Hunting the Wild Feature Vector: Understanding English Prosody

Greg Kochanski
Oxford University

This talk was given at Google Pittsburgh Labs, July, 2006. It is based primarily on http://kochanski.org/gpk/papers/2005/04pnp.pdf, by G. Kochanski, E. Grabe, J. Coleman and B. Rosner, http://scitation.aip.org/jasa/ 118(2), August 2005, pages 1038-1054.

Googling®? speech is good, but ∃ serious challenges.

- Copyright & DRM
- English has dialects
- Lots of speech is mixed with music & noise.
- ASR gives words, only.

I did not eat the dog.
I did not eat the dog.
I did not eat the dog.
I did not eat the dog.

FOCUS

Speech recognition systems can give you the words but not the prosody. An important part of the prosody in many languages is "focus", which can dramatically change the meaning of a sentence by emphasizing different

Prosody

Prosody describes acoustic properties that are not deducible from the local text. Punctuation is prosody.

Read this as “Prosody describes acoustic? Properties that are not **deducible** from the. Local text punctuation? **Is** prosody.”

Clearly, by messing with the prosody, you can seriously disturb the intelligibility of English. Thus, it carries important information; some of the prosody appears in written form as punctuation, but some is left to the reader.

As a side note, one can think of producing a play as an exercise in finding the correct prosody for the text.

What do we know?

* Strong intuition that prosody is important
  -but- difficulties naming and classifying

* Intuition that prosody has a lot to do with pitch
  [esp. for English questions]
  -and-
  Psychophysical experiments show
  pitch ≈ fundamental frequency of speech
  -but-
  fundamental frequency is not well correlated with anything [but some with questions]

∴ Our understanding of how prosody is expressed is poor.

Machine Learning

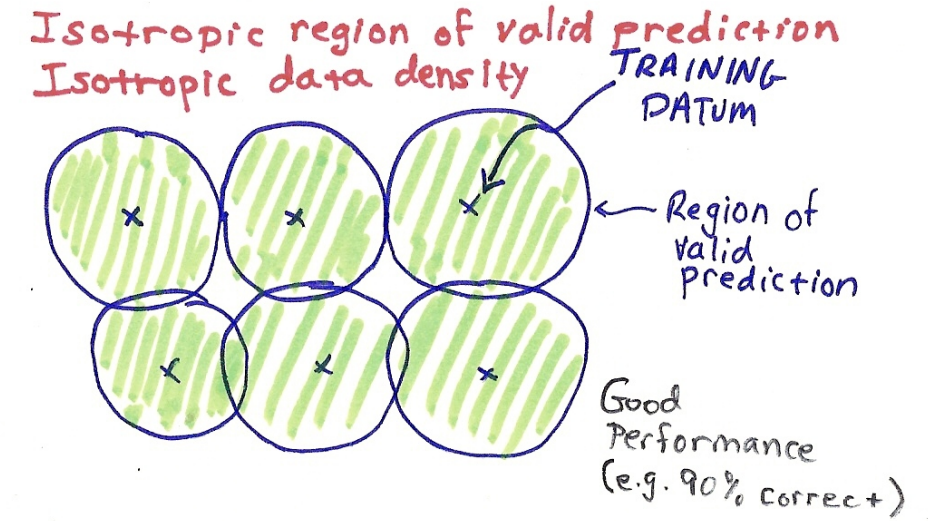
- Data is limited for supervised learning
  - Humans are bad at conscious classification & slow
  - Agreement between humans is ~80%
  - Biggest existing databases contain few $\times 10^3$ items

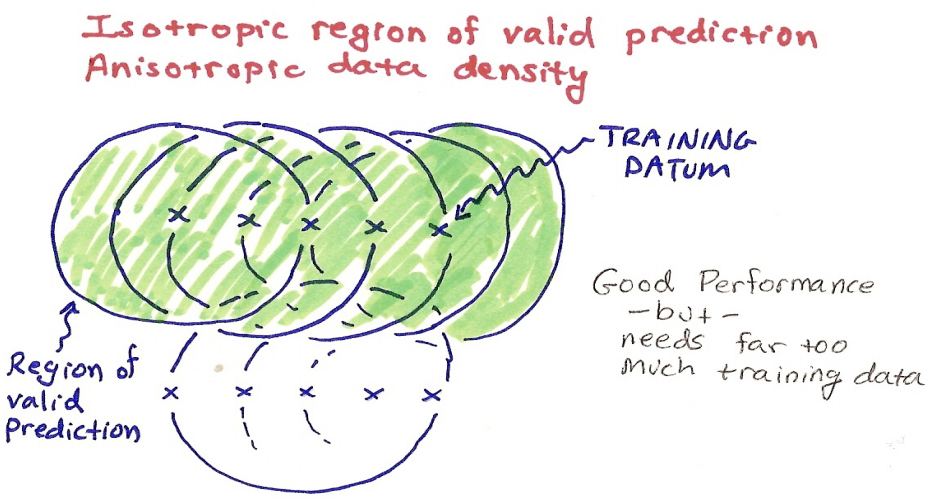
- Problem not suitable for unsupervised techniques
  - Zipf's law: long tail → usup splits common items
  - Research shows lack of distinct clusters
  - Result of unsupervised classification depends on feature rec, and we don't know it well
  - ∴ unsup-ervised results are unrelated to English

An important thing to note is that most machine learning systems operate in high dimensional spaces. In speech processing, things tend to happen in N=20 or N=39 dimensional spaces, for instance.

That high dimension makes the match between the feature vector and the problem extremely important.

For instance, if the feature vector is poorly chosen, it's easily possible to (as in the lower figure) have the data too dense in a few dimensions. And, if it's 3x too dense in each of 10 dimensions, that means you will need 3^{10} times more data than the optimal arrangement. This can be a rather large factor, and can have a huge impact on the cost and practicality of making a machine learning system work.





* The definition of the feature vector controls where the data lie.

Loosely speaking, most machine learning systems can be considered to be schemes for interpolating from a set of data to a point where you want the answer.

Typically, if you inquire at a point very close to training data, you'll get the right answer; if you inquire far enough away, you'll get some random, probably incorrect result. So, you can think of each training datum as lying in the center of a region of valid prediction (green circles here). The overall accuracy of the machine learning system is (approximately) given by the fraction of space covered by the green circles. In the top figure, the accuracy might be about 80%.

If you want the greatest accuracy, you want to densely cover space with the regions of valid prediction, as seen in the lower figure. Then, no matter where you inquire, you'll always be close to a training datum, and you should get the correct answer.

However, if data is scarce (and it often is), you want to arrange for space to be covered by regions of valid prediction, but not more densely than necessary. You don't want the regions to overlap too much, because any overlap increases the required amount of data.

Now, the region of valid prediction is set by the match between the actual system you are trying to predict and the details of the machine learning algorithm. However, the location and density of the training data is set by the amount of data you collect and the definition you choose for your feature vector.

Consequently, you need to choose the right feature vector to get good prediction accuracy with limited data.

What do we know about the Feature Vector?

What do we know about the feature vector for speech prosody?

- 1 or 2 components per syllable
  - X
  - Several acoustic properties
  - X
  - ~10 syllables per sentence

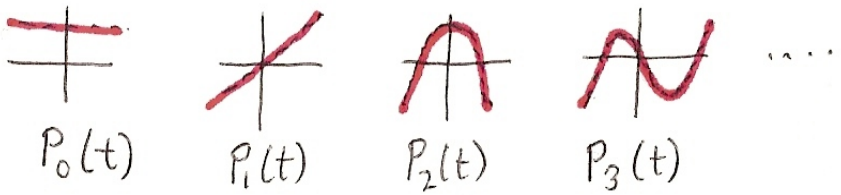
- All properties are time series
- Shapes matter
- Values don't (much)
- Generally continuous [from muscles]
- Probably local.

The actual values of most acoustic properties doesn't matter much to prosody. A shout still sounds like a shout, even if it comes from far away and thus has a low amplitude. Likewise, males and females can execute the same prosodic effects, even though the average female speaker has a pitch an octave higher than a typical male speaker.

The speakers muscles move smoothly and continuously, so you expect most of the acoustic properties of speech to change smoothly and continuously.

- We'll use orthogonal polynomials in window
  - compare focus syllables vs.
  - non-focal

Orthogonal Polynomials



assume

$$data(t) = C_0 P_0(t) + C_1 P_1(t) + C_2 P_2(t) + \ldots + error$$

FIND:

$C_0, C_1, C_2, C_3, \ldots$ to minimize error.

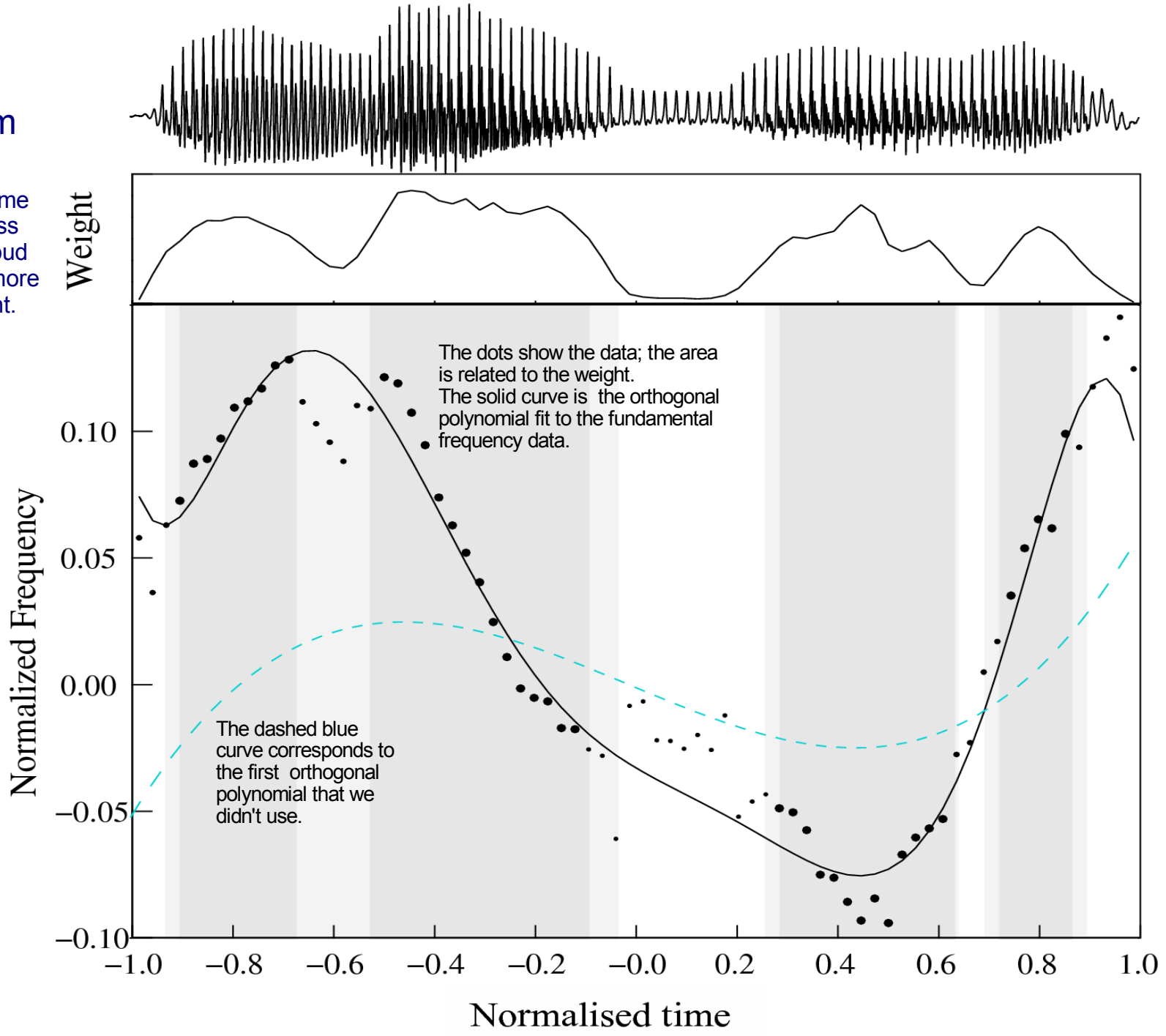
- Describes shape over time.
- Ignore P_0, then values don't matter.
- Always continuous.
- Local if done in window.

Orthogonal polynomials allow you to boil a complicated fundamental frequency contour down to a small handful of coefficients.

Fitting fundamental frequency data with orthogonal polynomials.

Speech waveform

Not all the data is equally valuable: some parts are more or less periodic, and then loud parts are probably more perceptuall important.



The Feature Vector

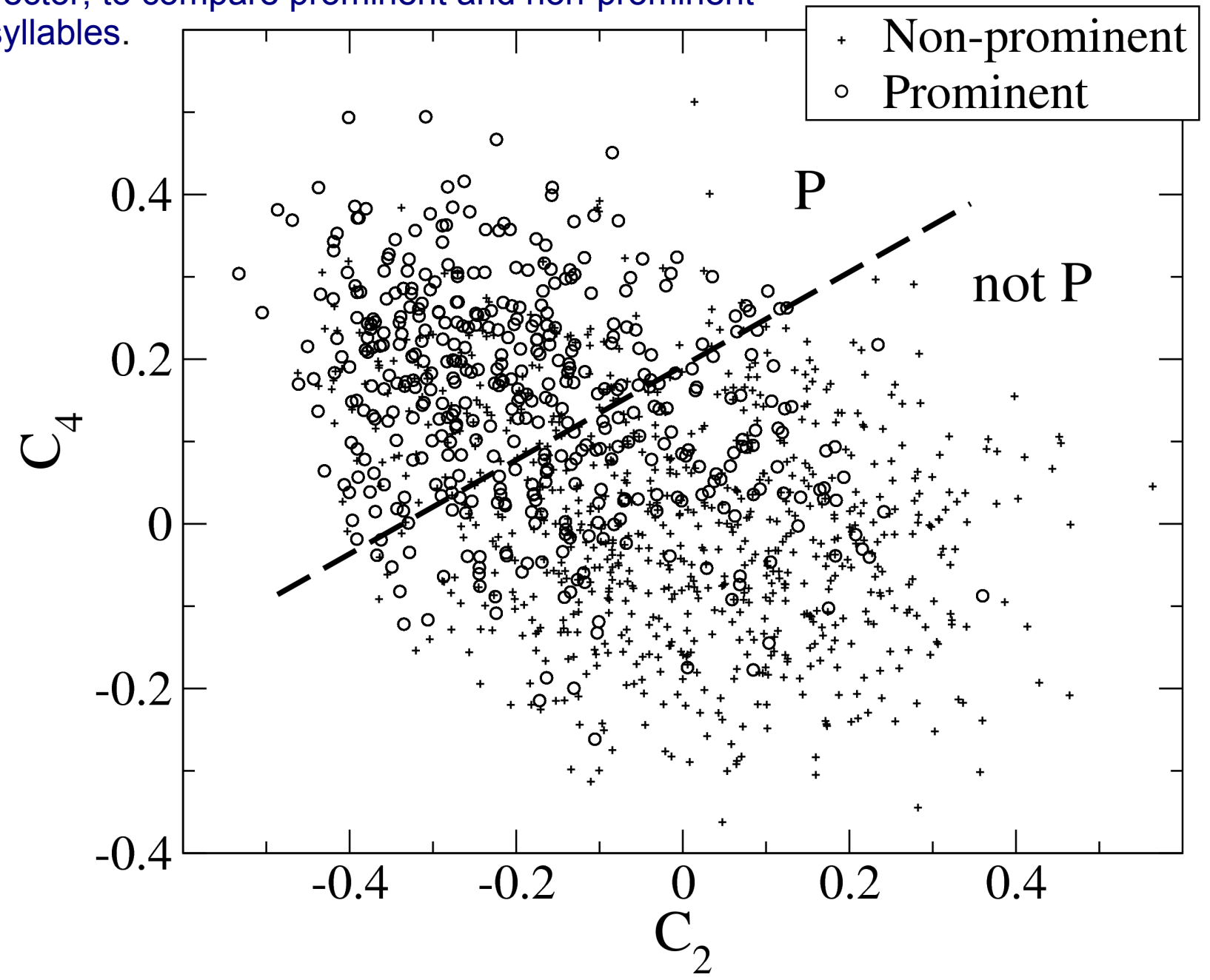
- Window ~2 syllables long [500 ms]
- Centered around human accent mark
- $C_0 \ldots C_4 \ldots C_7$ orthogonal poly coefs.

  X

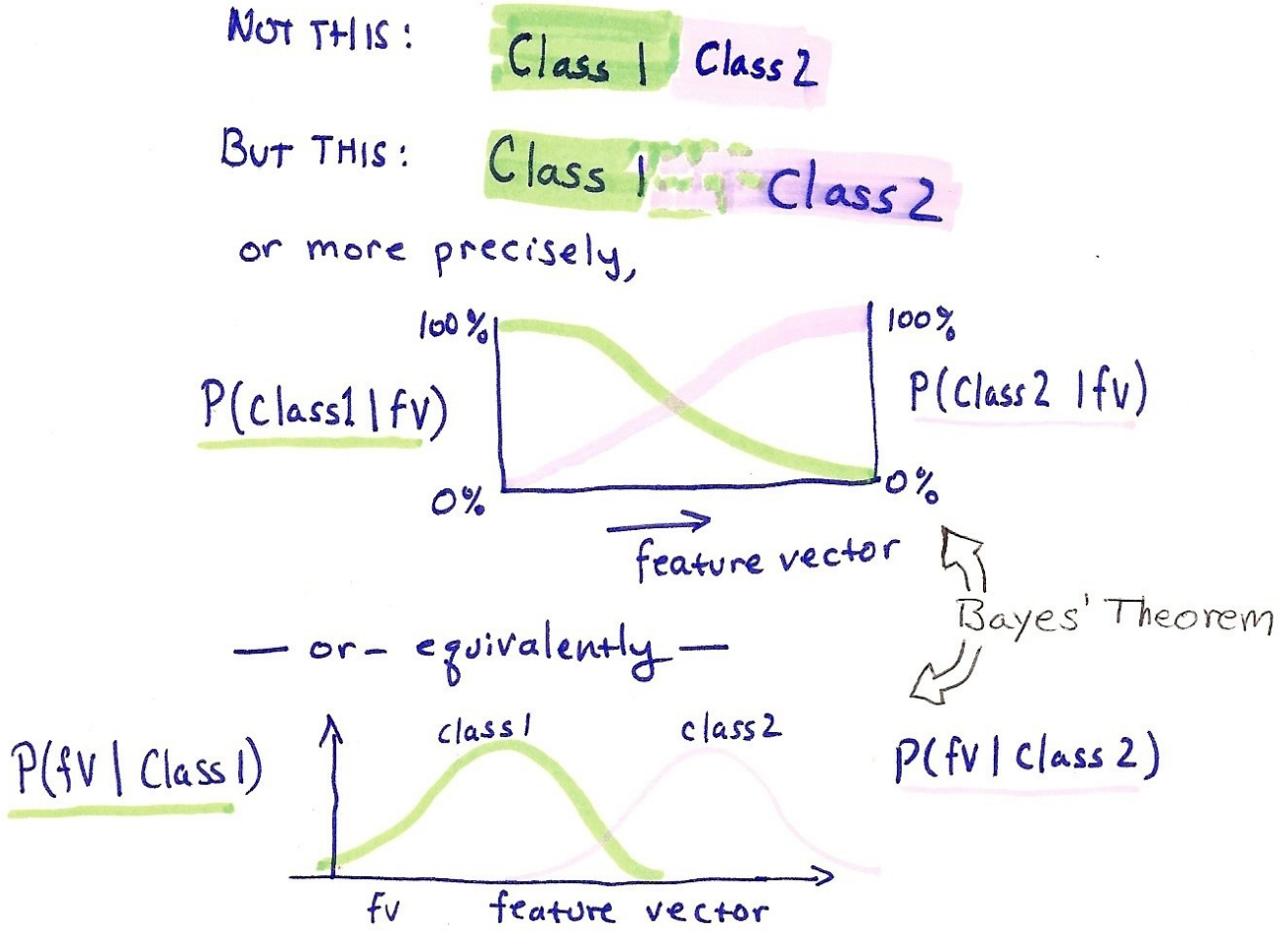
  { loudness, fundamental freq, local speech rate,
  spectral slope, voicing }

  ~25 components, depending on window size

This plot shows two components of the feature vector, to compare prominent and non-prominent syllables.



Classifier: Probabilistic & Bayesean



For the purpose of this talk, we use a Bayesean probabilistic classifier. This lets one accurately describe situations where the classes overlap.

Bayes' Theorem

TRAINING:

The "true" class

$$P(\alpha \mid fV, C) = \frac{P(fV \mid \alpha, C) \cdot P(\alpha)}{\int P(fV \mid \alpha, C) \cdot P(\alpha) \cdot d^N\alpha}$$

The observed feature vector

USE:

A "good" set of class boundaries

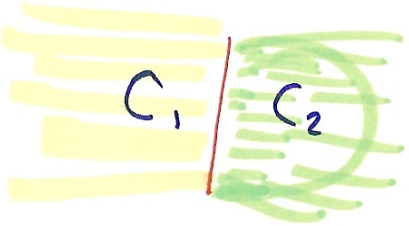
$$P(C \mid fV, \alpha) = \frac{P(fV \mid C, \alpha) \cdot P(C)}{P(fV \mid C=1, \alpha) \cdot P(C=1) + P(fV \mid C=2, \alpha) \cdot P(C=2)}$$

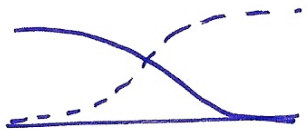
fV is the feature vector,
C is the class, and
α contains the parameters that define the classes and class boundaries.

So, what is $P(fv \mid \alpha, C)$?

Logistic:

$$P(fv \mid \alpha, C_1) = \prod_{\text{all data}} \frac{e^{(\vec{fv} - \vec{a}_c) \cdot \vec{b}_c}}{e^{(\vec{fv} - \vec{a}_0) \cdot \vec{b}_0} + e^{(\vec{fv} - \vec{a}_1) \cdot \vec{b}_1}}$$

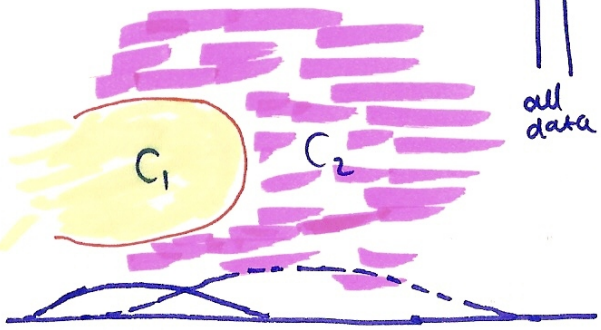




Two different kinds of classifiers:
Logistic gives you linear class boundaries,
Gaussian give you class boundaries that are conic sections (elipses and parabolas and hyperbolas).

Gaussian:

$$P(fv \mid \alpha, C_1) = \prod_{\text{all data}} \frac{e^{(\vec{fv} - \vec{a}_c) \cdot \overleftrightarrow{B}_c \cdot (\vec{fv} - \vec{a}_c) + d_c}}{e^{(\vec{fv} - \vec{a}_0) \cdot \overleftrightarrow{B}_0 \cdot (\vec{fv} - \vec{a}_0) + d_0} + e^{(\vec{fv} - \vec{a}_1) \cdot \overleftrightarrow{B}_1 \cdot (\vec{fv} - \vec{a}_1) + d_1}}$$



So how to train the classifier?

Almost: find $\hat{\alpha}$ to maximize

$$P(C = \text{"true" class} \mid fV, \alpha)$$

Why NOT?

- If # of parameters in α gets close to # of data, then $\exists$ many good α.
- If you have 25 data in 26-space, they all lie on a plane. $\Rightarrow$ TROUBLE

Really: Take $P(\alpha \mid fV, C)$ and sample values of α from that distribution.

We don't do a maximum-likelyhood training, because it breaks badly when data gets scarce. Instead, we compute the probability distribution of classifier parameters (α) and sample values of α from that distribution.

This behaves well when data is scarce, giving you all reasonably good ways of splitting the two data sets. In one-dimension, it reproduces Student's t-distribution (as it should).

Markov Chain Monte Carlo

Theorem: If you can compute $P(x)/P(y)$, then you can generate samples from $P(x)$.

At first glance, this seems to be a useless theorem. However, it turns out that *P(x)* and *P(y)* share the same nasty, hard-to-compute denominator. So, the ratio is much easier and faster to compute than either one individually.

Relevance: Recall that

$$P(\alpha \mid fv, G) = \frac{P(fv \mid \alpha, G)\ P(fv)}{\int P(fv \mid \alpha, G) \cdot P(fv)\ d^N fv}$$

NASTY →

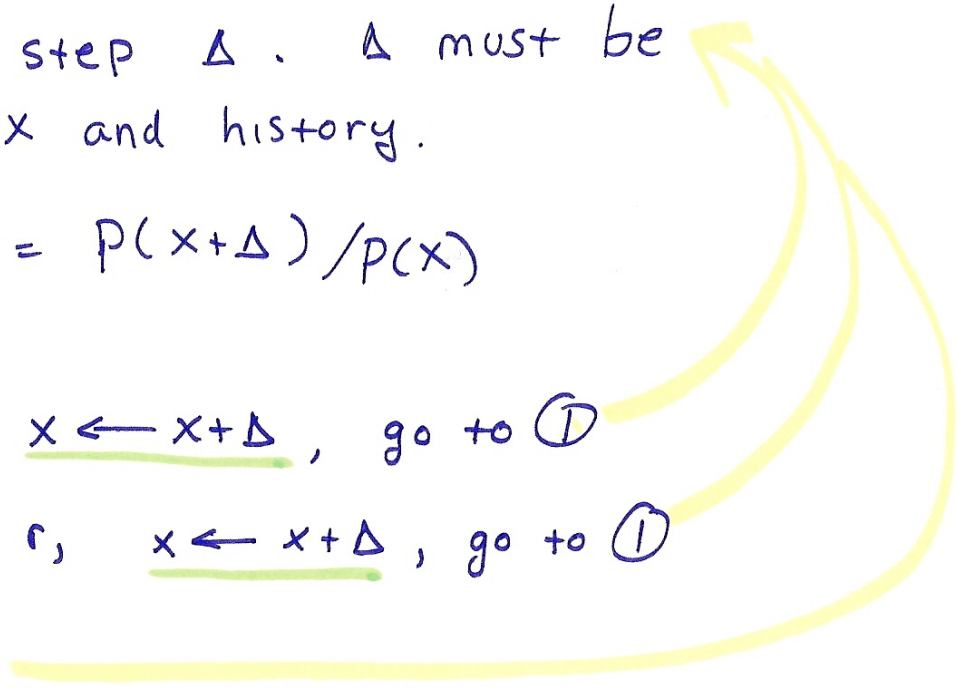
The integral in the denominator is over the space in which the feature vector lives. In this case, it's about a 5-dimensional integral. Multidimensional integrals are generally expensive and hard and slow to compute.

Markov Chain Monte Carlo:

How?

To sample from $P(x)$:

1. Pick a random step Δ. Δ must be independent of x and history.
2. compute $r = P(x+\Delta)/P(x)$
3. If $r > 1$, $x \leftarrow x+\Delta$, go to 1
4. with probability r, $x \leftarrow x+\Delta$, go to 1
5. Else go to 1



Result of MCMC:

Given a training set of labelled data, we build 90 classifiers (each separated by ~1000 mc mc steps).

They're separated by 1000 Markov steps so that the samples are relatively independent of one another.

Each of 90 classifiers is stored, so we can look at consistency of class boundaries.

Each is tested and we compute

$$K = \frac{\text{correct} - \text{chance}}{1 - \text{chance}}$$

K tells you how much better the classifier runs, relative to how well you could do without access to acoustic data. In other words, if you get K=0, you could do just as well by guessing with your eyes shut. K=1 corresponds to perfect classification.

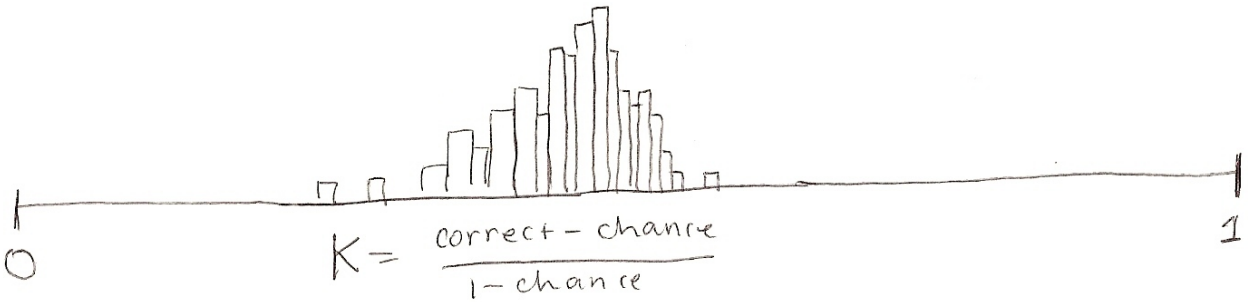
Q: But, what's special about the Test Set?

A: Nothing!

Repeat the whole process with 12 random splits of data into training/test set.

(...and, we replicate some of the data with different human labelers...)

Result: $12 \times 90 = 1080$ classifiers, all different, all consistent with the data

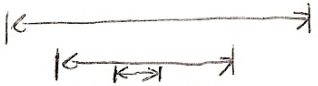


Now we can do some science

- Compare classifiers under different conditions.

  Better? Worse? Can't tell?

× Different window sizes



I am **NOT** a dog.

× Is the class boundary simple or complicated

Quadratic vs. linear

× Different acoustic properties

- fundamental frequency
- loudness
- voiced vs unvoiced
- local speaking rate / duration
- spectral slope / timbre

Here we built classifiers using different acoustic properties, and looking at different regions around the syllable centers.

Classifiers based upon loudness worked best. The best window size seems to be about a half-second, suggesting that the prominence information is spread out over about a quarter-second on either side of the syllable center. This means that the loudness contrasts between a syllable and its neighbors are important.

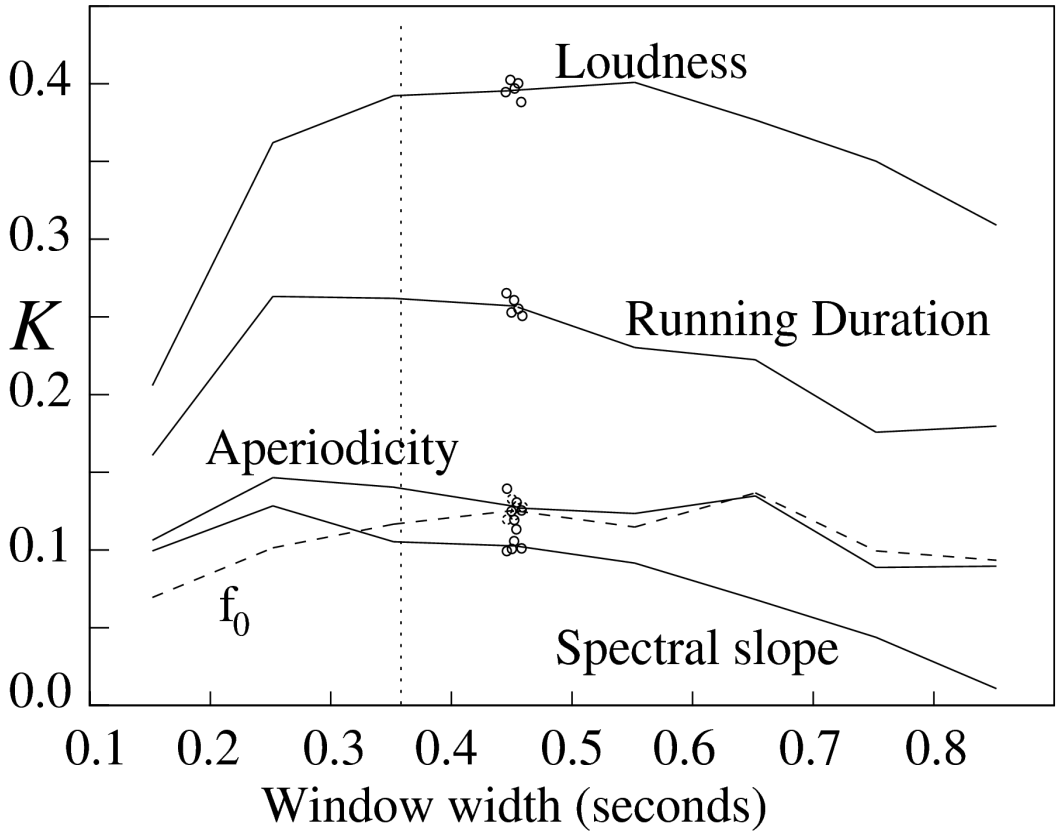


FIG. 4. Classifier performance *vs.* the size of the analysis window, w. Each curve shows performance of classifiers based on a different acoustic feature (f_0 is shown dashed to separate it from its neighbors). The vertical axis is the K-value, which shows how well each classifier performs relative to chance (shown as zero) and exact duplication of the human labels (shown as one). Plotted K-values are averages over seven dialects and three styles of speech. The vertical dotted line marks where the window includes neighboring syllable centers. The small clusters of points near $w = 0.45$ s show the reproducibility of the classifiers, derived from five classifier runs with slightly different window sizes.

We see generally consistent results across all seven dialects.

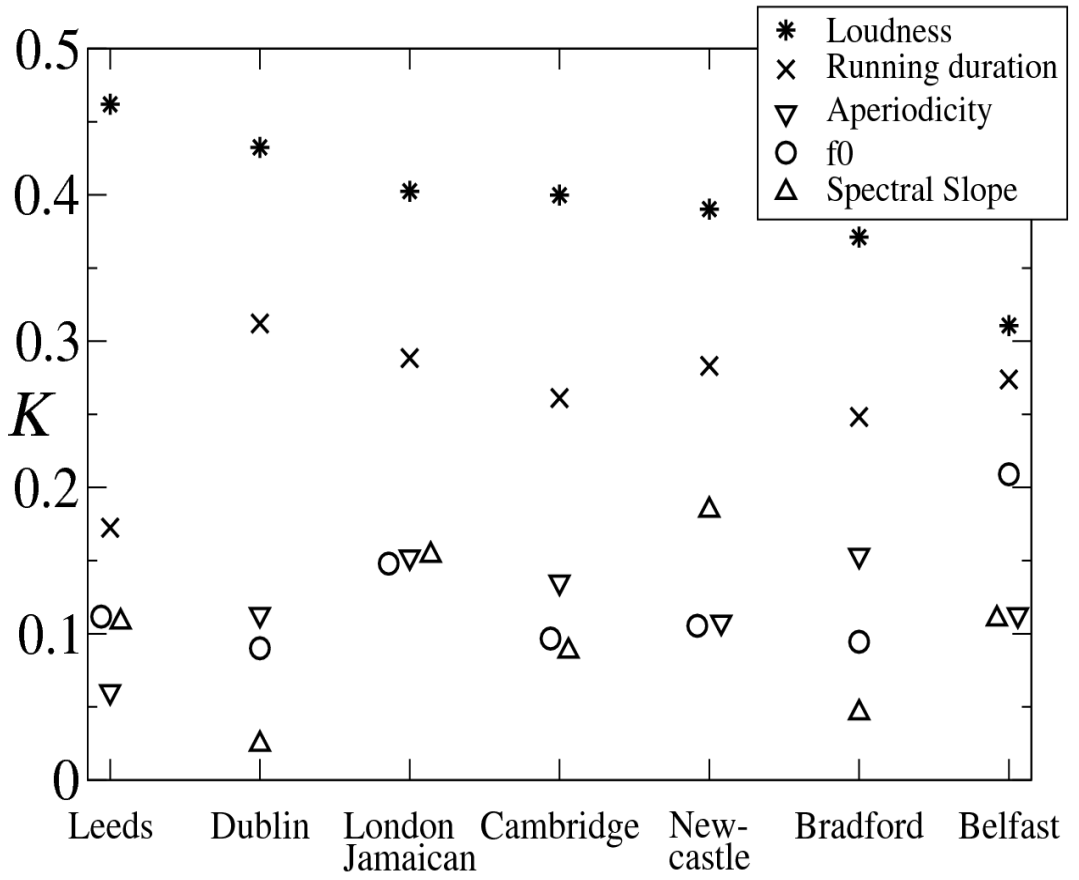


FIG. 7. Classifier performance for the five acoustic measures as a function of dialect. Each classifier is trained on a single dialect/style combination; symbols show the average over the three styles of speech.

We can reverse the analysis procedure that gave us our orthogonal polynomial coefficients, and reconstruct what the loudness profile would be for a datum in the center of each class.

This figure shows the contrast between prominent and non-prominent syllables. Prominent syllables have a substantial loudness bump.

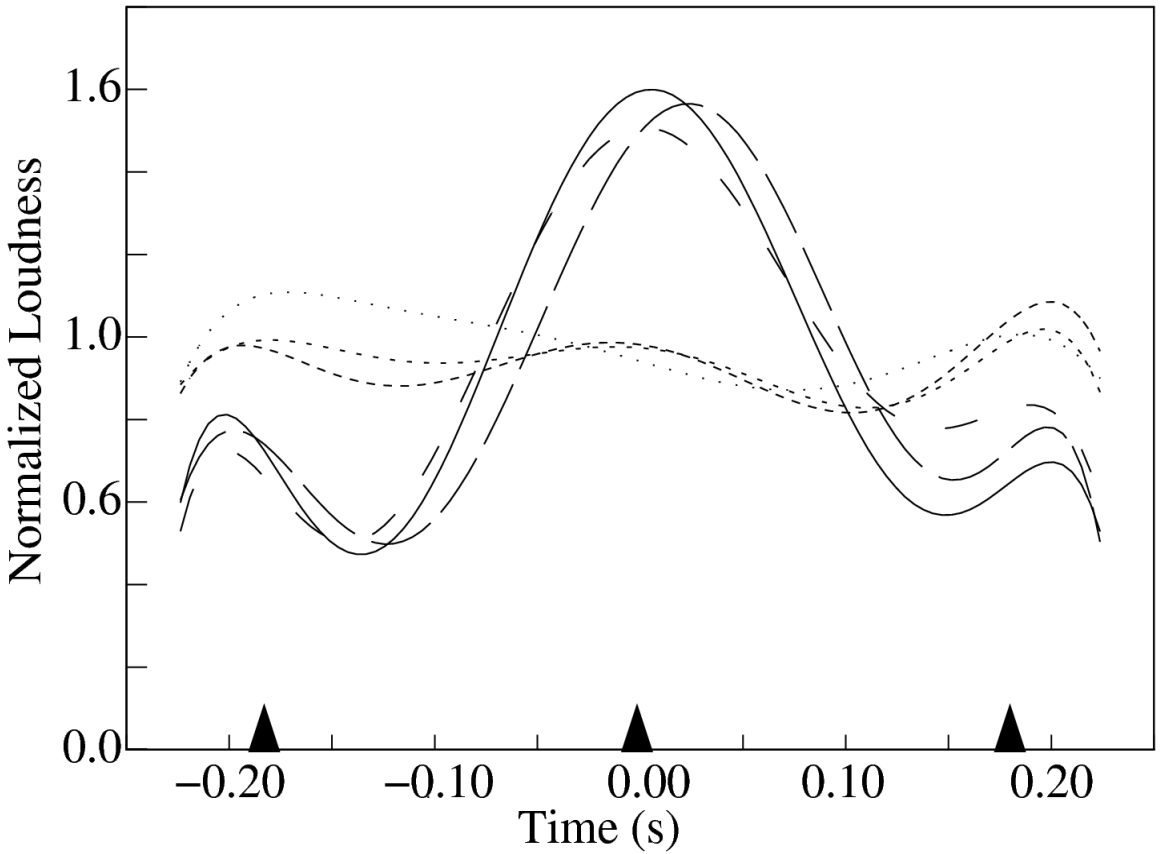


FIG. 8. Reconstructed loudness profiles for prominent (long dashes) and non-prominent (short dashes) syllables, for the primary and two secondary data sets. In each group, the primary data set is plotted with the most ink, followed by secondary sets GK then EL. The black triangles mark the median position of syllable centers. Zero on the time axis corresponds to the prominence mark.

Local Conclusions

- Reasonably good performance

  $K \sim 0.3$

  Human-Human agreement corresponds to

  $K \sim 0.6$

- Performance is likely data-limited
  (More complex classifiers that combine different acoustic properties could not be attempted.)

- The scope of prominence is 2 syllables (or perhaps more).

- Loudness & duration contrasts are important
- pitch is not too important.
- Different dialects had similar definitions of prominence.

But,

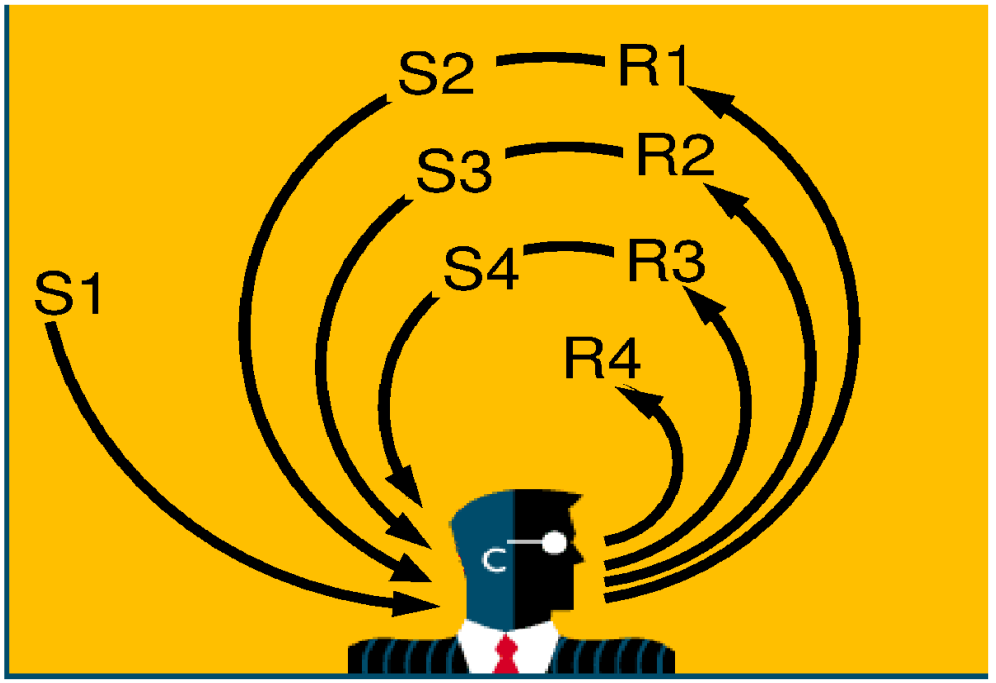
more recent experiments may point toward a better feature vector

Mimicry

This part of the talk is taken from The http://scitation.aip.org/jasa/ -- June 2006 -- Volume 119, Issue 6, pp. 4006-4015 " http://kochanski.org/gpk/papers/2005/2005mimicry." by Bettina Braun, http://kochanski.org/gpki, Esther Grabe, and Burton S. Rosner

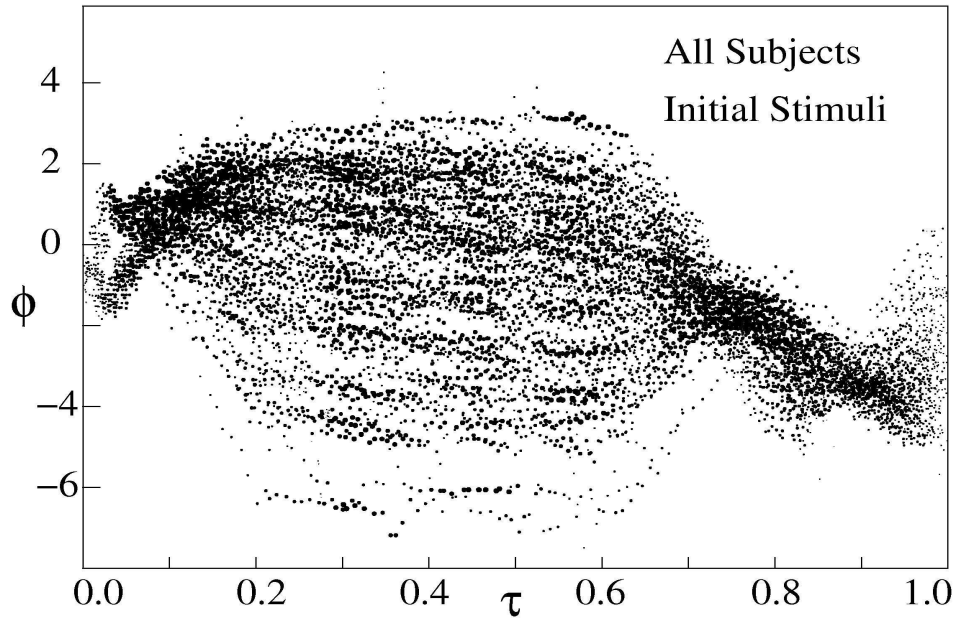
METHOD: ITERATIVE MIMICRY

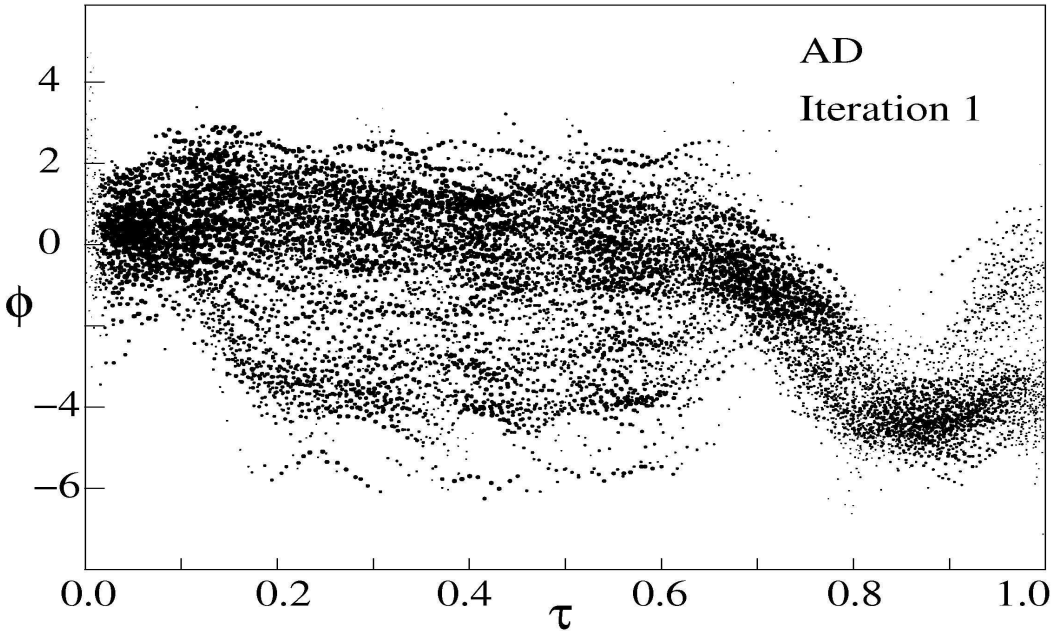
STIMULI



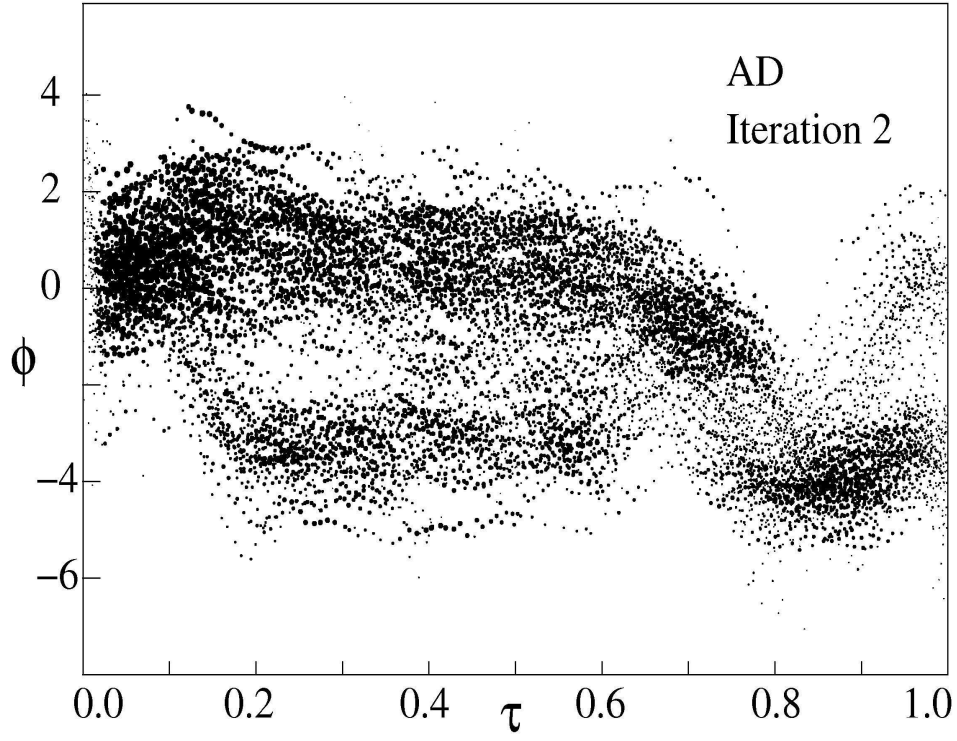
IMITATIONS (RESPONSES

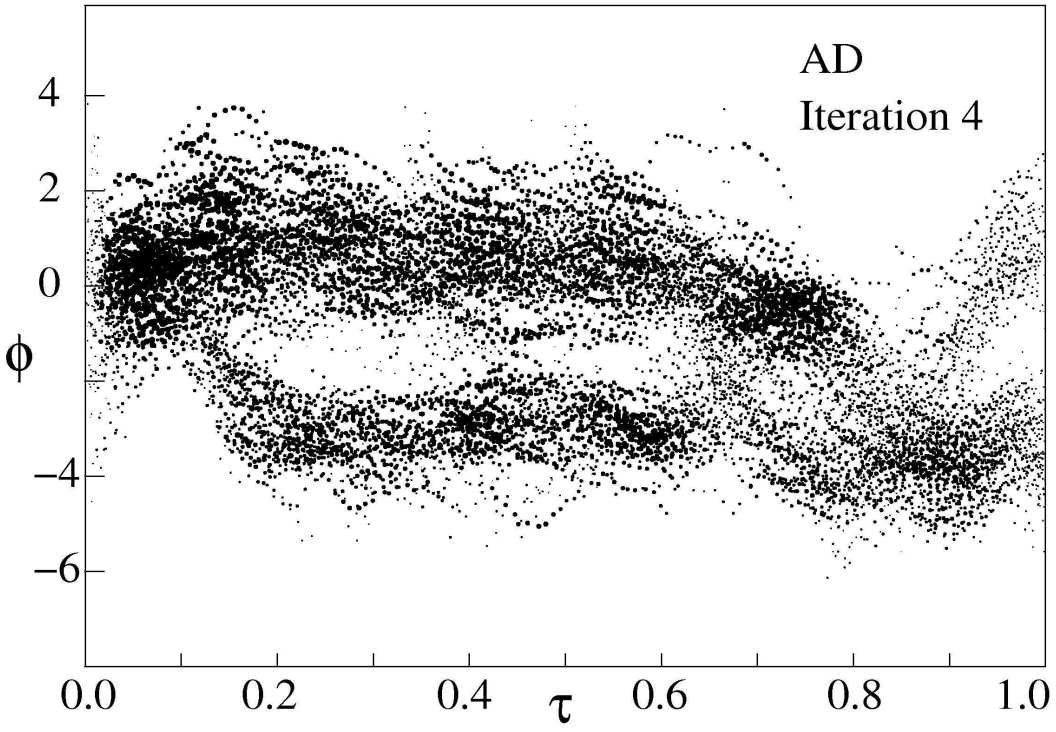
- Ten naïve speakers of Southern British English, four sessions each.
- 120 initial utterances were made, each resynthesized with a unique, physiologically possible, f_0 contour. Sentences were about 1.2 seconds long.
- In session 1, subjects mimicked each initial utterance. They heard S1 and produced R1. Each production was trimmed, amplitude-normalised, and stored.
- In sessions 2 through 4, the subject mimicked his/her own productions from the preceding session. (S2 →R2, ...)
- The f_0 track of each utterance obtained with ESPS get_f0. Each track was time-normalized.





These plots show fundamental frequency data from 115 utterances. In the initial stimuli, the distribution of f0 in the middle of the utterance starts uniform, but as people mimic themselves, the f0 contours gradually approach two preferred branches.





More Conclusions

- No evidence that intonation is made of small, localized chunks.
  - Maybe it's not local?
- Seems to be high vs. low.
  - linear additivity in orthogonal polynomials may not be a good approximation?
- Convergence to the attractors was slow.
  - Attractors are deduced, not observed.
  - There are no hard categories, probably.

∴ The search for the perfect feature vector continues....