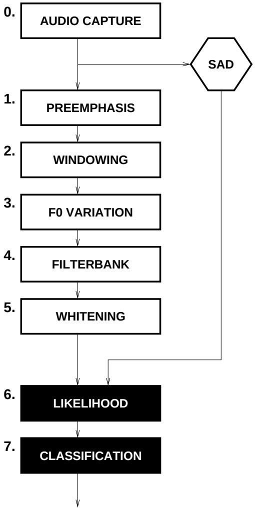
Step 4: Application of Filterbank



1. Compress spectral representation to 7-element vector. (SpeechProsody'2008)

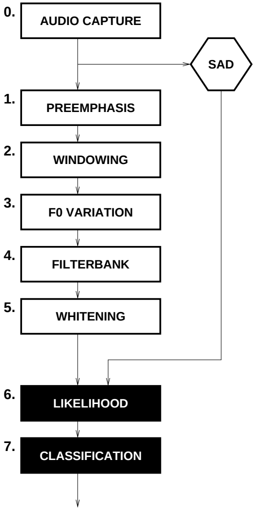


Step 6: Modeling



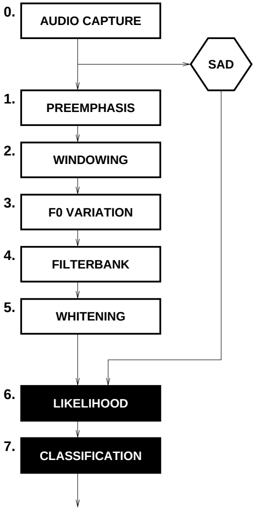
1. For each class (SC/¬SC), train 10 HMMs.
2. Maximum likelihood classification.
3. → 100 candidate dividing hyperplanes.
4. Compute the mean/min/max discrimination over these 100.
5. Compute the single hyperplane ("prod") between 2 class products of 10 models each.

Step 6: Modeling



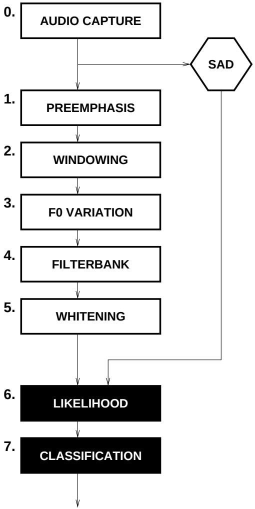
1. For each class (SC/¬SC), train 10 HMMs.
2. Maximum likelihood classification.
3. → 100 candidate dividing hyperplanes.
4. Compute the mean/min/max discrimination over these 100.
5. Compute the single hyperplane ("prod") between 2 class products of 10 models each.

Step 6: Modeling



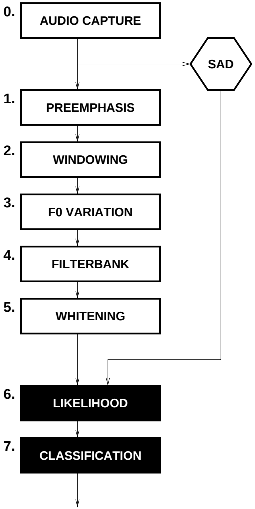
1. For each class (SC/¬SC), train 10 HMMs.
2. Maximum likelihood classification.
3. → 100 candidate dividing hyperplanes.
4. Compute the mean/min/max discrimination over these 100.
5. Compute the single hyperplane ("prod") between 2 class products of 10 models each.

Step 6: Modeling



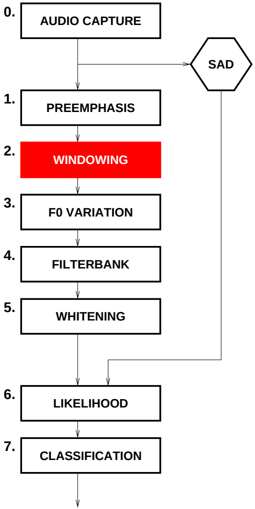
1. For each class (SC/¬SC), train 10 HMMs.
2. Maximum likelihood classification.
3. → 100 candidate dividing hyperplanes.
4. Compute the mean/min/max discrimination over these 100.
5. Compute the single hyperplane ("prod") between 2 class products of 10 models each.

Step 6: Modeling



1. For each class (SC/¬SC), train 10 HMMs.
2. Maximum likelihood classification.
3. → 100 candidate dividing hyperplanes.
4. Compute the mean/min/max discrimination over these 100.
5. Compute the single hyperplane ("prod") between 2 class products of 10 models each.

Focus of This Work



- In this work, investigate sensitivity of speaker-change prediction performance on windowing policy

Two Experiments

Observation: Baseline asymmetric windows are known to have poor frequency resolution.

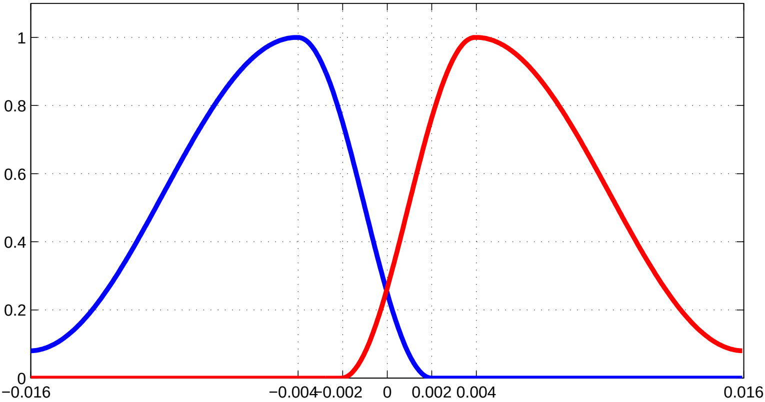
1. Keep window separation fixed; increase overlap to symmetrize.
2. Keep overlap fixed; increase window separation to symmetrize.

Experiment 1

- keep window maxima a constant t_{sep} apart
- less asymmetry $\leftrightarrow$ more window support overlap

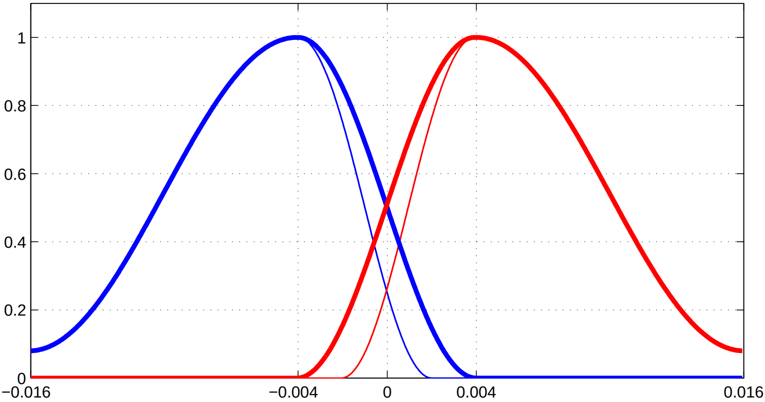
Experiment 1

- keep window maxima a constant t_{sep} apart
- less asymmetry ↔ more window support overlap



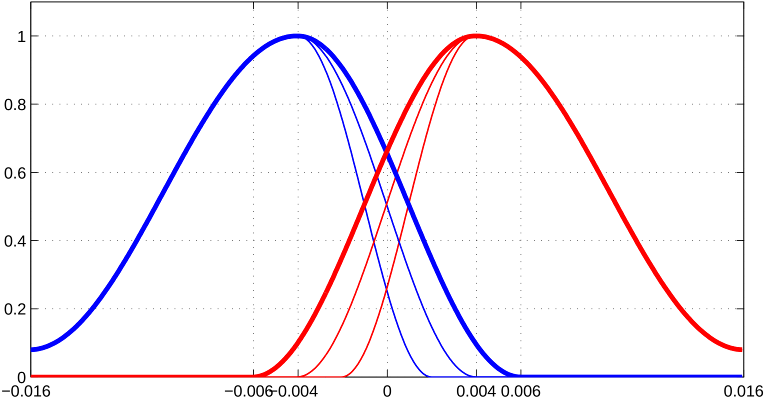
Experiment 1

- keep window maxima a constant t_{sep} apart
- less asymmetry ↔ more window support overlap



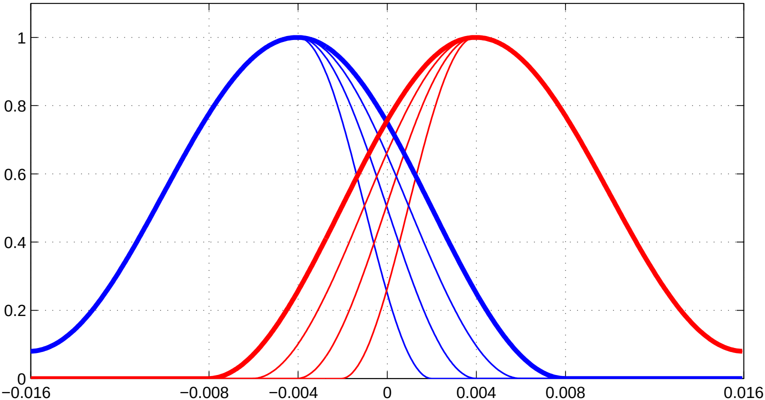
Experiment 1

- keep window maxima a constant t_{sep} apart
- less asymmetry ↔ more window support overlap



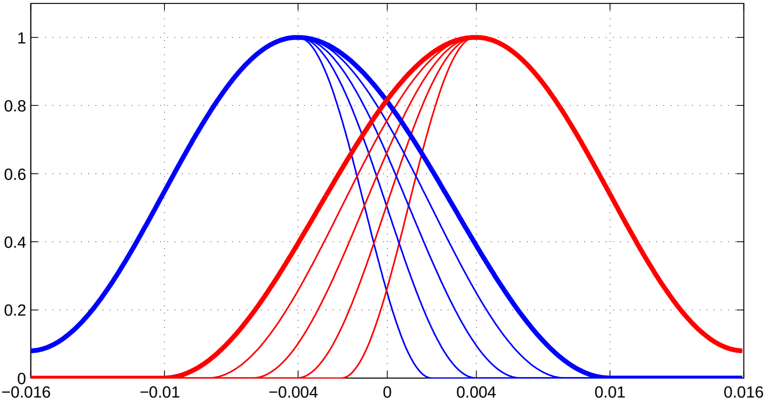
Experiment 1

- keep window maxima a constant t_{sep} apart
- less asymmetry ↔ more window support overlap

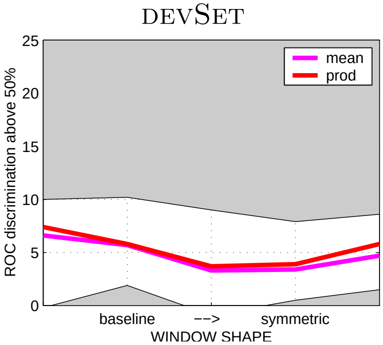


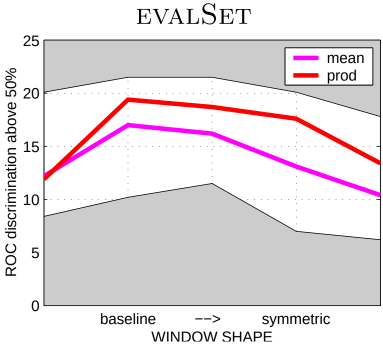
Experiment 1

- keep window maxima a constant t_{sep} apart
- less asymmetry $\leftrightarrow$ more window support overlap



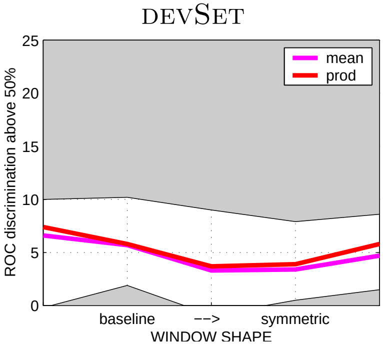
Experiment 1: Results

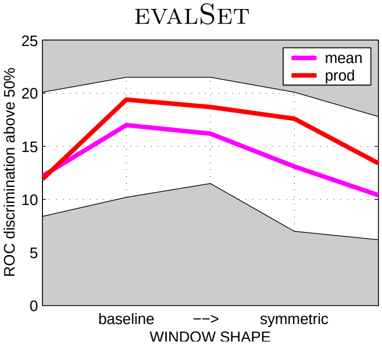




- symmetric windows appear to lead to:
  - lower ROC discrimination than baseline, in all cases

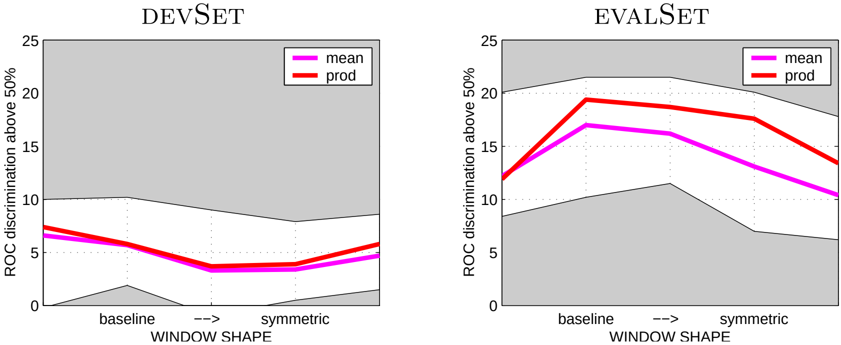
Experiment 1: Results





- symmetric windows appear to lead to:
  - lower ROC discrimination than baseline, in all cases

Experiment 1: Results



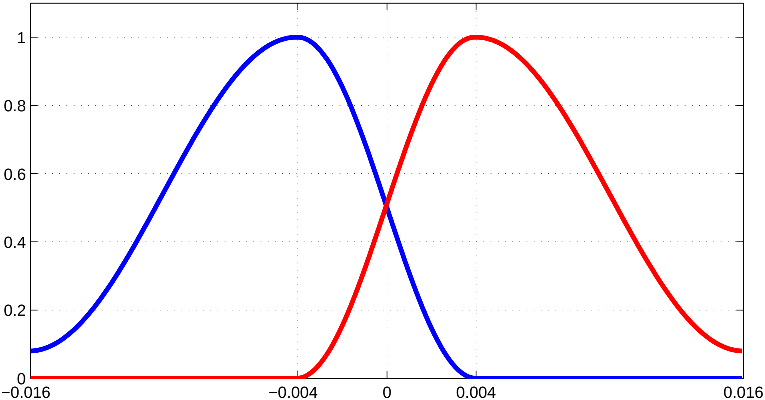
- symmetric windows appear to lead to:
  - lower ROC discrimination than baseline, in all cases

Experiment 2

- keep window support overlap constant
- less asymmetry ↔ window maxima further apart

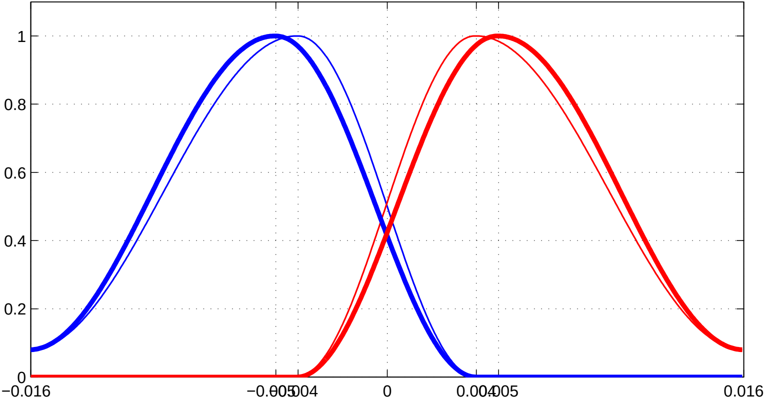
Experiment 2

- keep window support overlap constant
- less asymmetry ↔ window maxima further apart



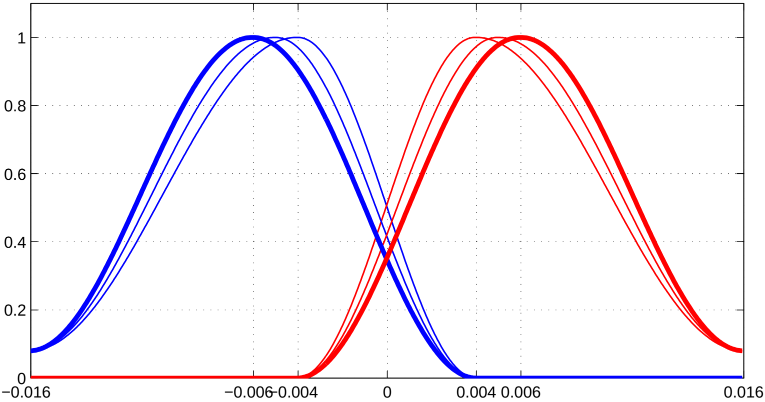
Experiment 2

- keep window support overlap constant
- less asymmetry ↔ window maxima further apart



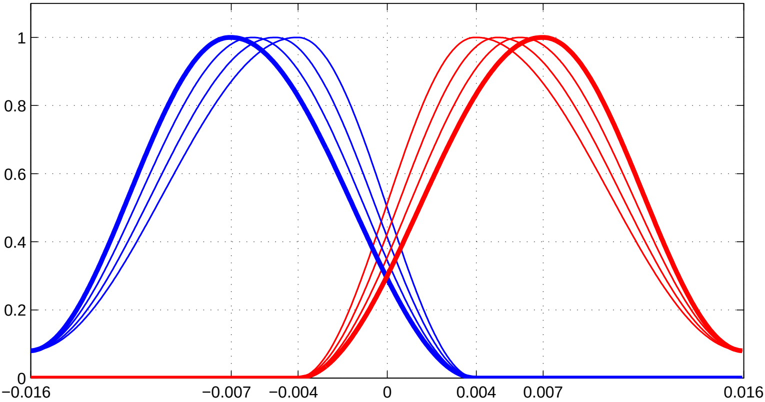
Experiment 2

- keep window support overlap constant
- less asymmetry ↔ window maxima further apart

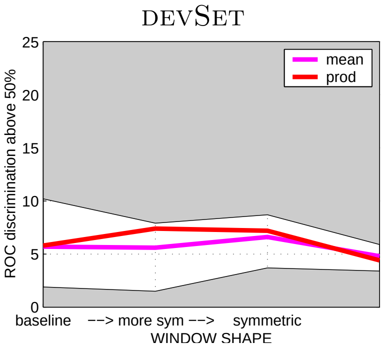


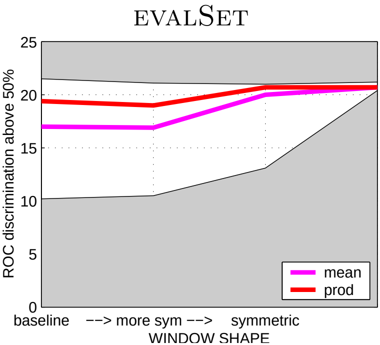
Experiment 2

- keep window support overlap constant
- less asymmetry ↔ window maxima further apart



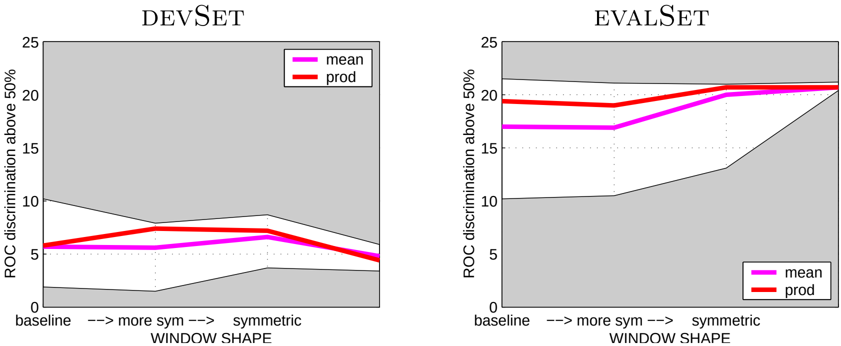
Experiment 2: Results





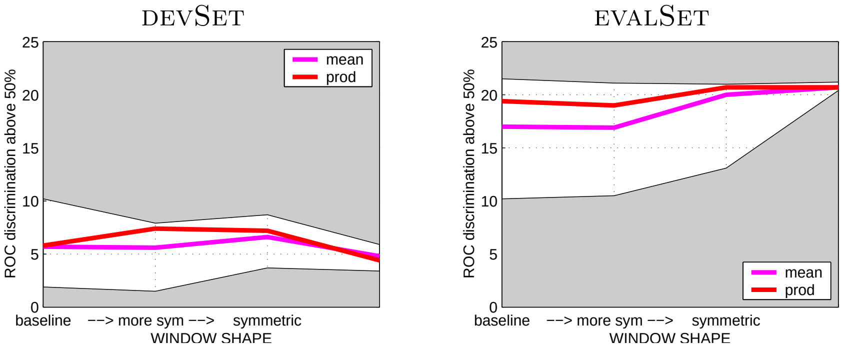
- symmetric windows appear to lead to:
  - higher ROC discrimination than baseline, in all cases
  - smaller variability between best and worst partitions

Experiment 2: Results



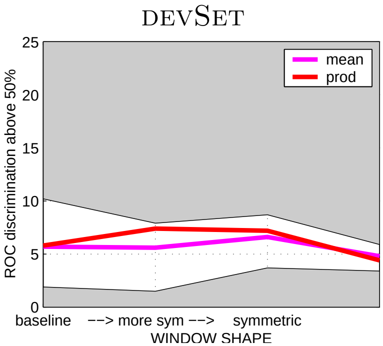
- symmetric windows appear to lead to:
  - higher ROC discrimination than baseline, in all cases
  - smaller variability between best and worst partitions

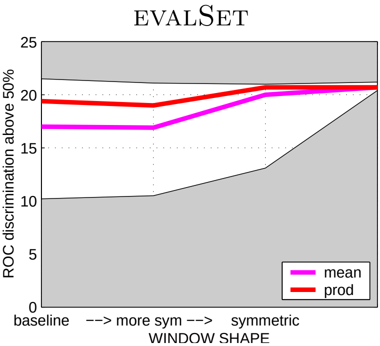
Experiment 2: Results



- symmetric windows appear to lead to:
  - higher ROC discrimination than baseline, in all cases
  - smaller variability between best and worst partitions

Experiment 2: Results





- symmetric windows appear to lead to:
  - higher ROC discrimination than baseline, in all cases
  - smaller variability between best and worst partitions

Conclusions

- t_{sep}: separation between window maxima
- t_{fra}: duration of analysis frame

1. when $t_{sep} > \frac{1}{3}t_{fra}$, **symmetric-support windows appear best**
2. when $t_{sep} < \frac{1}{3}t_{fra}$, first priority should be **to limit overlap in support to a maximum of t_{sep}** at the expense of symmetry if necessary
3. results suggest that better ROC discrimination may be possible when symmetric-support windows are placed even further apart in time than tried here

Thanks for attending.

(kornel@cs.cmu.edu)