Dimensionality Reduction II: ICA

Sam Norman-Haignere

Jan 21, 2016

Motivation

- Many signals reflect linear mixtures of multiple ‘sources’ :

  $\Longrightarrow$ Audio signals from multiple speakers

  $\Longrightarrow$ Neuroimaging measures of neural activity

  (e.g. EEG, fMRI, calcium imaging)

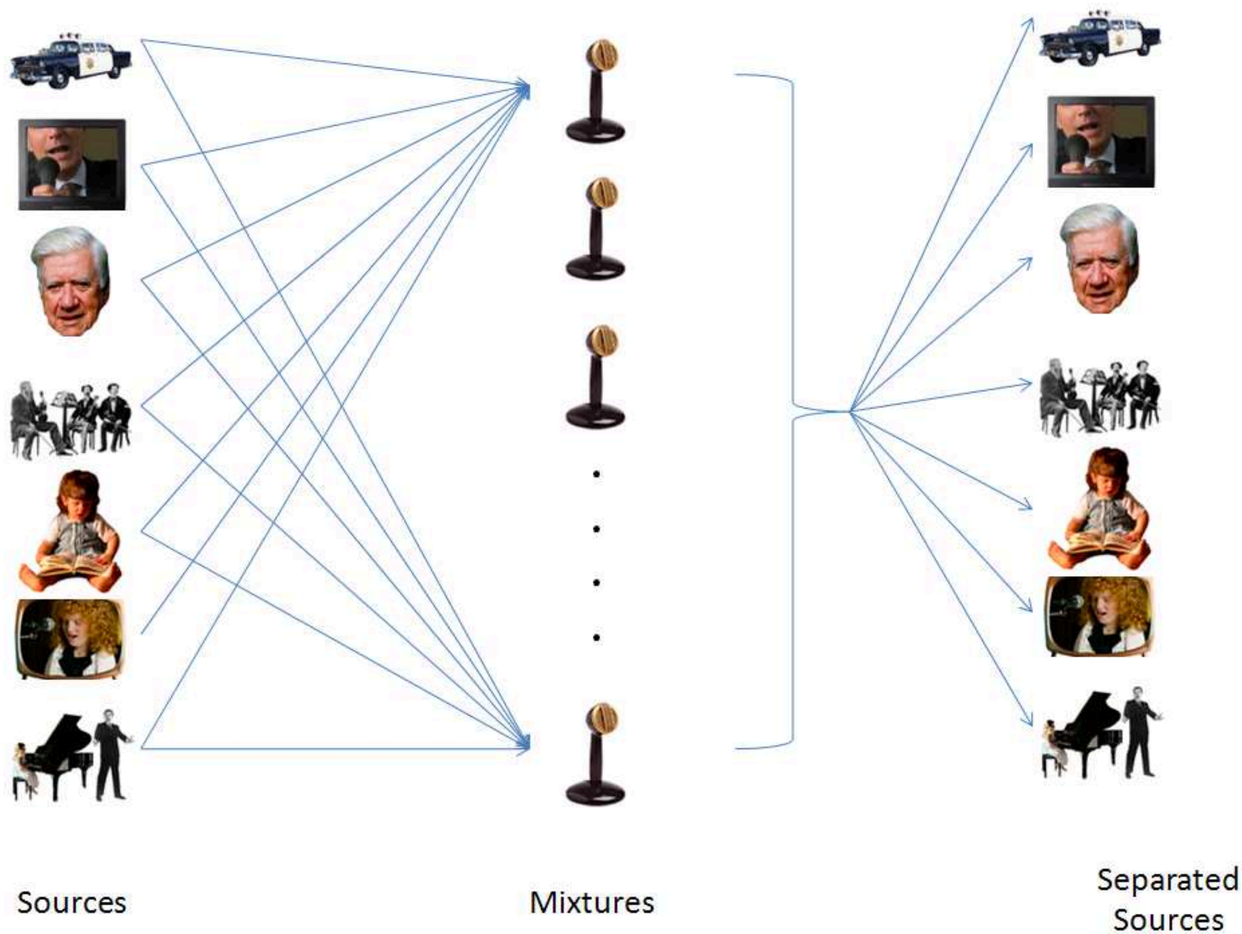
- Often want to recover underlying sources from the mixed signal

  $\Longrightarrow$ ICA algorithms provide general-purpose statistical machinery

  (given certain key assumptions)

Classic Example: Cocktail Party Problem

- Several sounds being played simultaneously
- Microphones at different locations record the mixed signal



Classic Example: Cocktail Party Problem

- Several sounds being played simultaneously
- Microphones at different locations record the mixed signal

ICA can recovers individual sounds sources

$\Longrightarrow$ Only true if # microphones $\geq$ # sources

http://research.ics.aalto.fi/ica/cocktail/cocktail_en.cgi

Classic Example: Cocktail Party Problem

Why is this a classic demo?

⇒Impressive

⇒Assumptions of ICA fit the problem well:

1. Sound source waveforms close to independent
2. Audio mixing truly linear
3. Sound source waveforms have non-Gaussian amplitude distribution

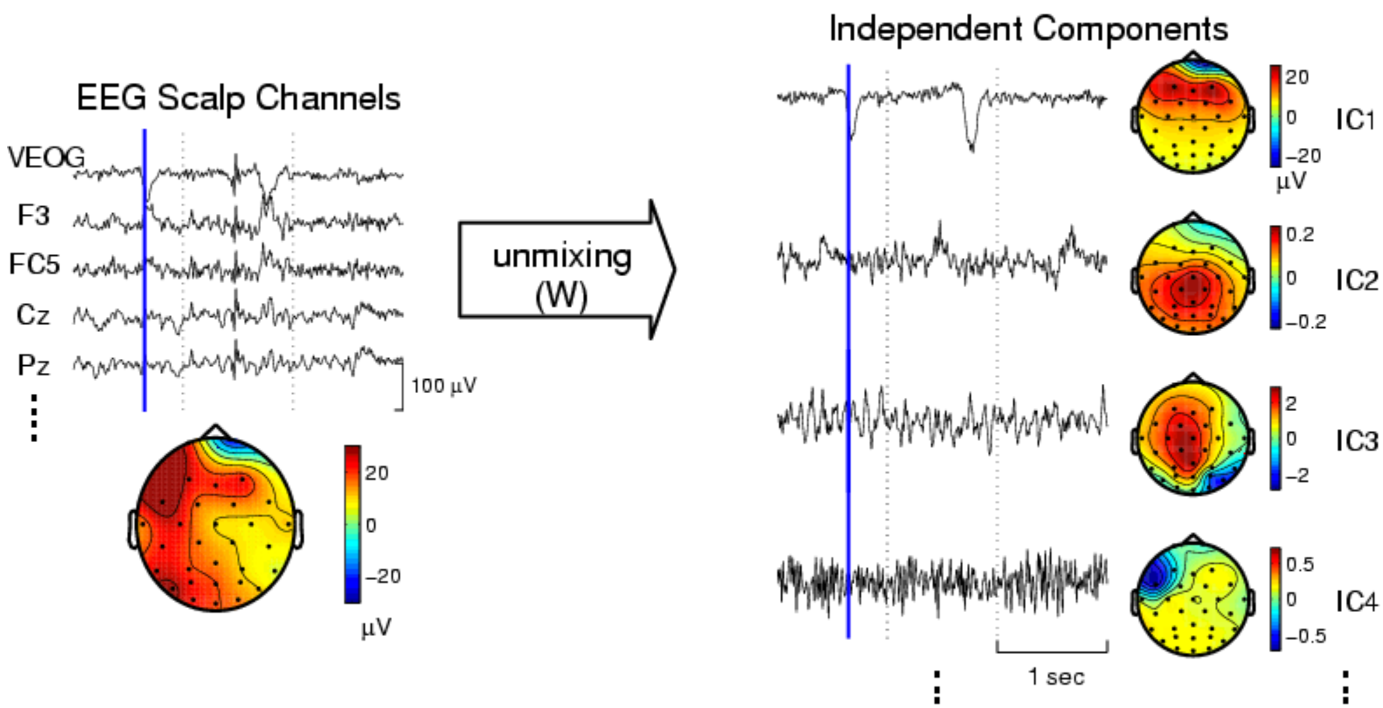
HISTOGRAM OF SPEECH WITH GAUSSIAN OVERLAID

Neuroimaging Examples: EEG

Frequently used to denoise EEG timecourses

⇒Artifacts (e.g. eye blinks) mostly independent of neural activity and have non-Gaussian amplitude distribution

⇒EEG channels modeled as linear mixture of artifacts and neural activity



Neuroimaging Examples: fMRI

fMRI ‘voxels’ contain hundreds of thousands of neurons

$\Longrightarrow$ Plausibly contain neural populations with distinct selectivity

fMRI responses (reflecting blood) approximately linear function of neural activity

$\Longrightarrow$ Use component analysis to unmix responses from different neural populations?

Working Example from My Research (for Convenience)

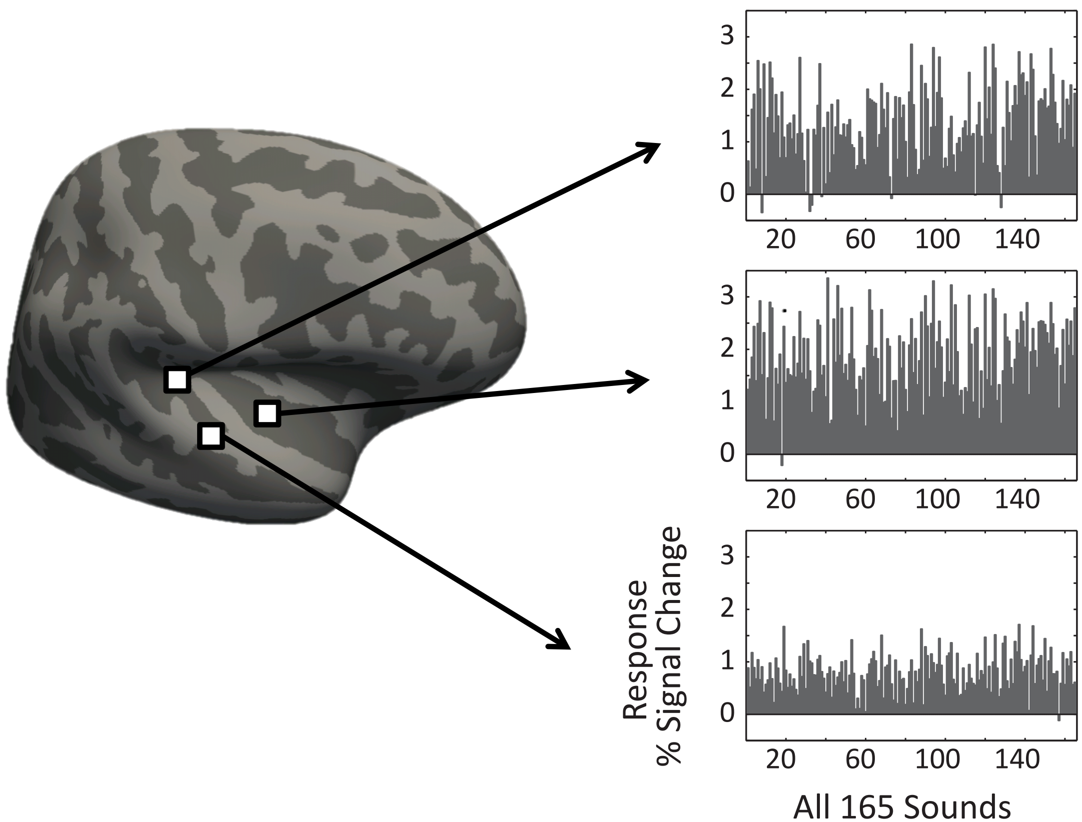
- Measured fMRI responses to 165 natural sounds:

1. Man speaking
2. Flushing toilet
3. Pouring liquid
4. Tooth-brushing
5. Woman speaking
6. Car accelerating
7. Biting and chewing
8. Laughing
9. Typing
10. Car engine starting
11. Running water
12. Breathing
13. Keys jangling
14. Dishes clanking
15. Ringtone
16. Microwave
17. Dog barking
18. Walking (hard surface)
19. Road traffic
20. Zipper
21. Cellphone vibrating
22. Water dripping
23. Scratching
24. Car windows
25. Telephone ringing
26. Chopping food
27. Telephone dialing
28. Girl speaking
29. Car horn
30. Writing
31. Computer startup sound
32. Background speech
33. Songbird
34. Pouring water
35. Pop song
36. Water boiling
37. Guitar
38. Coughing
39. Crumpling paper
40. Siren

…

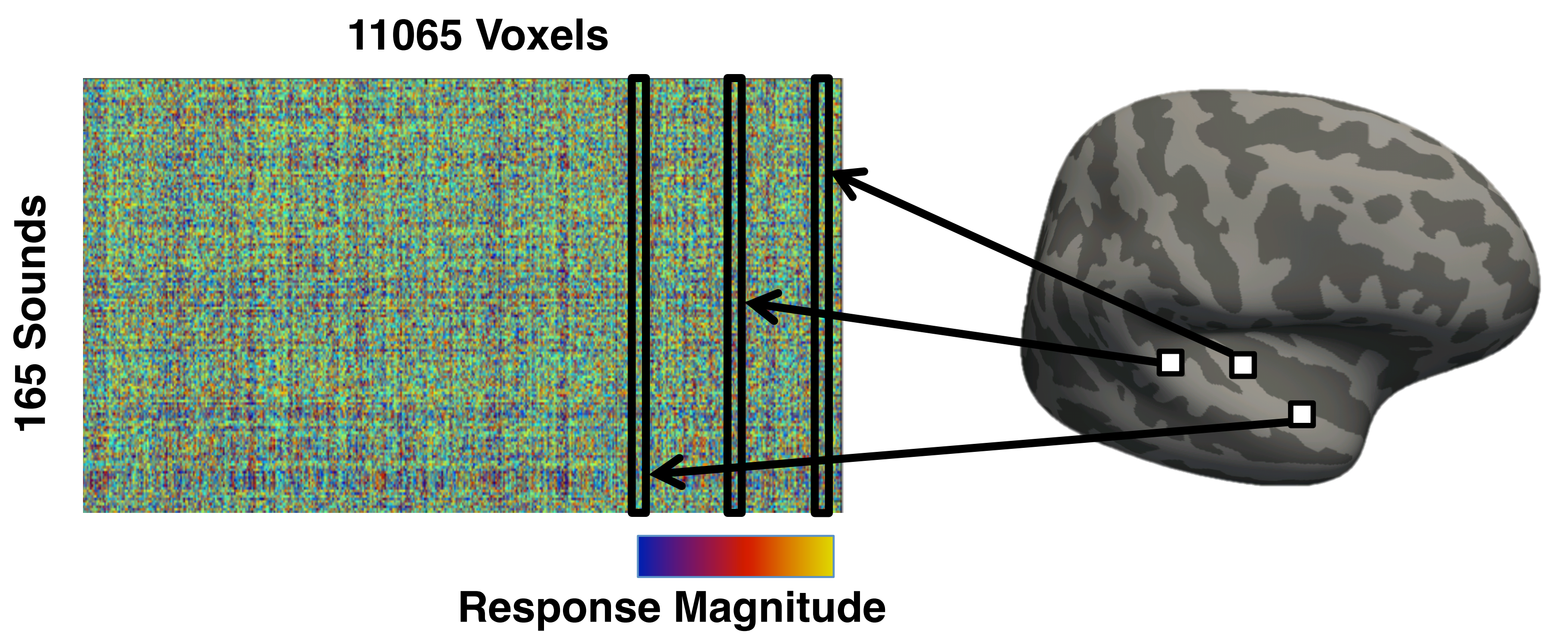
Working Example from My Research (for Convenience)

- Measured fMRI responses to 165 natural sounds:
- For each voxel, measure average response to each sound



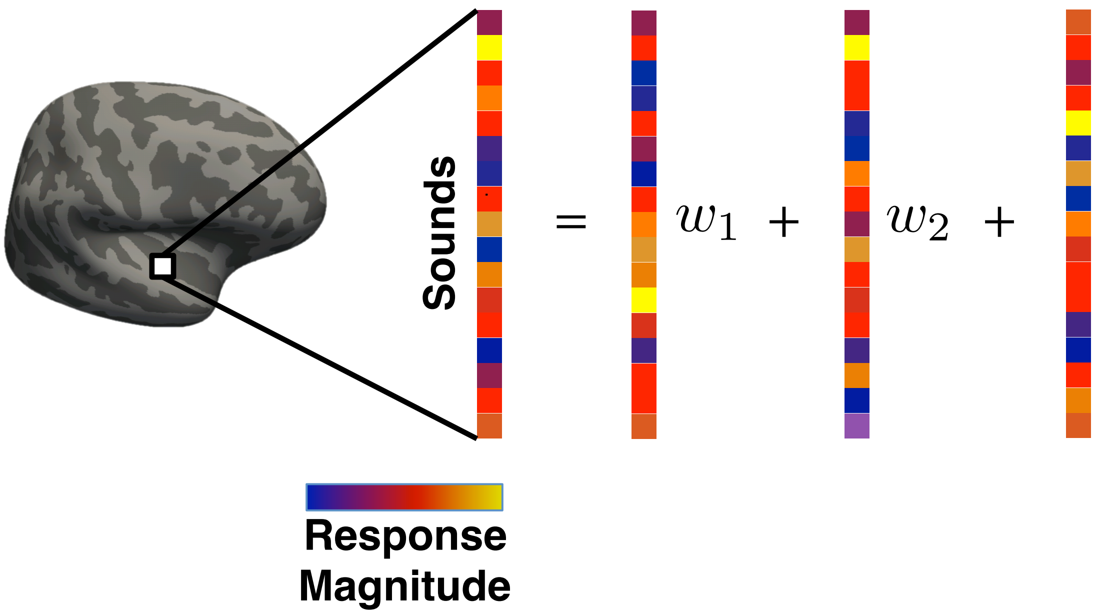
Working Example from My Research (for Convenience)

- Measured fMRI responses to 165 natural sounds:
- For each voxel, measure average response to each sound
- Compile all voxel responses into a matrix



Hypothesis: Perhaps a small number of neural populations – each with a canonical response to the sound set – explain the response of thousands of voxels?

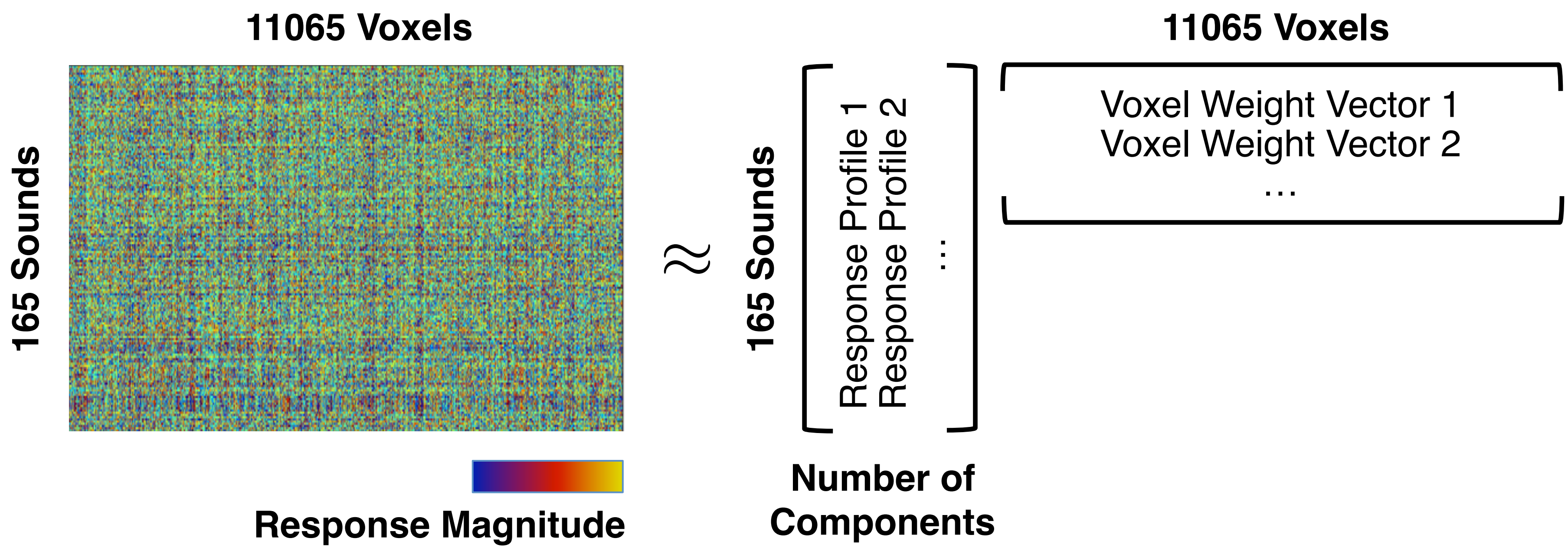
Linear Model of Voxel Responses



$$\mathbf{v} = \mathbf{r_1} w_1 + \mathbf{r_2} w_2 + \mathbf{r_3} w_3 + \ldots \qquad \mathbf{v} = \sum_{i=1}^{N} \mathbf{r_i} w_i$$

Voxel responses modeled as weighted sum of response profiles

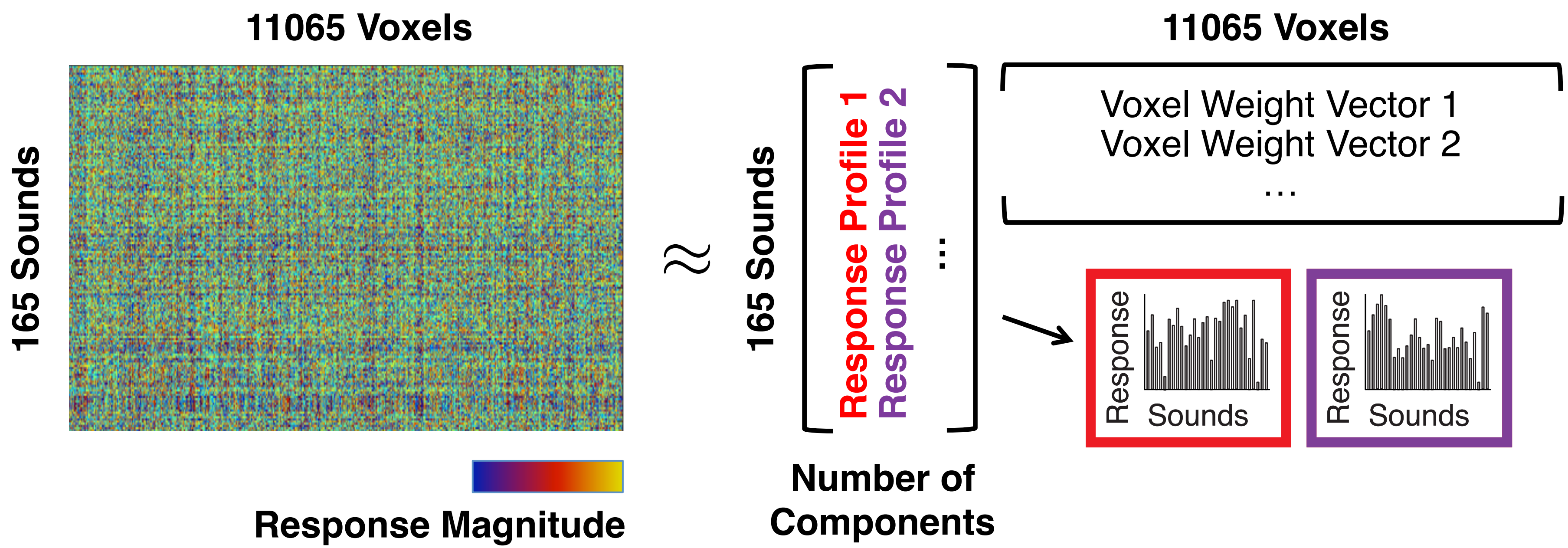
Matrix Factorization



Factor response matrix into set of components, each with:

1. Response profile to all 165 sounds
2. Voxel weights specifying contribution of each component to each voxel

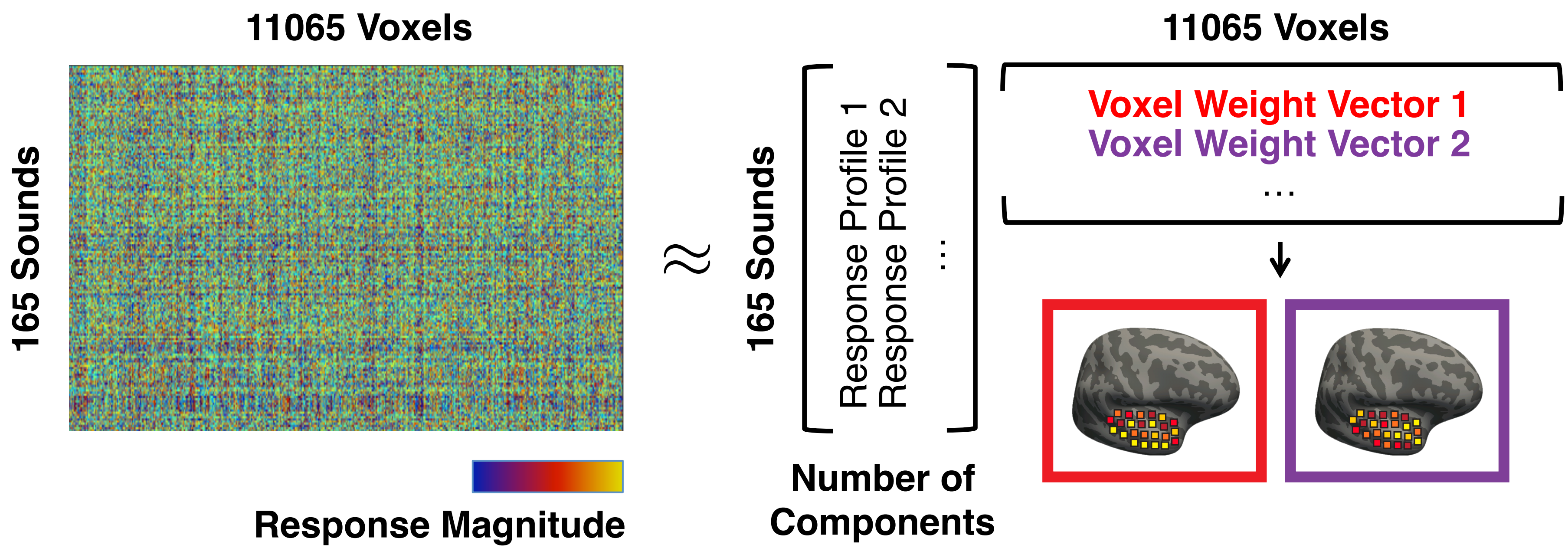
Matrix Factorization



Factor response matrix into set of components, each with:

1. Response profile to all 165 sounds
2. Voxel weights specifying contribution of each component to each voxel

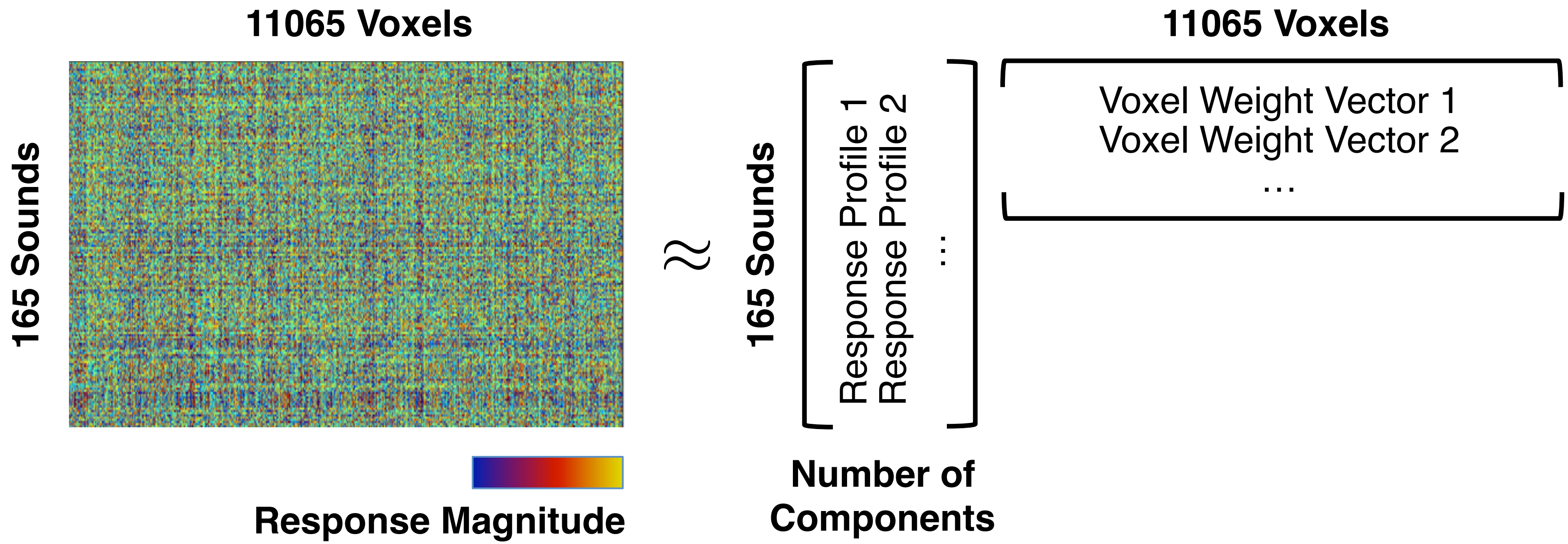
Matrix Factorization



Factor response matrix into set of components, each with:

1. Response profile to all 165 sounds
2. Voxel weights specifying contribution of each component to each voxel

Matrix Factorization

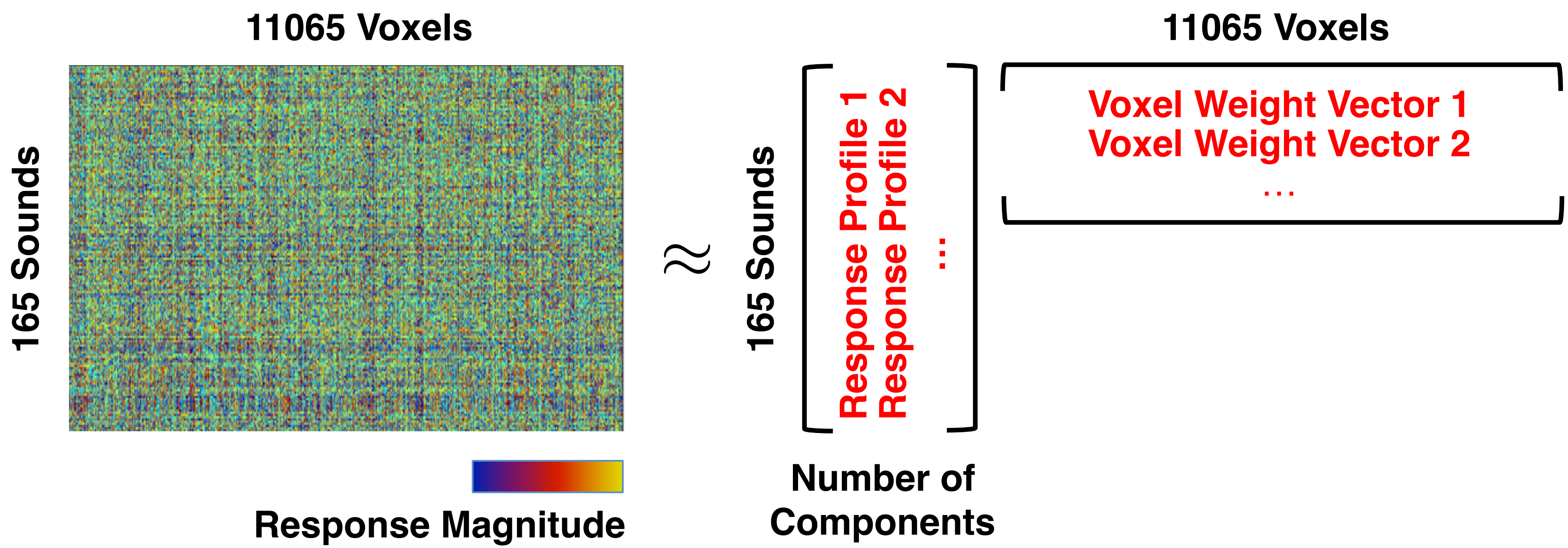


Matrix approximation ill-posed (many equally good solutions)

⇒Must be constrained with additional assumptions

⇒Different techniques make different assumptions

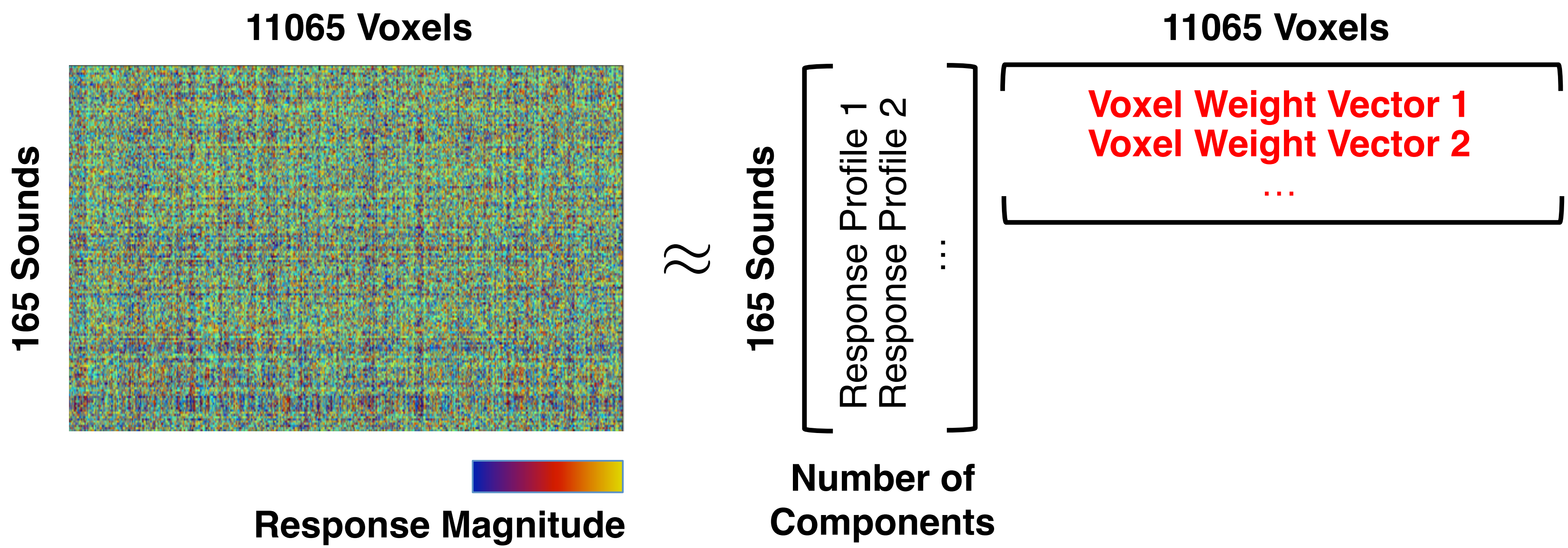
Principal Components Analysis (PCA)



For PCA to infer underlying components, they must:

1. Have uncorrelated response profiles and voxel weights
2. Explain different amounts of response variance

Independent Components Analysis (ICA)



For ICA to infer underlying components, they must:

1. Have non-Gaussian and statistically independent voxel weights

An Aside on Statistical Independence

Saying that voxel weights are independent means:

$\Rightarrow$ The weight of one component tells you nothing about the weight of another

$$p(w_1, w_2) = p(w_1)p(w_2)$$

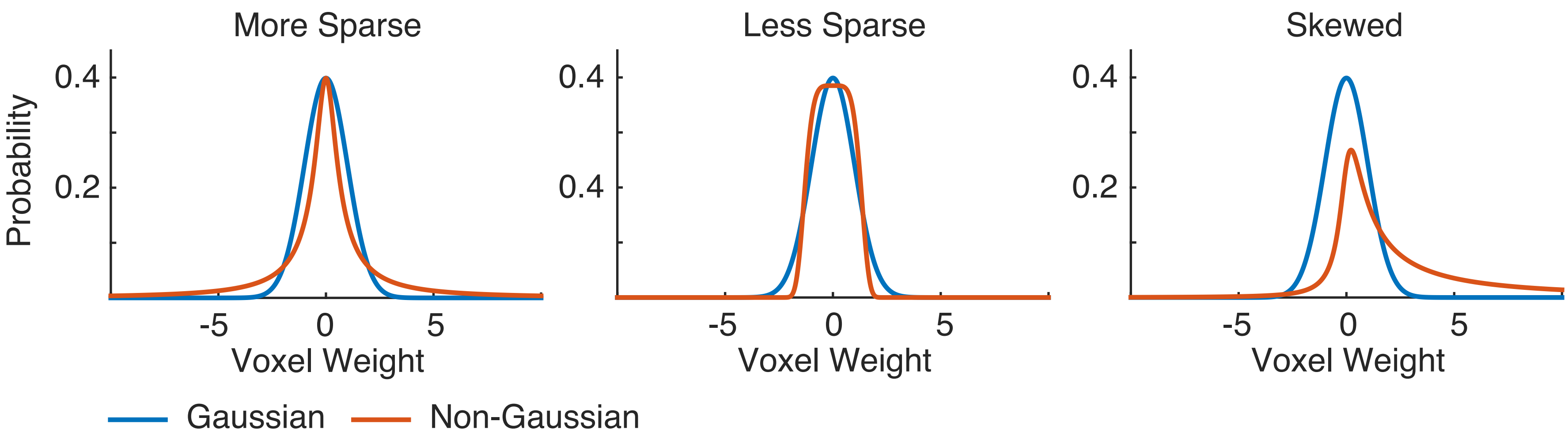
Statistical independence a stronger assumption uncorrelatedness

$\Rightarrow$ All independent variables are uncorrelated

$\Rightarrow$ Not all uncorrelated variables are independent:

An Aside on non-Gaussianity

Many ways for a distribution to be non-Gaussian:



Non-Gaussinity and Statistical Indepedence

Central limit theorem (non-technical):

Sums of independent non-Gaussian distributions become more Gaussian

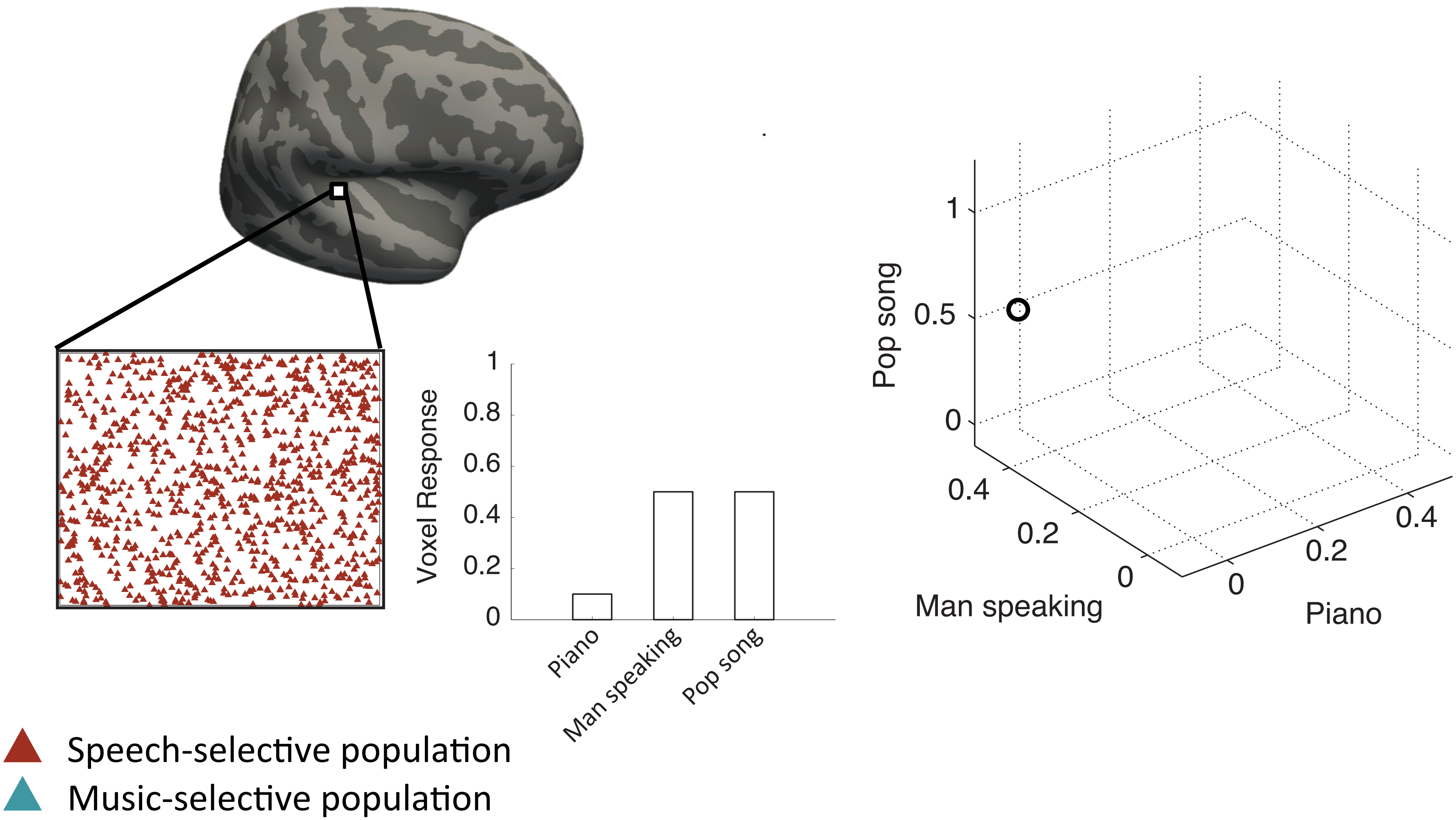
Consequence:

Maximally non-Gaussian projections of the data are more likely to be sources

What if the sources have a Gaussian distribution?

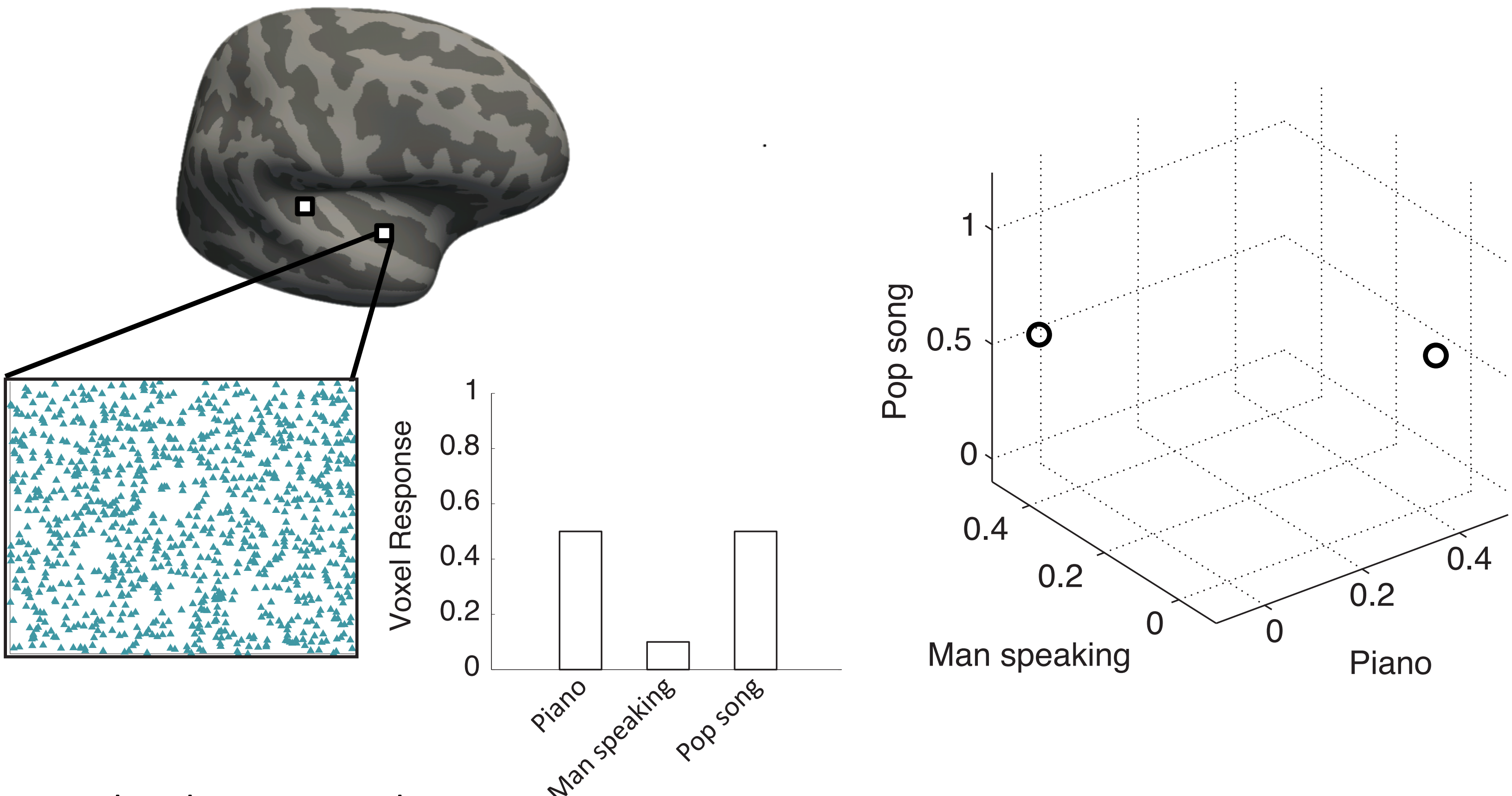
Out of luck: Sums of Gaussian distributions remain Gaussian

Toy Example



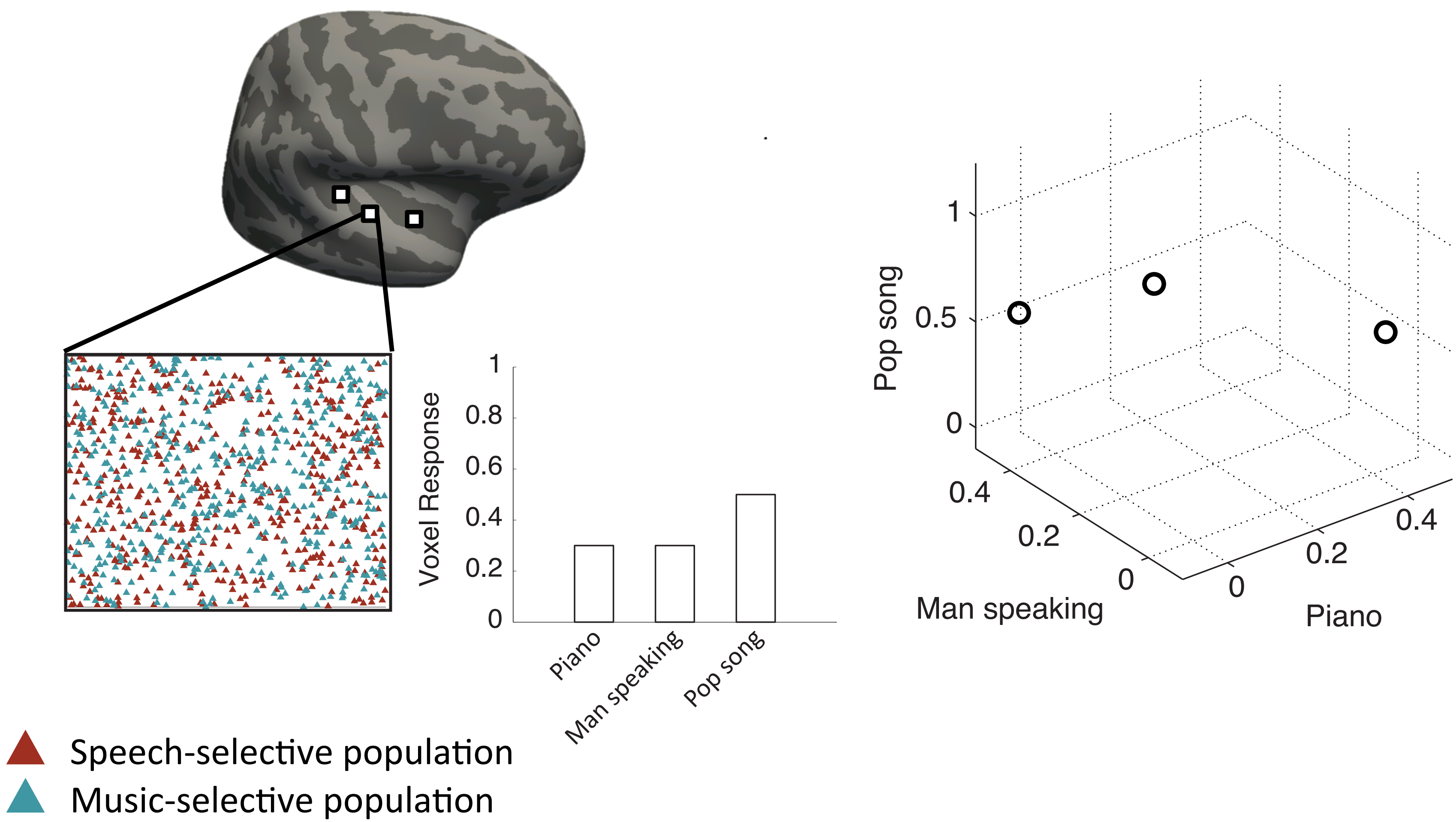
Voxel Response
1
0.8
0.6
0.4
0.2
0
Piano
Man speaking
Pop song
Pop song
1
0.5
0
Man speaking
0.4
0.2
0
Piano
0
0.2
0.4
Speech-selective population
Music-selective population

Toy Example



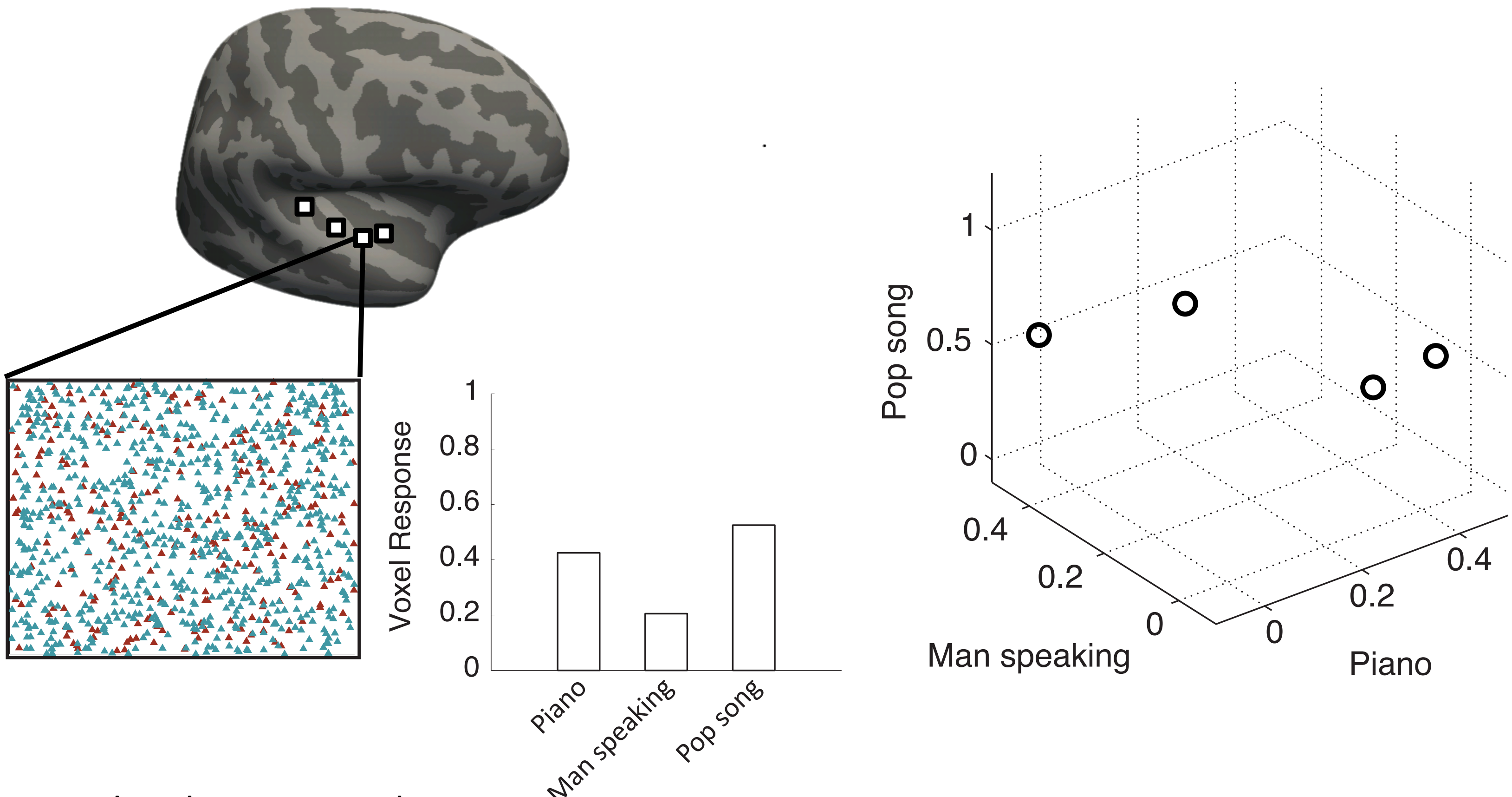
Speech-selective population
Music-selective population

Toy Example



Voxel Response
1
0.8
0.6
0.4
0.2
0
Piano
Man speaking
Pop song
Pop song
1
0.5
0
Man speaking
0.4
0.2
0
Piano
0
0.2
0.4
Speech-selective population
Music-selective population

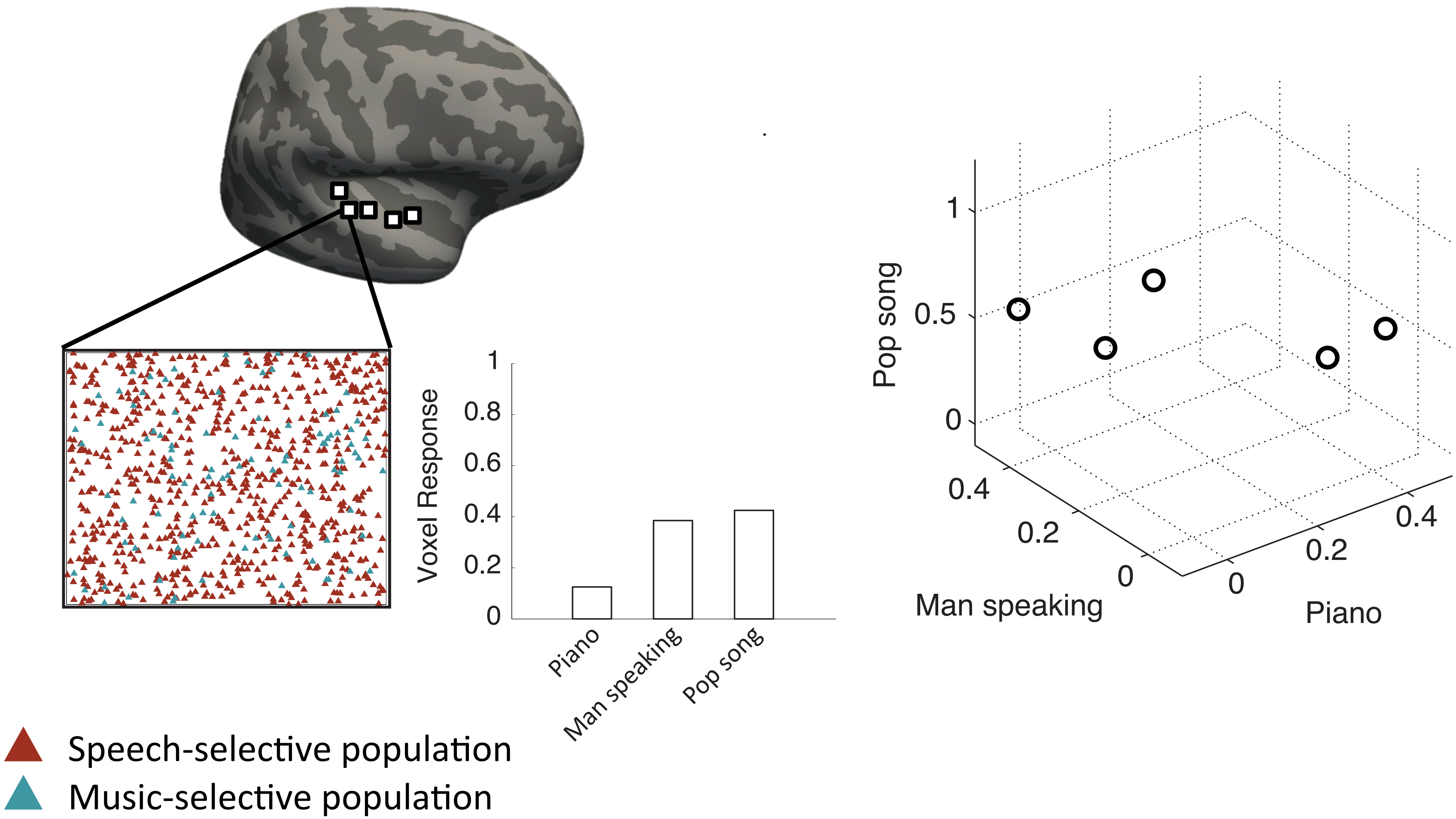
Toy Example



Speech-selective population

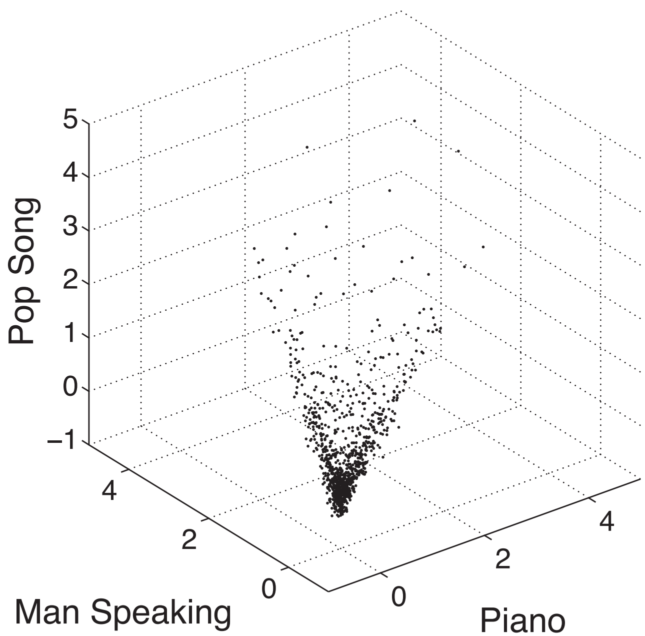
Music-selective population

Toy Example

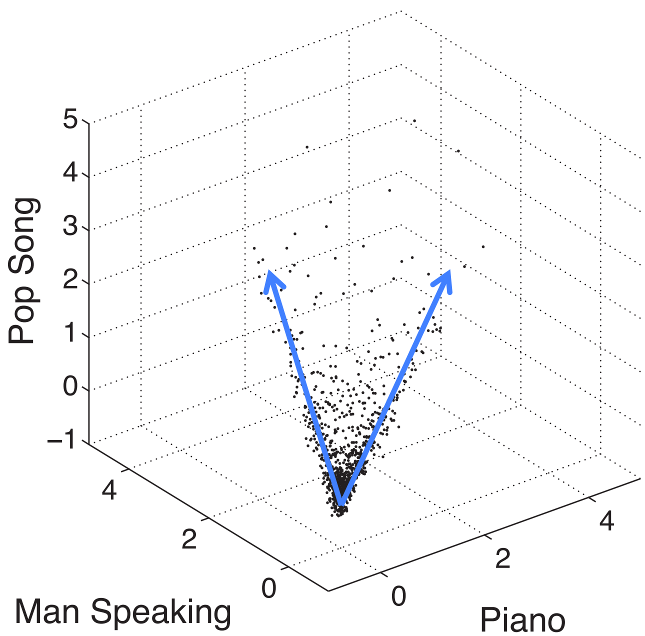


1
0.8
0.6
0.4
0.2
0
Voxel Response
Piano
Man speaking
Pop song
1
0.5
0
Pop song
0.4
0.2
0
Man speaking
0
0.2
0.4
Piano
Speech-selective population
Music-selective population

Toy Example

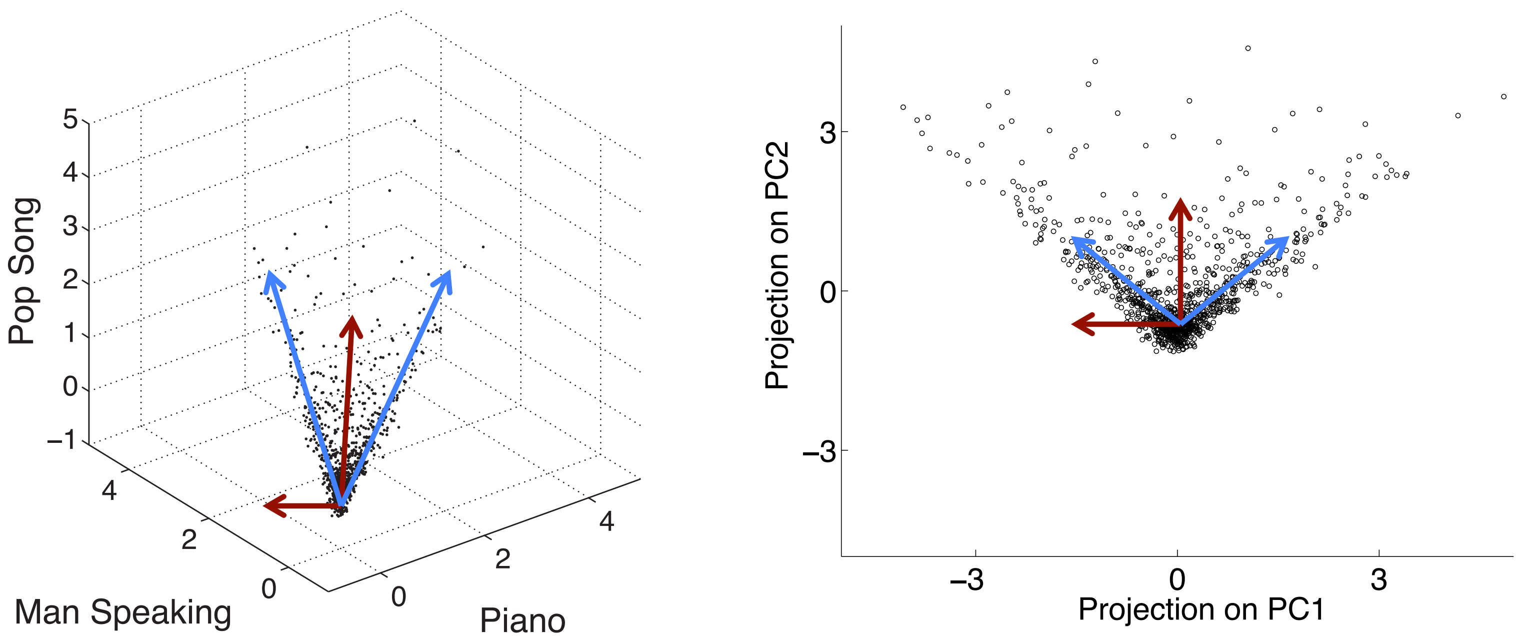


Toy Example



"True" dimensions

Limitation of PCA

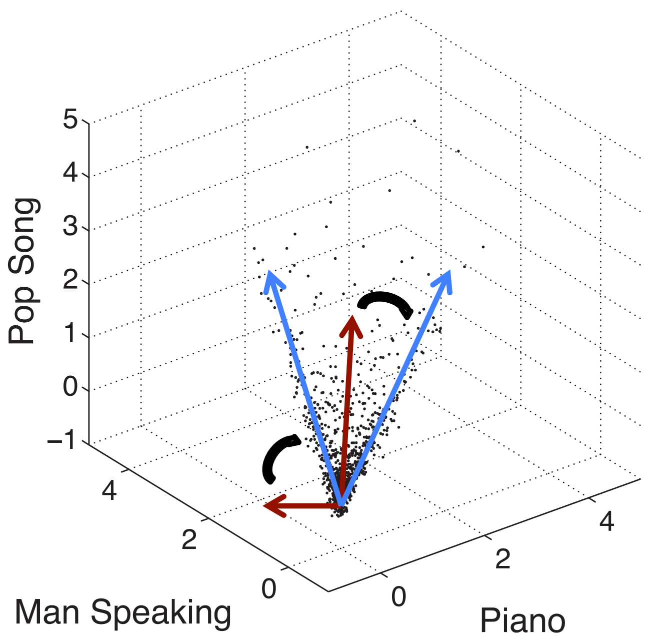


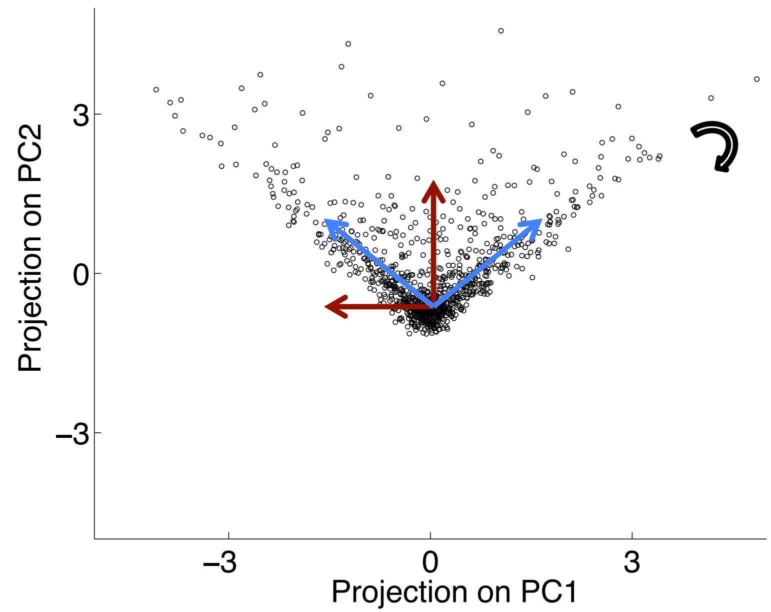
“True” dimensions

PCA Dimensions

- PCA infers the right subspace
- But the specific directions are misaligned

Limitation of PCA



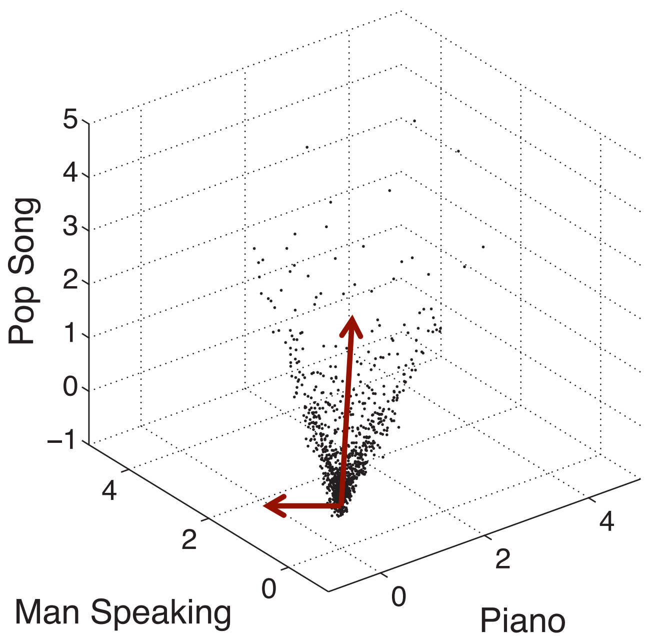


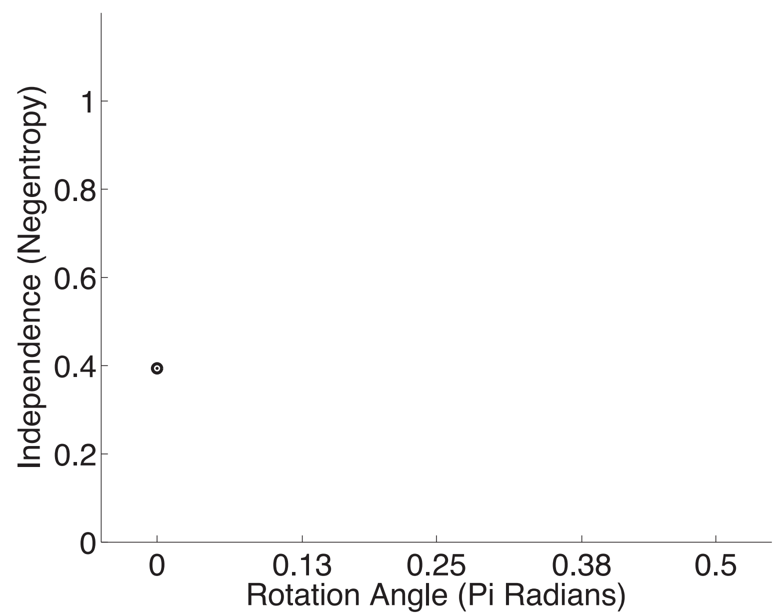
“True” dimensions

PCA Dimensions

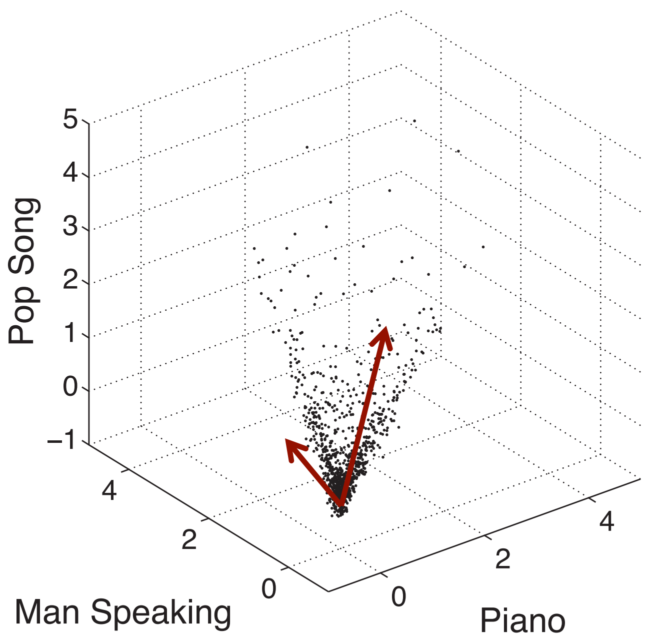
Can recover the “true” dimensions by rotating in the whitened PCA space

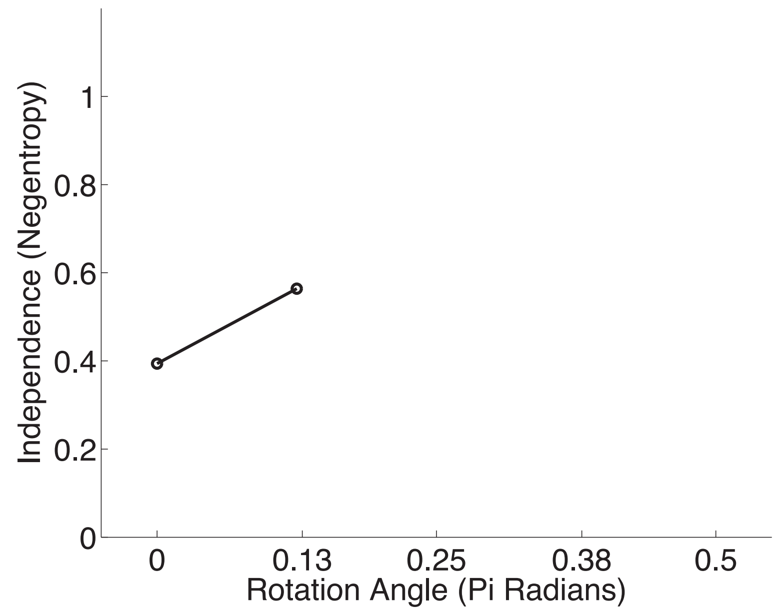
Rotating PCA Dimensions



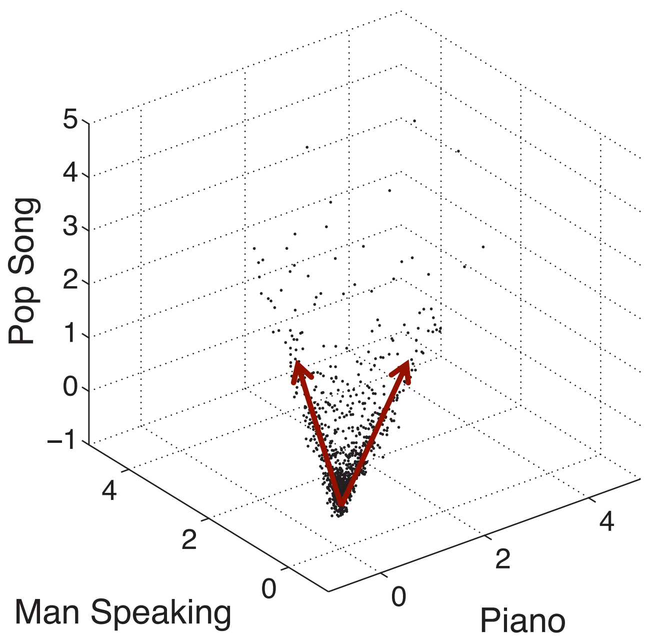


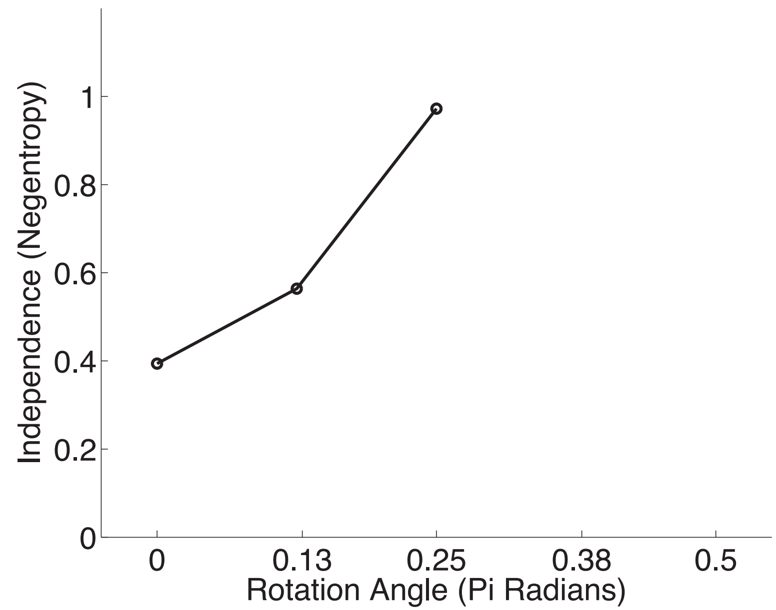
Rotating PCA Dimensions



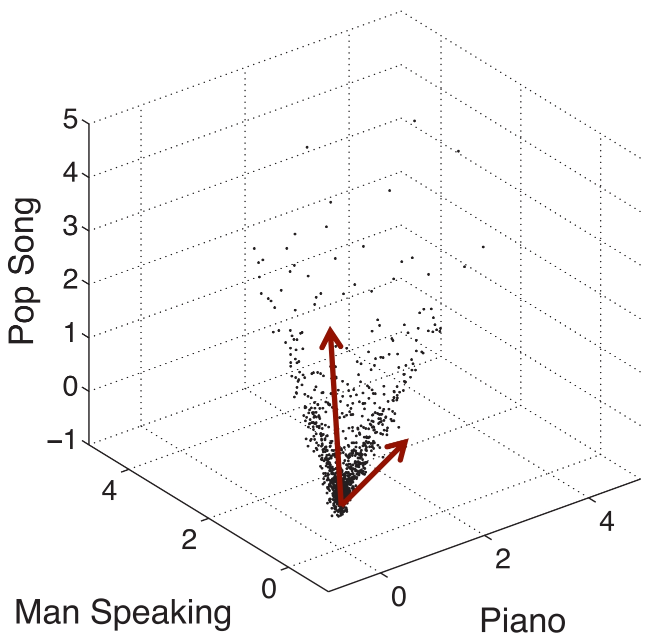


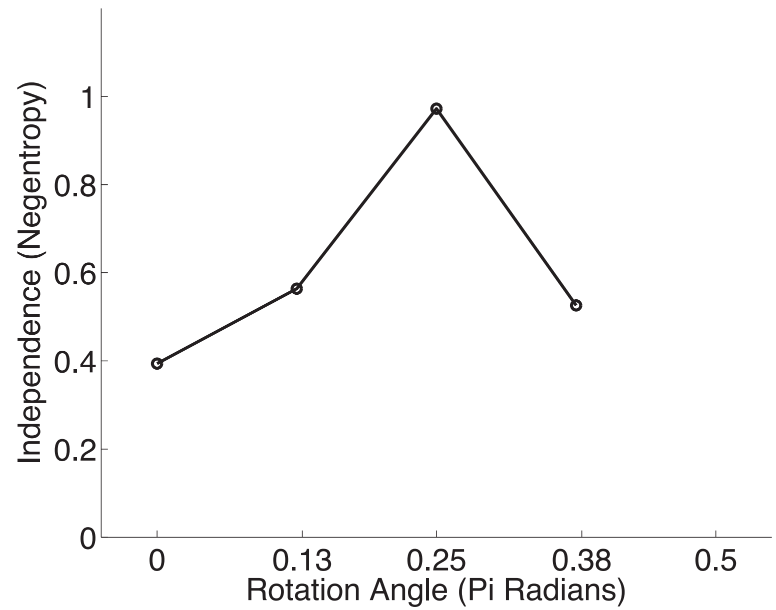
Rotating PCA Dimensions



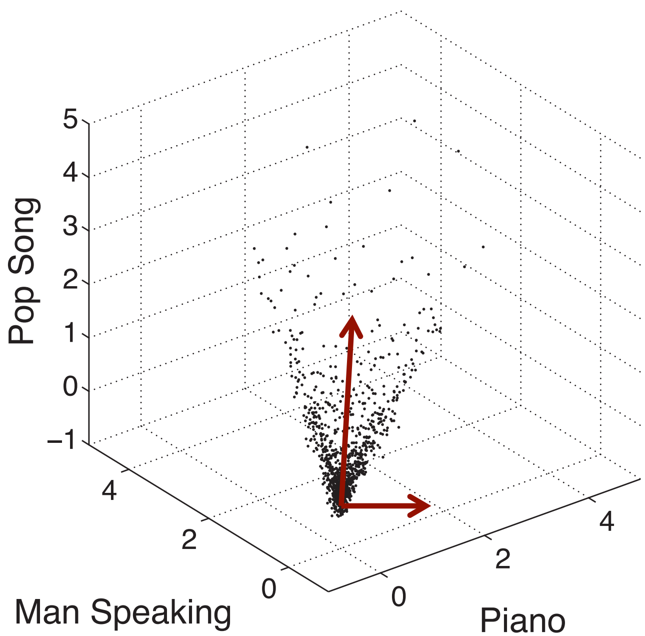


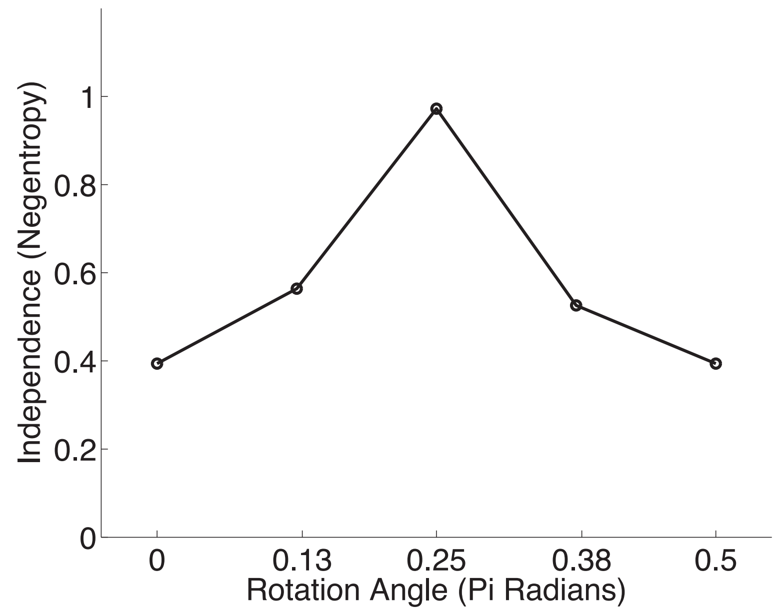
Rotating PCA Dimensions



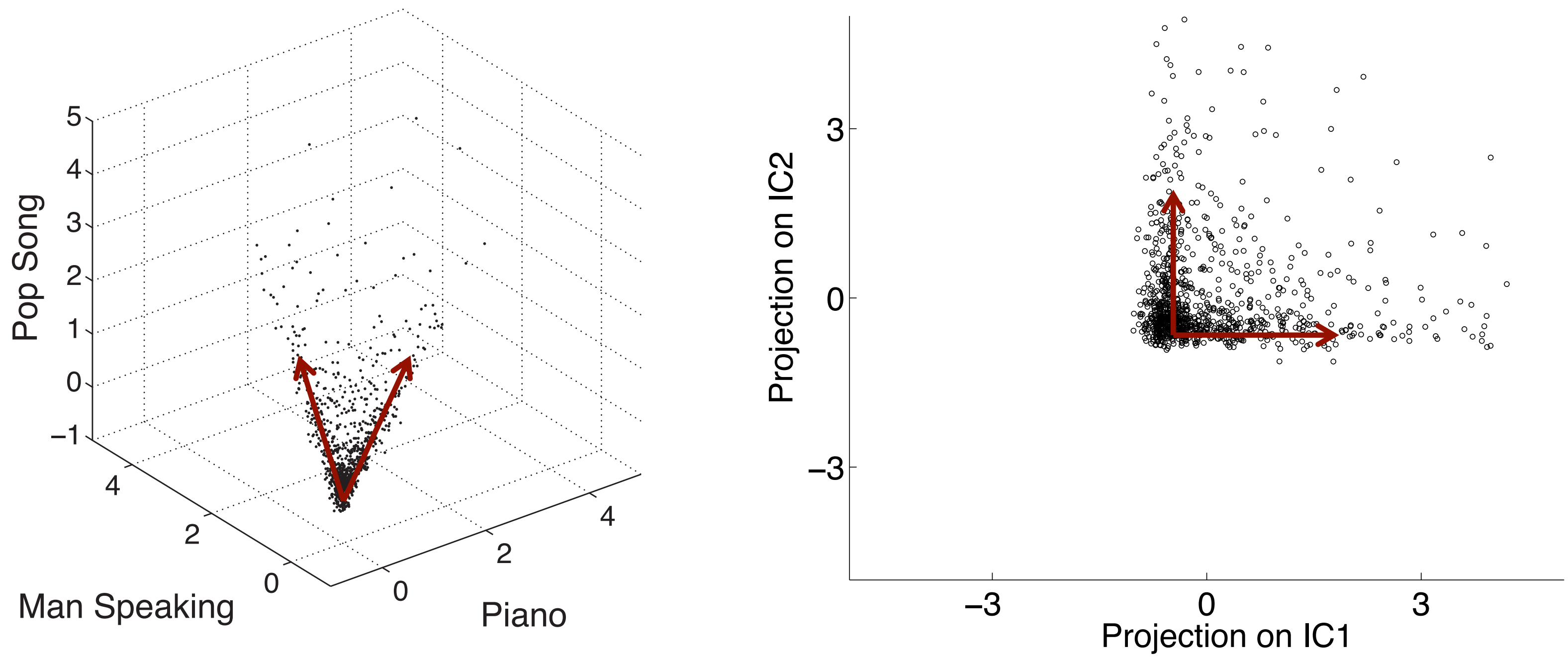


Rotating PCA Dimensions



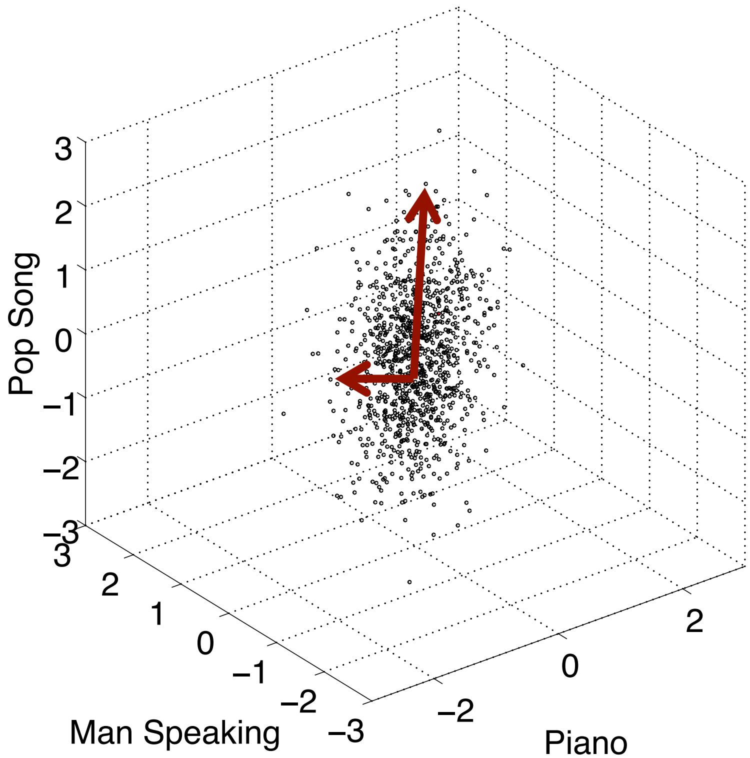


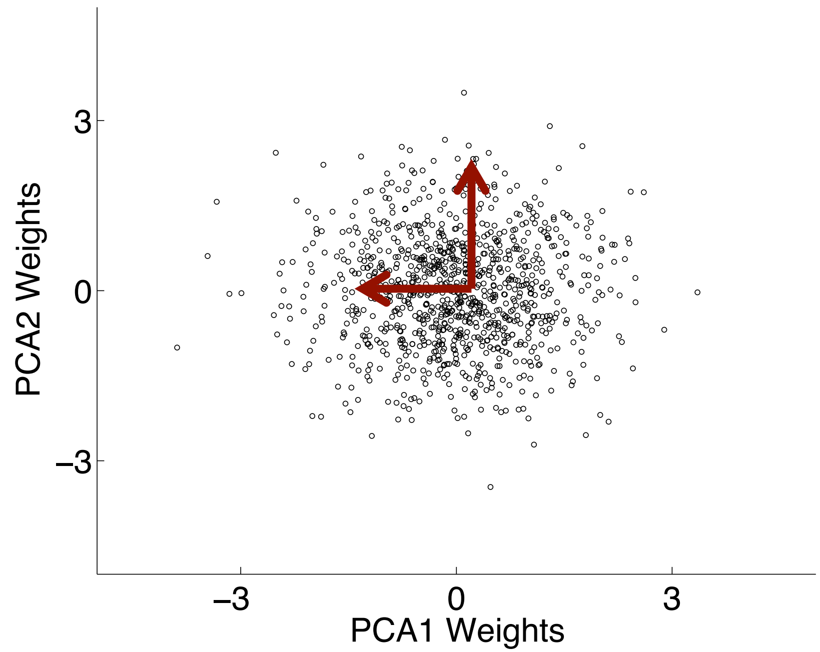
ICA Dimensions



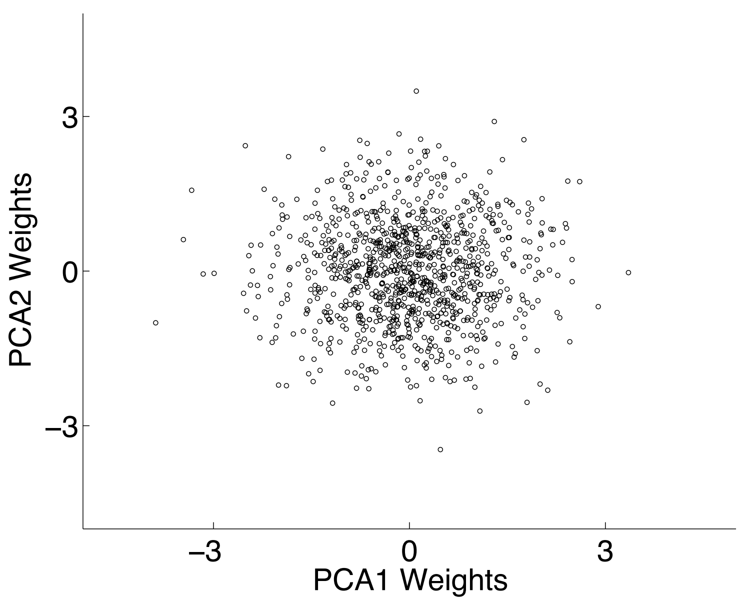
ICA rotates PCA components to maximize statistical independence / non-Gaussianity

What if the data is Gaussian?

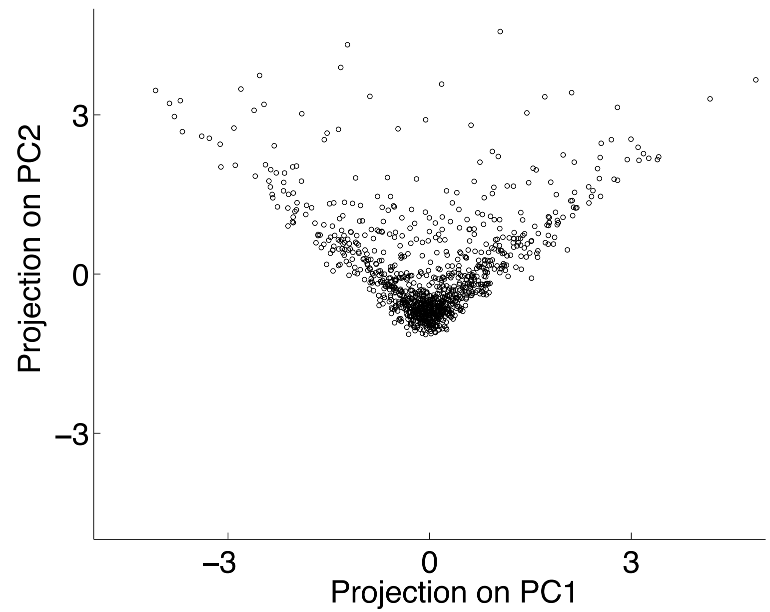




Gaussian Data



Non-Gaussian Data



For Gaussian distributions

⟹ Projections on PCA components are circularly symmetric

⟹ No “special directions”

For non-Gaussian distributions:

⟹ Can recover latent components by searching for “special directions” that have maximally non-Gaussian projections

A Simple 2-Step Recipe

1. PCA: whiten data

   $\Longrightarrow$ Possibly discard low-variance components

   $\Longrightarrow$ How many components to discard?

2. ICA: rotate whitened PCA components to maximize non-Gaussianity

   $\Longrightarrow$ How to measure non-Gaussianity?

   $\Longrightarrow$ How to maximize your non-Gaussianity measure?

Measuring Non-Gaussianity: Negentropy (gold standard)

- Definition: difference in entropy from a Gaussian

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y})$$

- Gaussian distribution is maximally entropic (for fixed variance)

  $\Longrightarrow$All non-Gaussian distributions have positive negentropy

- Maximizing negentropy closely related to minimizing mutual information

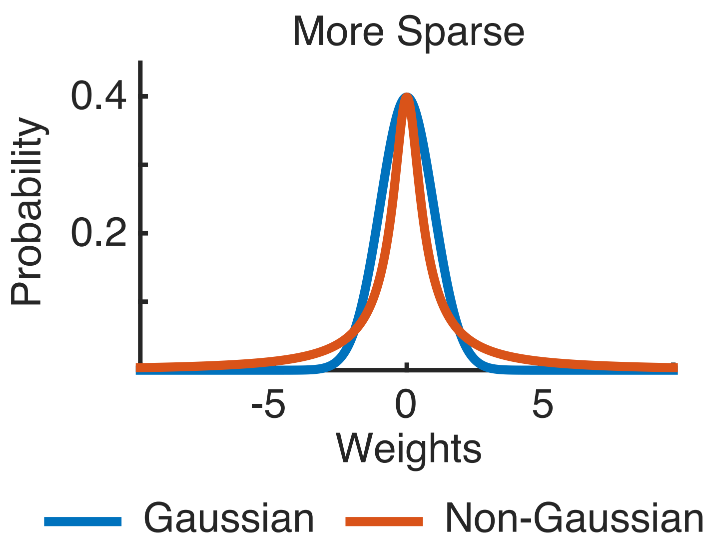
- Cons: in practice, can be hard to measure and optimize

Measuring Non-Gaussianity: Kurtosis (approximation)

- Definition: 4th moment of the distribution

$$\mathbf{E}[y^4]$$

- Useful for sparse, 'heavy tailed' distributions (which are common)



Measuring Non-Gaussianity: Kurtosis (approximation)

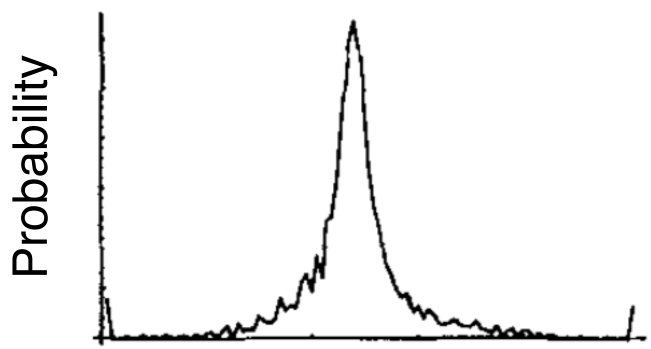
- Definition: 4th moment of the distribution

$$\mathbf{E}[y^4]$$

- Useful for sparse, ‘heavy tailed’ distributions (which are common)

  $\Longrightarrow$ Many audio sources have a sparse distribution of amplitudes

Histogram of Speech
Waveform Amplitudes



Waveform Amplitudes

Bell & Sejnowski, 1995

Measuring Non-Gaussianity: Kurtosis (approximation)

- Definition: 4th moment of the distribution

$$\mathbf{E}[y^4]$$

- Useful for sparse, 'heavy tailed' distributions (which are common)
  - $\Longrightarrow$ Many audio sources have a sparse distribution of amplitudes
  - $\Longrightarrow$ Natural images tend to be sparse (e.g. Olshausen & Field, 1997)

- Very easy to measure and optimize

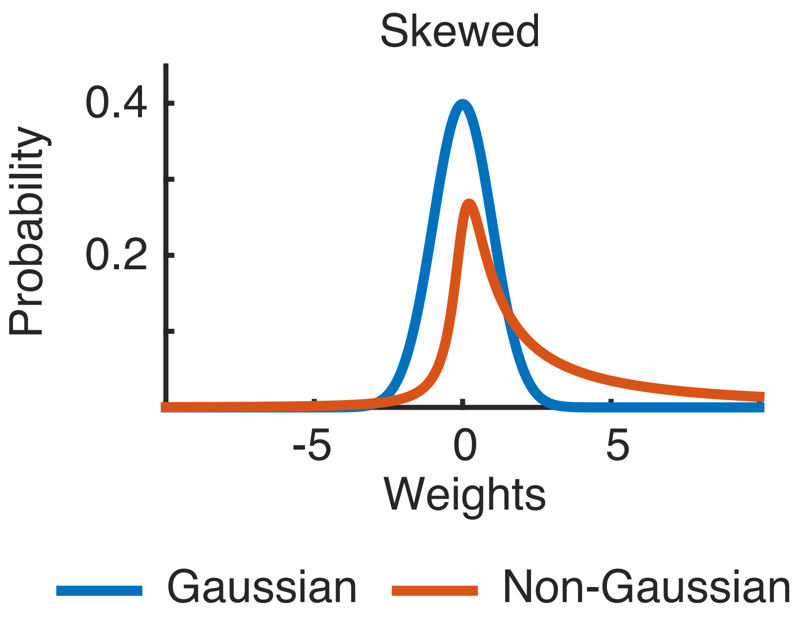
- Cons: only useful if the source distributions are sparse, sensitive to outliers

Measuring Non-Gaussianity: Skew (approximation)

- Definition: 3rd moment of the distribution

$$\mathbf{E}[y^3]$$

- Useful for distributions with a single heavy tail



Measuring Non-Gaussianity: Skew (approximation)

- Definition: 3rd moment of the distribution

$$\mathbf{E}[y^3]$$

- Useful for distributions with a single heavy tail

- Again easy to measure and optimize

- Only useful if the source distributions are skewed

Measuring Non-Gaussianity

Bottom line:

- Negentropy a general-purpose measure of non-Gaussianity, but often hard to use in practice
- Parametric measures can be more effective if tailored to the non-Gaussianity of the source distribution

Non-Gaussianity Maximization

- Brute-force search
  - $\Rightarrow$ e.g. iteratively rotate pairs of components to maximize non-Gaussianity
  - $\Rightarrow$ Easy-to-implement, effective in low-dimensional spaces

- Gradient-based (many variants)
  - $\Rightarrow$ More complicated to implement, effective in high dimensions

- All optimization algorithms attempt to find local, not global, optima
  - $\Rightarrow$ Useful to test stability of local optima
  - $\Rightarrow$ e.g. run algorithm many times from random starting points

A Simple 2-Step Recipe Applied to fMRI Data!

1. PCA: whiten data

   $\Longrightarrow$ Possibly discard low-variance components

   $\Longrightarrow$ How many components to discard?

2. ICA: rotate whitened PCA components to maximize non-Gaussianity

   $\Longrightarrow$ How to measure non-Gaussianity?

   $\Longrightarrow$ How to maximize your non-Gaussianity measure?

A Simple 2-Step Recipe Applied to fMRI Data!

1. PCA: whiten data

   $\Longrightarrow$ Possibly discard low-variance components

   $\Longrightarrow$ **How many components to discard?**

2. ICA: rotate whitened PCA components to maximize non-Gaussianity

   $\Longrightarrow$ How to measure non-Gaussianity?

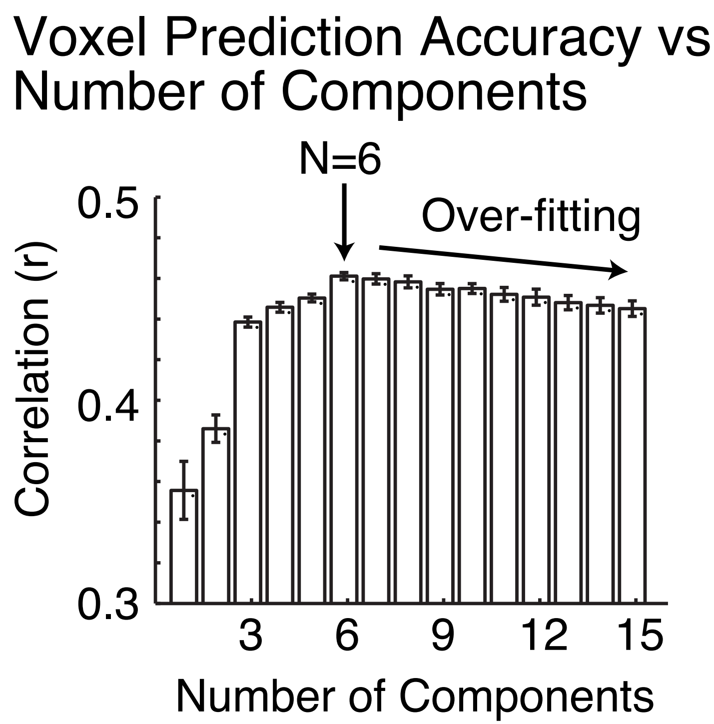
   $\Longrightarrow$ How to maximize your non-Gaussianity measure?

Choosing the Number of Components

Using cross-validation to select components

⇒ Project voxel responses onto principal components (using subset of data)

⇒ Predict responses from left-out data using different numbers of components



A Simple 2-Step Recipe Applied to fMRI Data!

1. PCA: whiten data

   $\Longrightarrow$ Possibly discard low-variance components

   $\Longrightarrow$ **Keep top 6 Components**

2. ICA: rotate whitened PCA components to maximize non-Gaussianity

   $\Longrightarrow$ How to measure non-Gaussianity?

   $\Longrightarrow$ How to maximize your non-Gaussianity measure?

A Simple 2-Step Recipe Applied to fMRI Data!

1. PCA: whiten data

   ⇒ Possibly discard low-variance components

   ⇒ **Keep top 6 Components**

2. ICA: rotate whitened PCA components to maximize non-Gaussianity

   ⇒ **Use negentropy to measure non-Gaussianity**

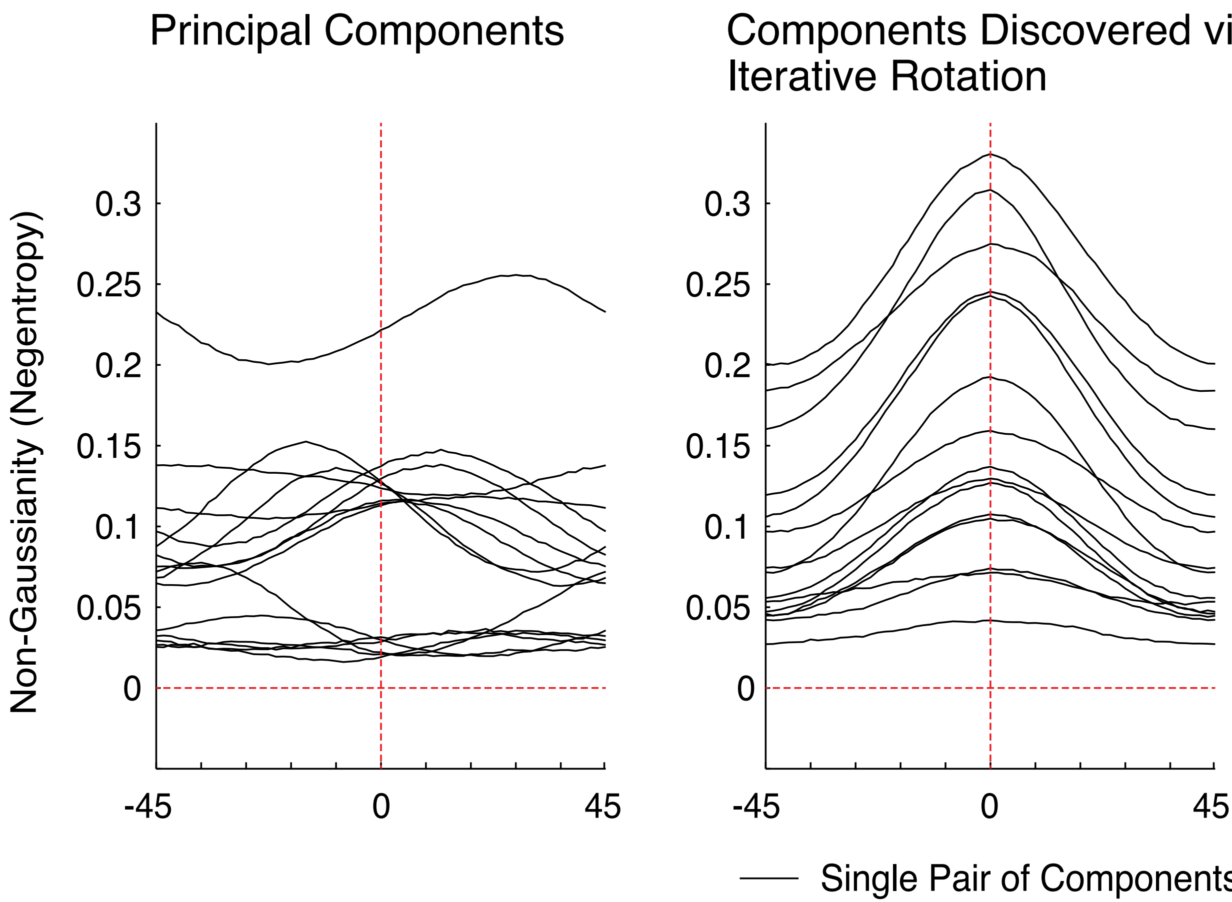
   ⇒ **Maximize negentropy via brute-force rotation**

   ⇒ Feasible because:

      1. Many voxels / data points (>10,000 voxels)

      2. Low-dimensional data (just 6 dimensions)

Rotating the Top 6 Principal Components



- Negentropy of principal components can be increased by rotation
- Rotation algorithm discovers clear optimum

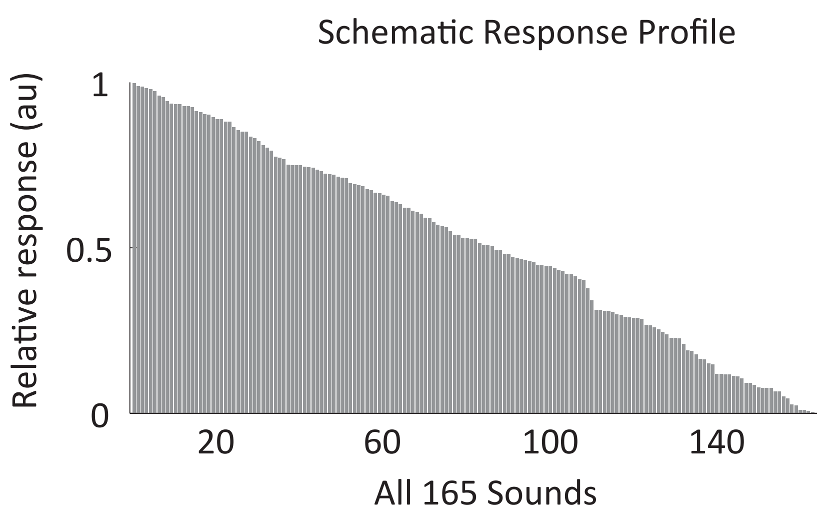
Probing the Inferred Components

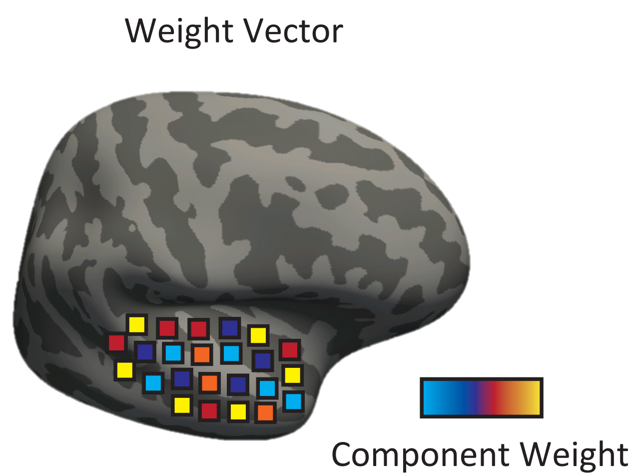
We now have 6 dimensions, each with:

1. A response profile (165-dimensional vector)

2. A weight vector, specifying its contribution to each voxel

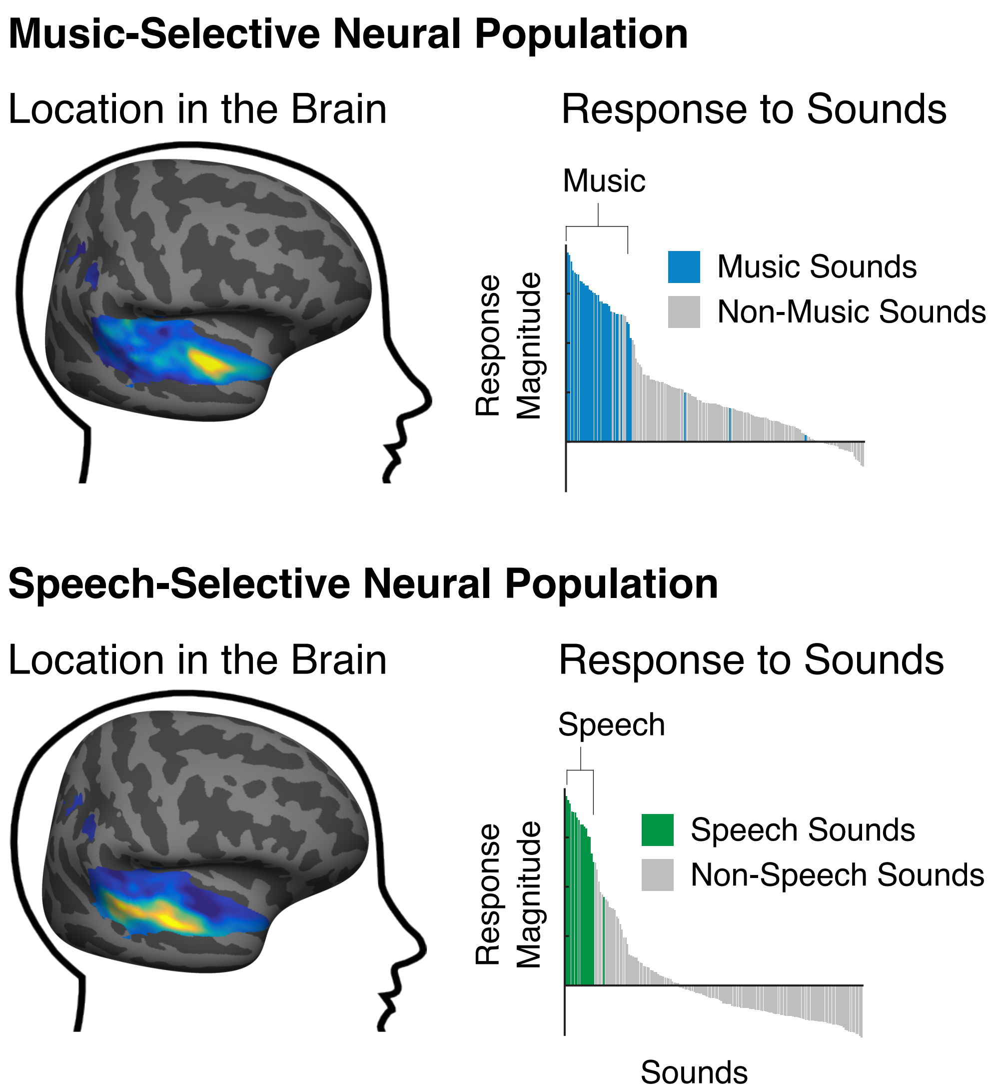
⇒ **Both response profile and anatomy unconstrained**





All 6 inferred components have interpretable properties

⇒ 2 components highly selective for speech and music, respectively



All 6 inferred components have interpretable properties

⇒ 2 components highly selective for speech and music, respectively

Music-selectivity highly diluted in raw voxels

⇒ fMRI signals likely blur activity from different neural populations

PCA components differ qualitatively from ICA components, with less clear functional/anatomical properties

⇒ ICA components have response profiles with substantial correlations

Conclusions

1. Core assumptions of ICA

   ⇒Measured signals reflects linear mixture of underlying sources

   ⇒Sources are non-Gaussian and statistically independent

2. When core assumptions hold, the method is very effective, and requires few additional assumptions about the nature of the underlying sources

Exercise: 2-Step Recipe Applied to the Cocktail Party

Dataset

$\Longrightarrow$ 5 audio soundtracks mixed from 3 sources

2-step recipe:

1. Project data onto the top 3 principal components
2. Iteratively rotate pairs of components to maximize negentropy

$\Longrightarrow$ Listen to the unmixed signals!