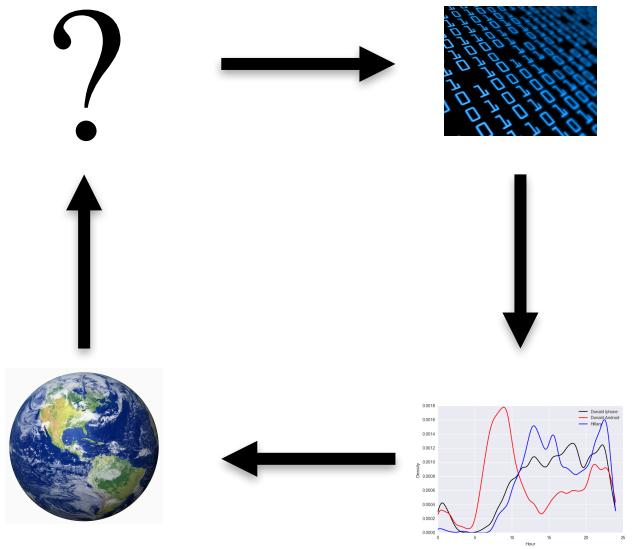
Data Science 100
Lec 25:
EM and Hierarchical Clustering

DS100
Spring
2017

Slides by:
Bin Yu
binyu@stat.berkeley.edu
Thanks to Andrew Do and Rebecca Barter for assistance on data analysis



Recap

- Clustering is an old activity and is used for information organization
- New name: **PQRS**

  (actually, it has been used for Physician Quality Reporting System!)
- K-means algorithm: initial values, choice of K
- PCA – dimensionality reduction (details to come)

Silhouette (Peter J. Rousseeuw, 1986): graphical method for K selection

Given k and k clusters, given any data point i , let a_i
be the average distance or dissimilarity of i with all other points in the same cluster.
For Euclidean k-means, use Euclidean distance for dissimilarity

a_i measures how well i fits into its cluster

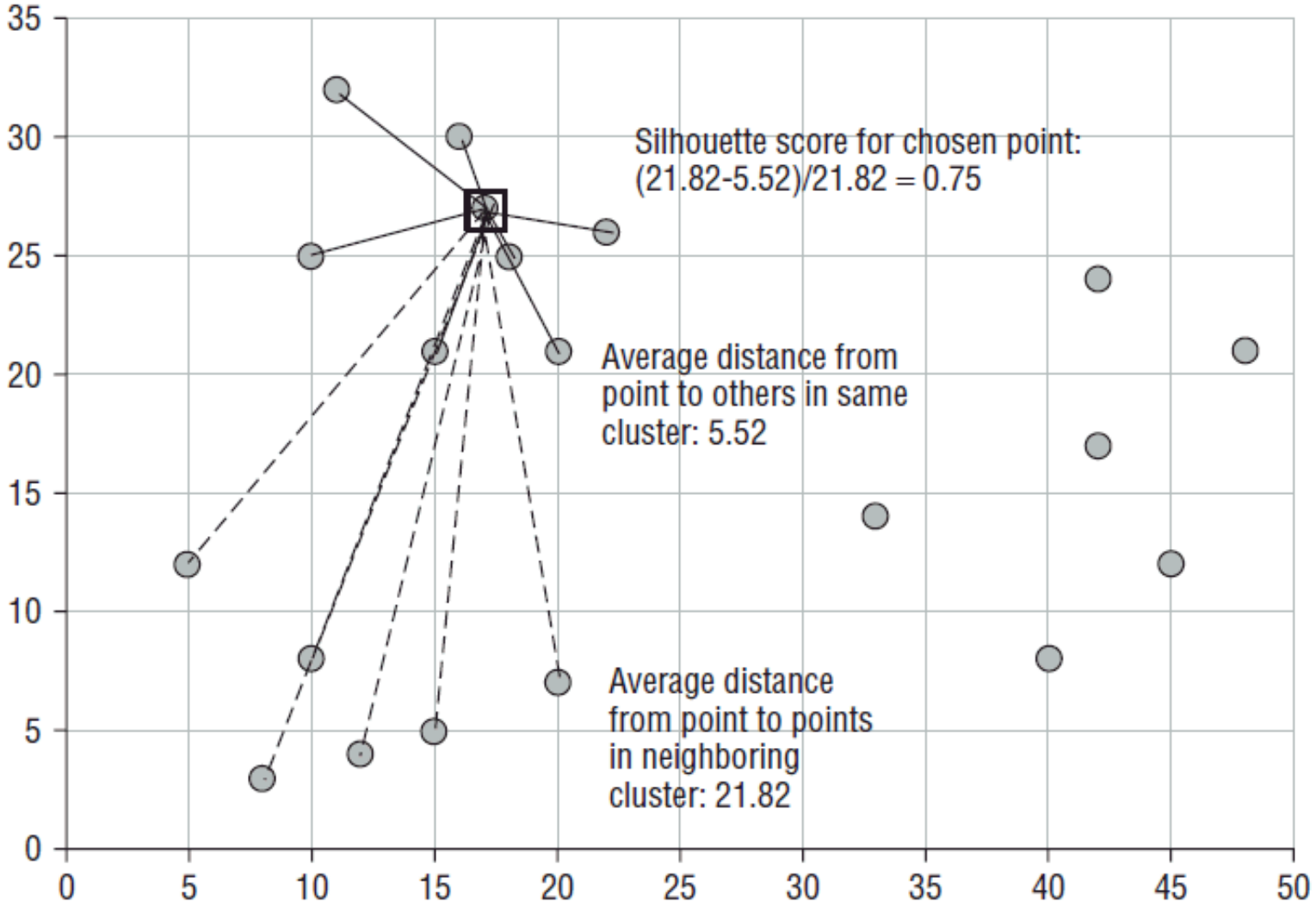
b_i is the smallest average distance of i to other clusters (each cluster gets an average distance)

$$s_i = \frac{b_i - a_i}{max(b_i, a_i)}$$ which is between -1 and 1 and called the Silhouette score

s_i is close to 1 if point i is in a tight cluster and far away from other clusters; close to -1, if it is in a loose cluster and close to other clusters.

Maximize over k $\frac{1}{n}\sum_{i=1}^{n} s_i$ (called the average Silhouette width)

An example: consider the ith point in the box



a_i = 0.5.52

b_i = 21.82 (because the other cluster is further away by visal inspection)

s_i = 0.75 is the Silhouette score

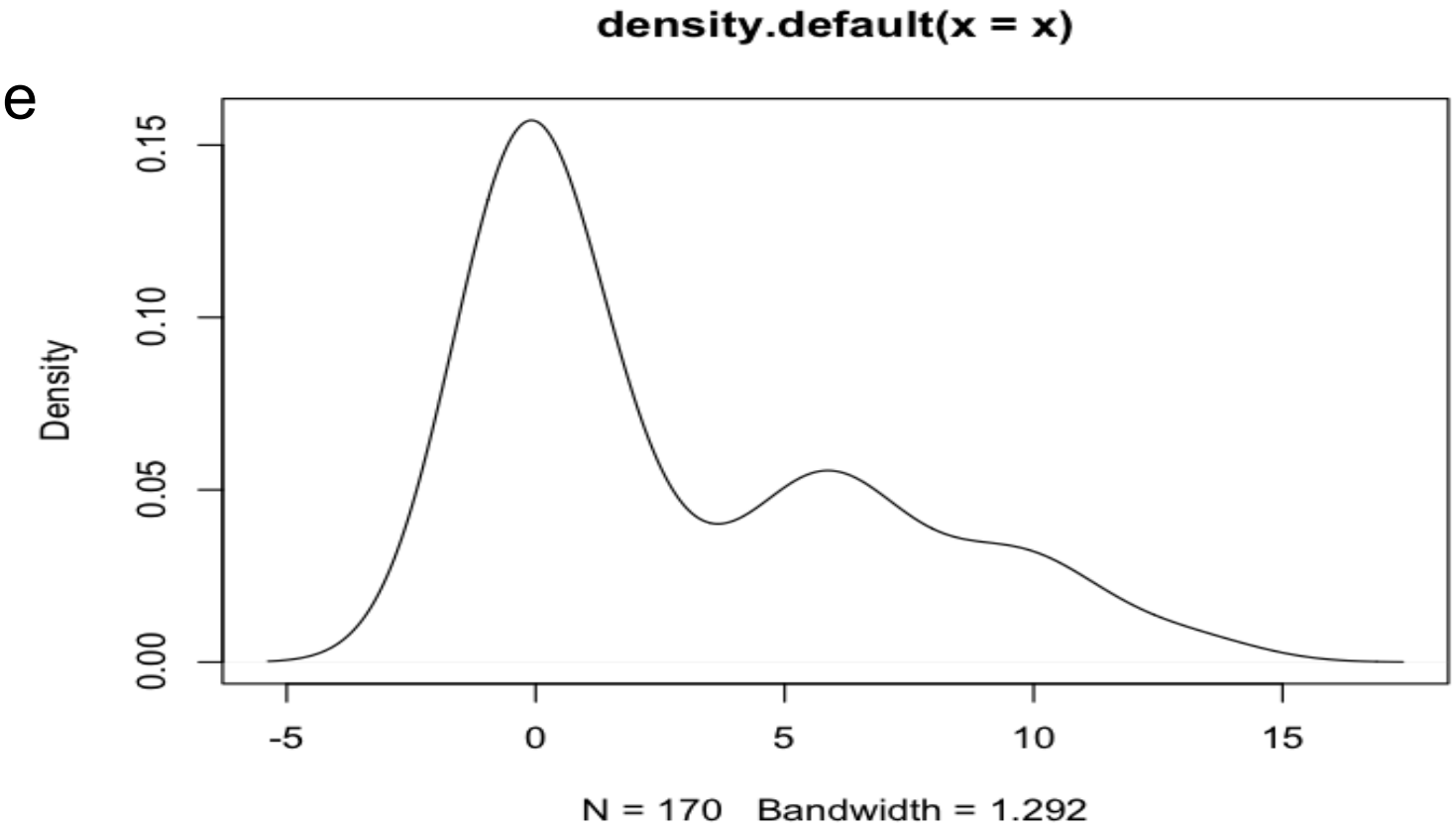
So the ith point is in a pretty tight cluster

Trying out Silhouette with simulated data

Example 1: simulated data from mixture of 3 Gaussians of n=170
N(0, 1), N(5, 4), and N(8, 9), with proportions 0.58, 0.17, and 0.23

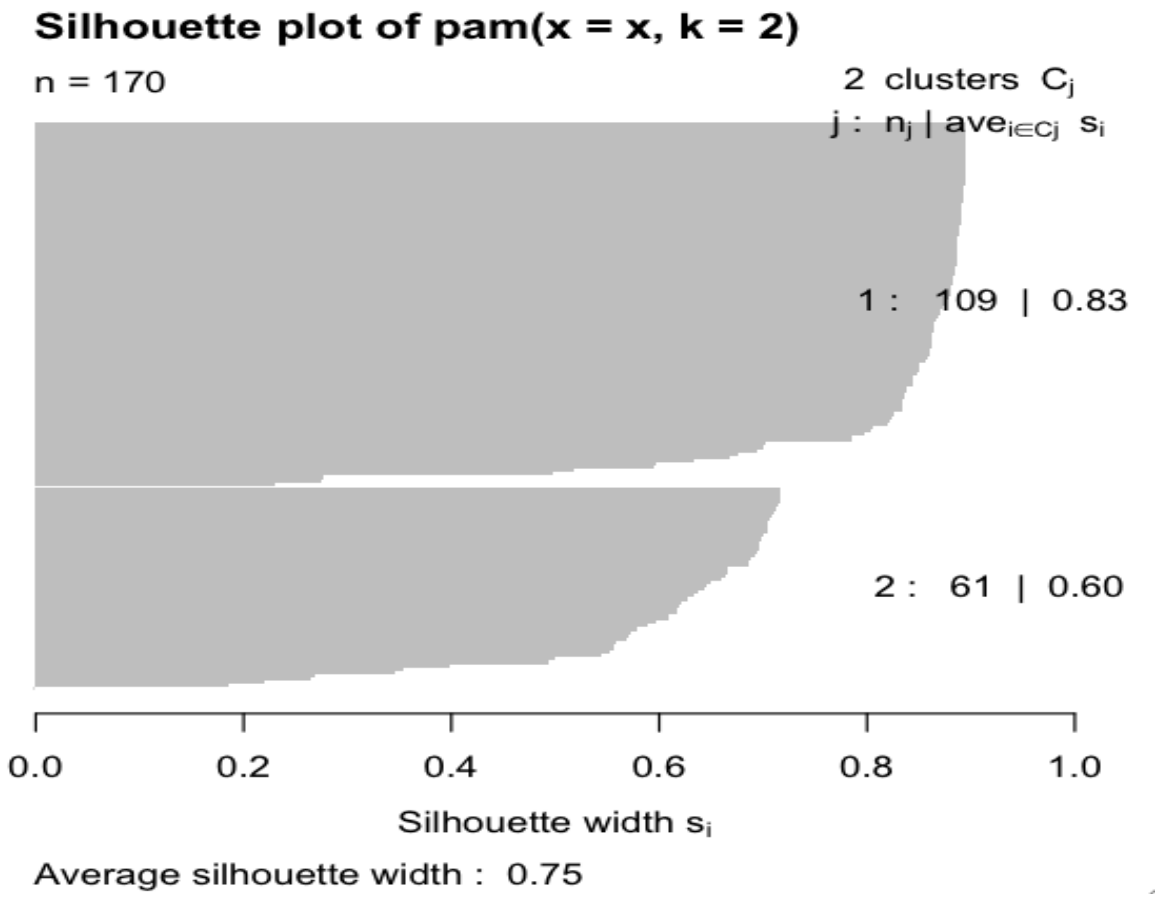
Note that the SDs of the three Gaussian components are 1, 2, and 3, respectively.

The third Gaussian centered at 8 is not very visible.



Trying out Silhouette with PAM results for K=2

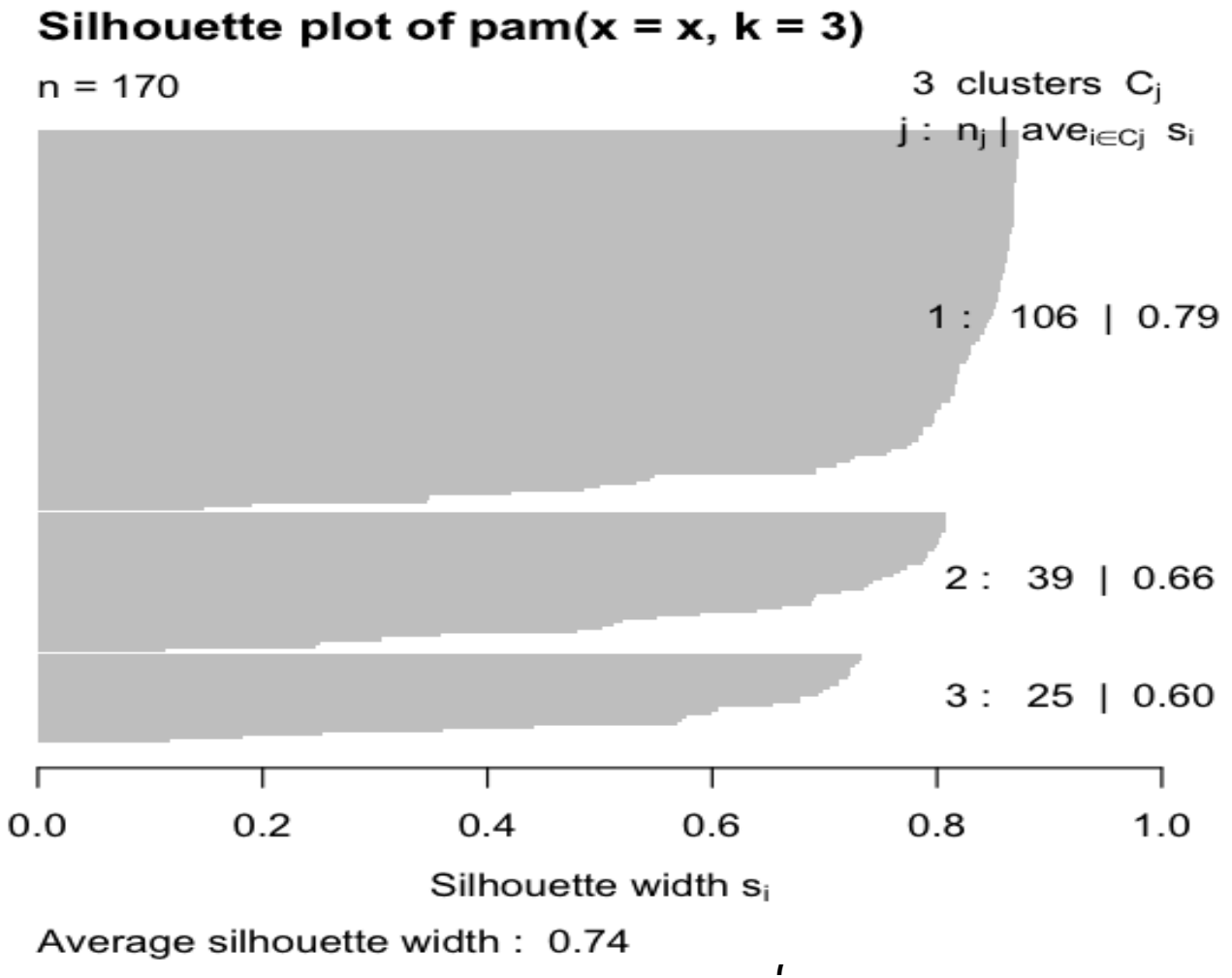
Example 1: Silhouette for the results of two-clusters using **PAM**, which is the squared Euclidean K-means with the K-means centers as the data points closest to centers



Trying out Silhouette with data for PAM with K=3

Example 1: Silhouette for the results of 3-clusters using pam

The average silhouette width is a bit worse than the two-cluster result and because 2-cluster is simpler so I would prefer the previous two-cluster result.



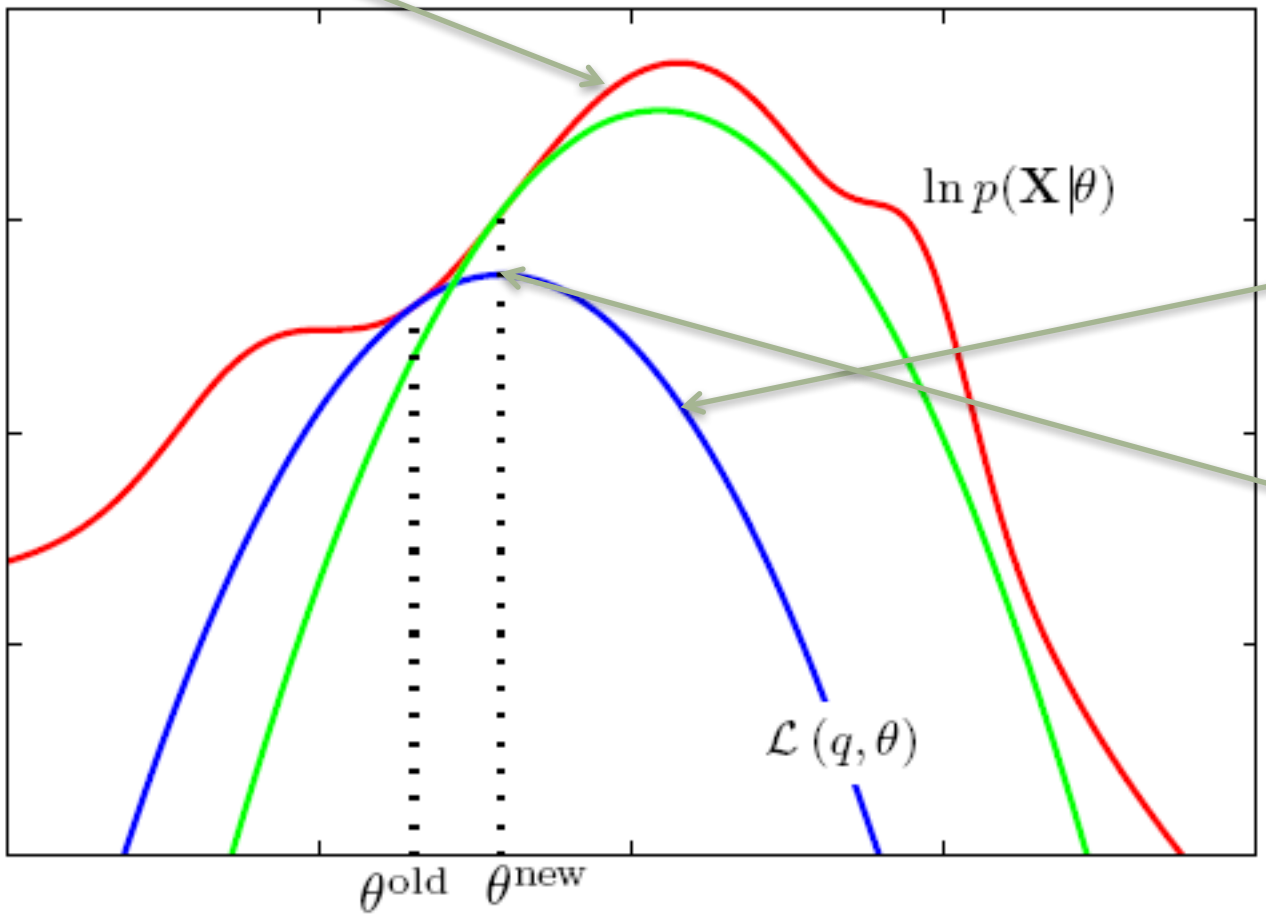
Model based clustering: Gaussian mixture models

- Applications: speech recognition with the hidden Markov Models (HMMs)
- Given iid data, no closed form for maximum likelihood.
- Numerical optimization routines can be used or the EM algorithm can be used.

The EM (Expectation-Maximization) algorithm is an iterative algorithm as gradient descent, however, the E- and M-steps have statistical meanings. The EM framework utilizes some missing data notation.

The EM algorithm is effective, when the maximum likelihood (M-step) is easy to solve with complete data (the missing data plus the observed data), and when the missing data is easy to impute (E-step). Modern generalizations of EM use approximations of various forms in the M- and E- steps.

EM (Dempster et al , 1977) to maximize the red curve – log likelihood function



At a current estimate point,

- E-step

makes a (lower bound blue) curve (in a systematic way) on the log likelihood function

- M-step

maximizes the blue curve for the next estimate

- now on the green curve

- ..

Example: Mixture of 2 Gaussians

$$\pi N(\mu_1, \sigma_1^2) + (1-\pi) N(\mu_2, \sigma_2^2)$$

How to simulate a random variable with this distribution?

1. Simulate a Bernoulli variable Z with prob (Z=1)=π
2. If Z=1, simulate X from $N(\mu_1, \sigma_1^2)$; if Z=0, simulate from $N(\mu_2, \sigma_2^2)$

We only observe X so Z is called the missing (membership) variable or hidden variable.
(X,Z) is called complete data; X – incomplete data

Given iid data $X_1, ..., X_n$

- Suppose $Z_1, ..., Z_n$ are the corresponding missing variables

- If we know the labels or have complete data, it is easy to estimate the parameters in an obvious way

$$\hat{\pi} = \frac{\sum_i Z_i}{n}$$

$$\hat{\mu}_1 = \frac{\sum_i \hat{Z}_i X_i}{\sum_i \hat{Z}_i} \qquad \hat{\sigma}_1^2 = \frac{\sum_i Z_i (X_i - \hat{\mu}_1)^2}{\sum_i Z_i}$$

$$\hat{\mu}_2 = \frac{\sum_i (1 - Z_i) X_i}{\sum_i (1 - Z_i)} \qquad \hat{\sigma}_2^2 = \frac{\sum_i (1 - Z_i)(X_i - \hat{\mu}_2)^2}{\sum_i (1 - Z_i)}$$

EM estimates soft memberships

- If we know guesses of the parameters $\hat{\pi}, \hat{\mu}_1, \hat{\sigma}_1^2, \hat{\mu}_2, \hat{\sigma}_2^2,$ denote the two Gaussian densities $\hat{\phi}_1 \sim N(\hat{\mu}_1, \hat{\sigma}_1^2) \quad \hat{\phi}_2 \sim N(\hat{\mu}_2, \hat{\sigma}_2^2)$

E-step estimates Z's

$$\hat{Z}_i = \frac{\hat{\pi}\hat{\phi}_1(X_i)}{\hat{\pi}\hat{\phi}_1(X_i) + (1 - \hat{\pi}\hat{\phi}_2(X_i))}$$

M-step uses soft membership $\hat{Z}_i$ in (0,1) in the complete data maximum likelihood estimate $\hat{\pi} = \frac{\sum_i \hat{Z}_i}{n}$

$$\hat{\mu}_1 = \frac{\sum_i \hat{Z}_i X_i}{\sum_i \hat{Z}_i} \qquad \hat{\sigma}_1^2 = \frac{\sum_i \hat{Z}_i (X_i - \hat{\mu}_1)^2}{\sum_i \hat{Z}_i}$$

$$\hat{\mu}_2 = \frac{\sum_i (1 - \hat{Z}_i) X_i}{\sum_i (1 - \hat{Z}_i)} \qquad \hat{\sigma}_2^2 = \frac{\sum_i (1 - \hat{Z}_i)(X_i - \hat{\mu}_1)^2}{\sum_i (1 - \hat{Z})_i}$$

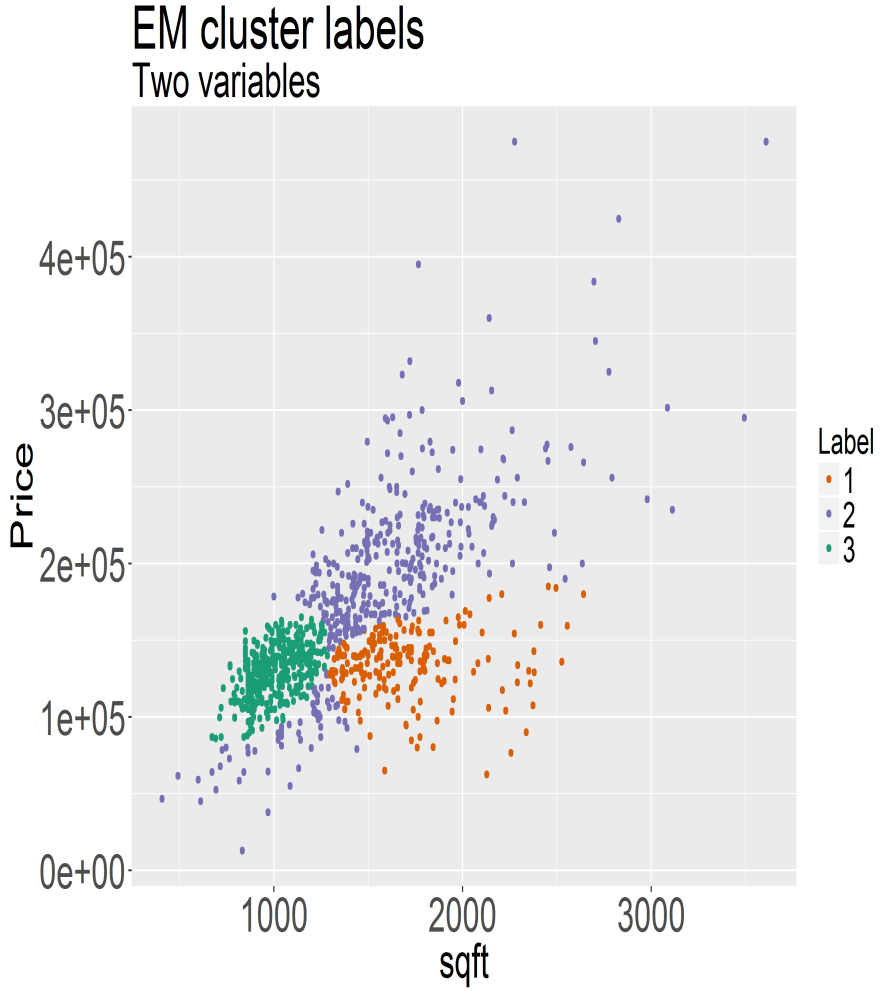
EM for two component Gaussian mixture

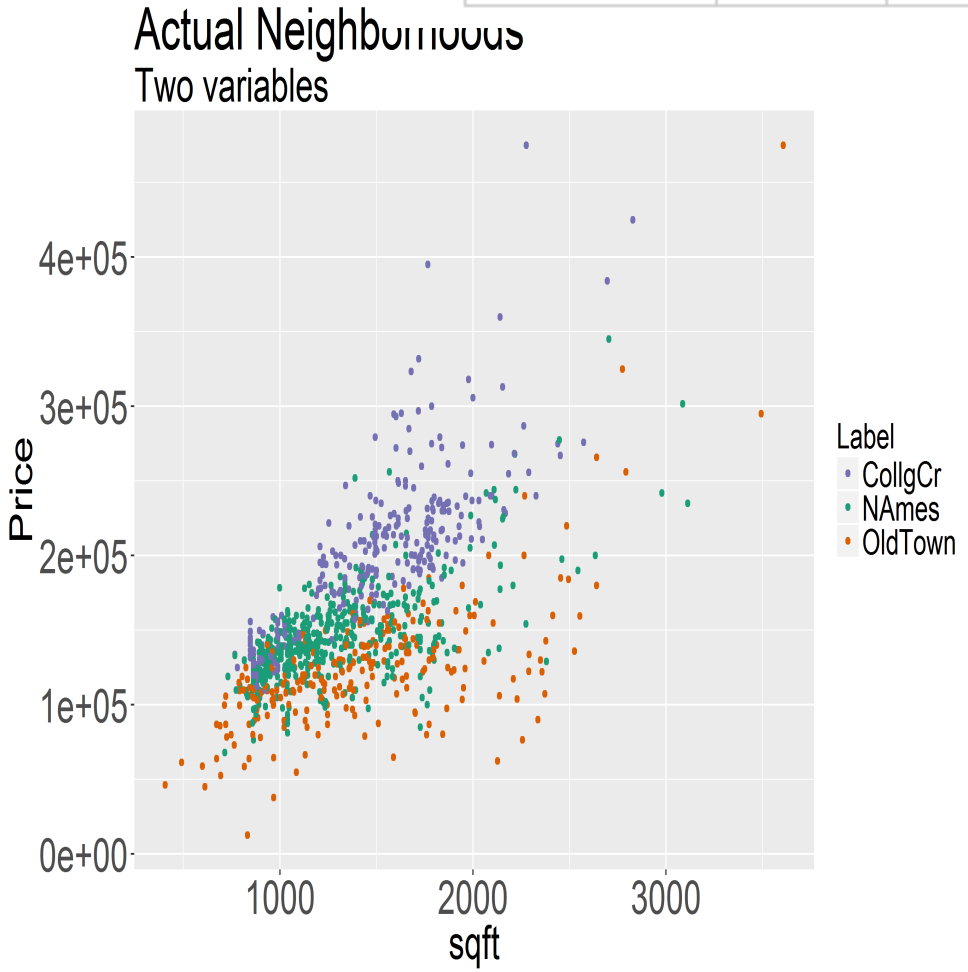
- Start with initial estimates of the 5 parameters
- Iterate between the E- and M-steps from previous slide until convergence
- It is guaranteed to converge to a local minimum of the likelihood function of the observed or incomplete data $X_1, ..., X_n$

Back to Ames data with EM

better results than K-means for CollgCr, but worse for the other two

	1	2	3
OldTown	0.46	0.26	0.27
CollgCr	0.00	0.76	0.24
NAmes	0.19	0.28	0.53

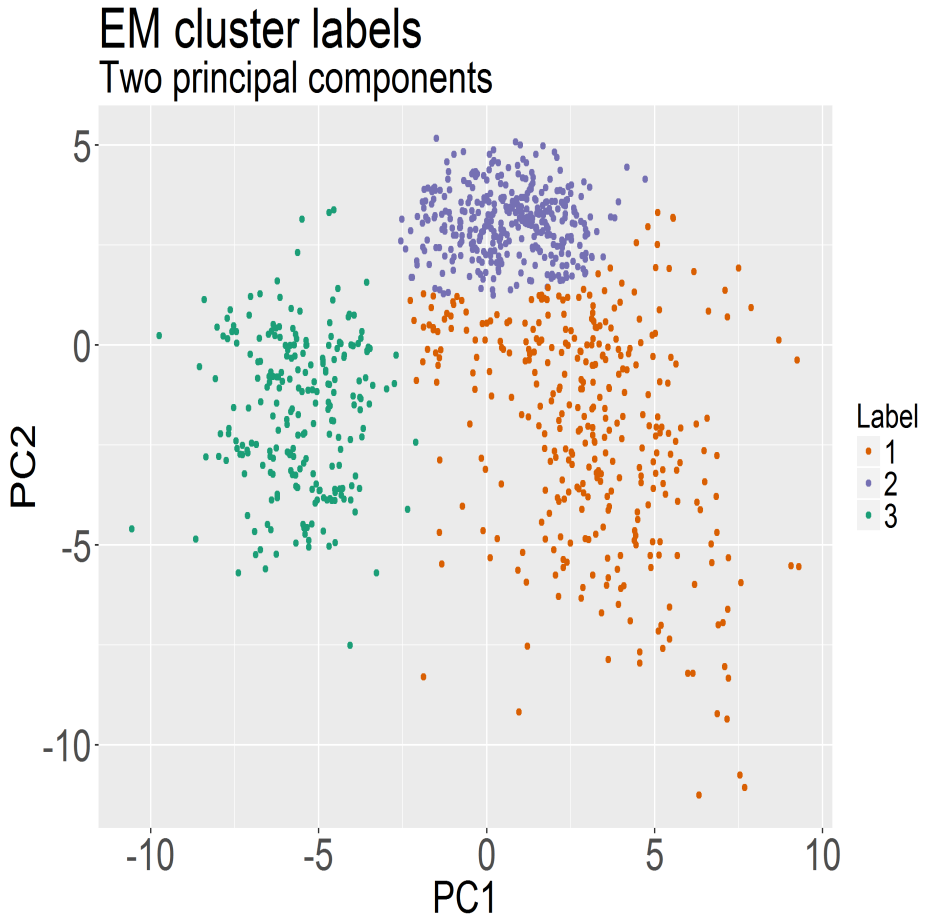


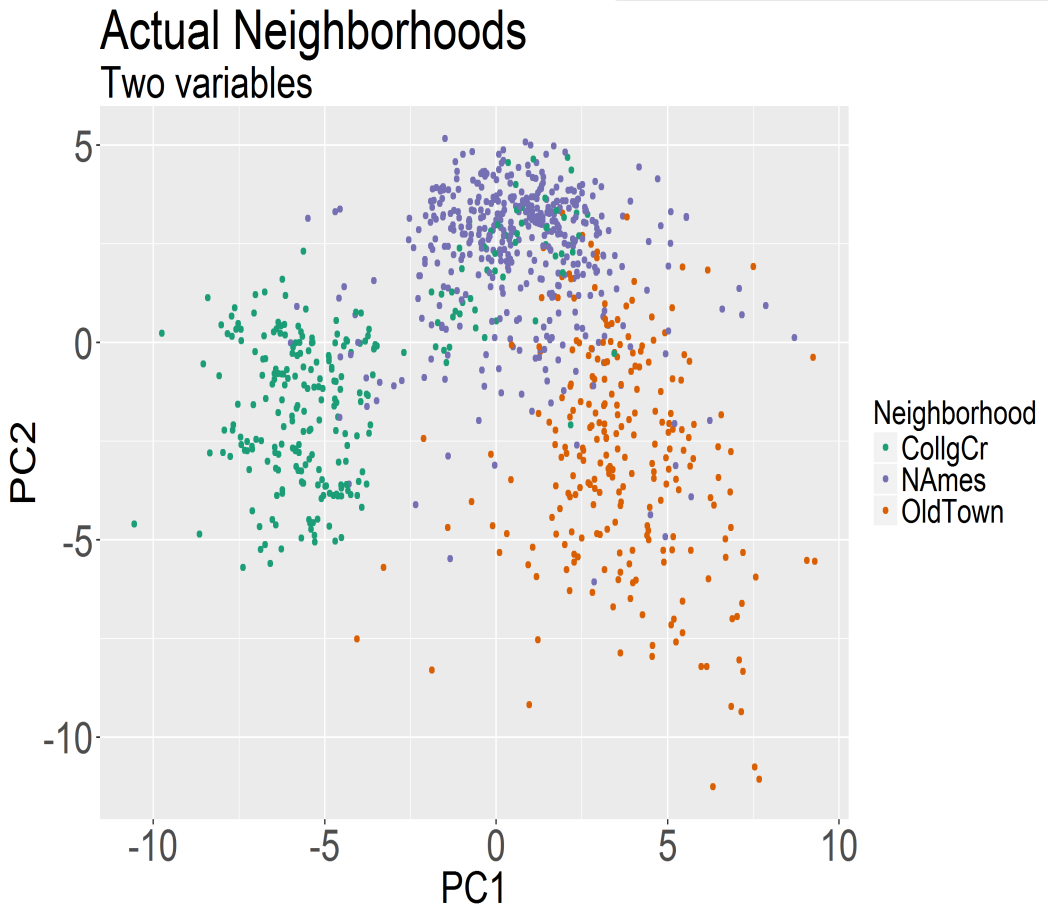


EM on the first 2 PCs

PCA often makes data more Gaussian with better results than using raw data; similar with K-means for OldTown, worse than K-means for the other two.

	1	2	3
OldTown	0.95	0.04	0.01
NAmes	0.23	0.72	0.05
CollgCr	0.07	0.14	0.79

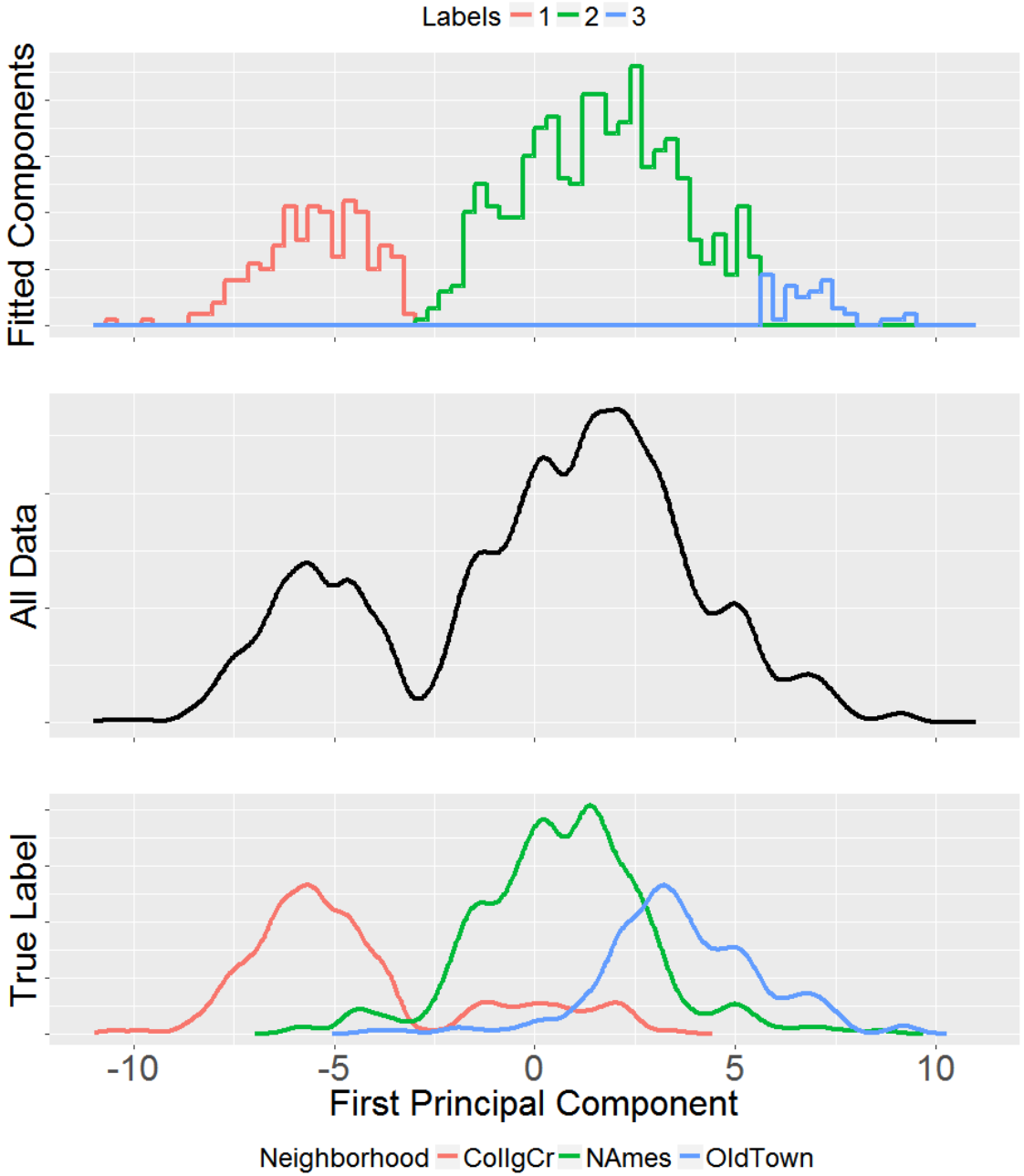




EM on first PC only

- OldTown got worse—other two are similar.
- Note that PC didn't know we were going to do clustering – PC has a different goal.
- Not much info in data for EM to differentiate the old town or the third component.

	1	2	3
CollgCr	0.79	0.21	0.00
NAmes	0.04	0.95	0.01
OldTown	0.01	0.84	0.15



Hierarchical clustering

For a hierarchical clustering algorithm, we need

- a distance or dissimilarity measure between any two points.

- a rule to calculate the distances/dissimilarities between disjoint clusters of objects. This between cluster distance can generally be calculated directly from the distances of the various elements involved in the clustering.

Hierarchical clustering

For a hierarchical clustering algorithm, we need

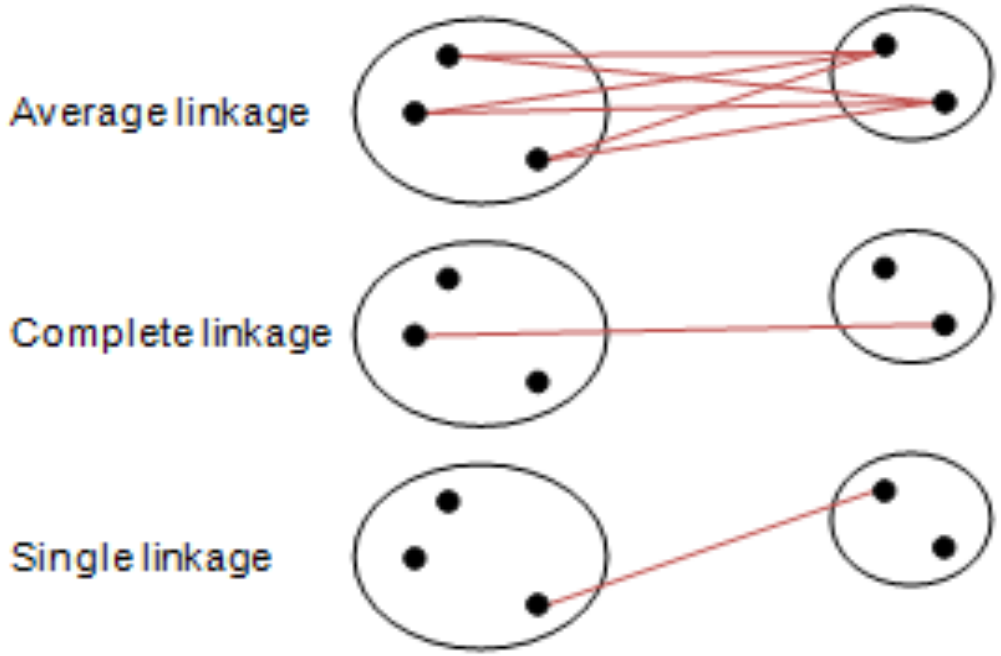
- A. a distance or dissimilarity measure between any two points.

- B. a rule to calculate the distances/dissimilarities between disjoint clusters of objects.

  This between cluster distance can generally be calculated directly from the distances of the various elements involved in the clustering.

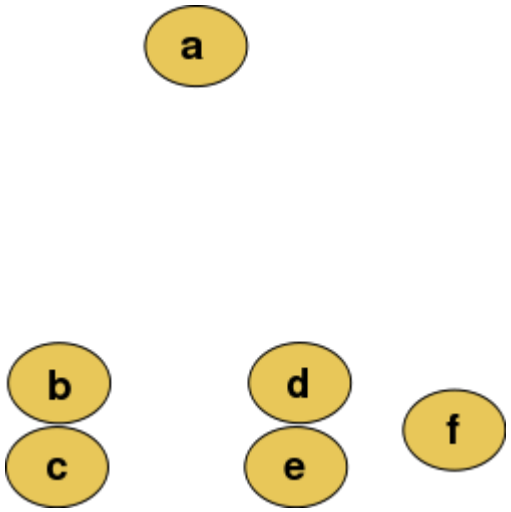
Three B. rules

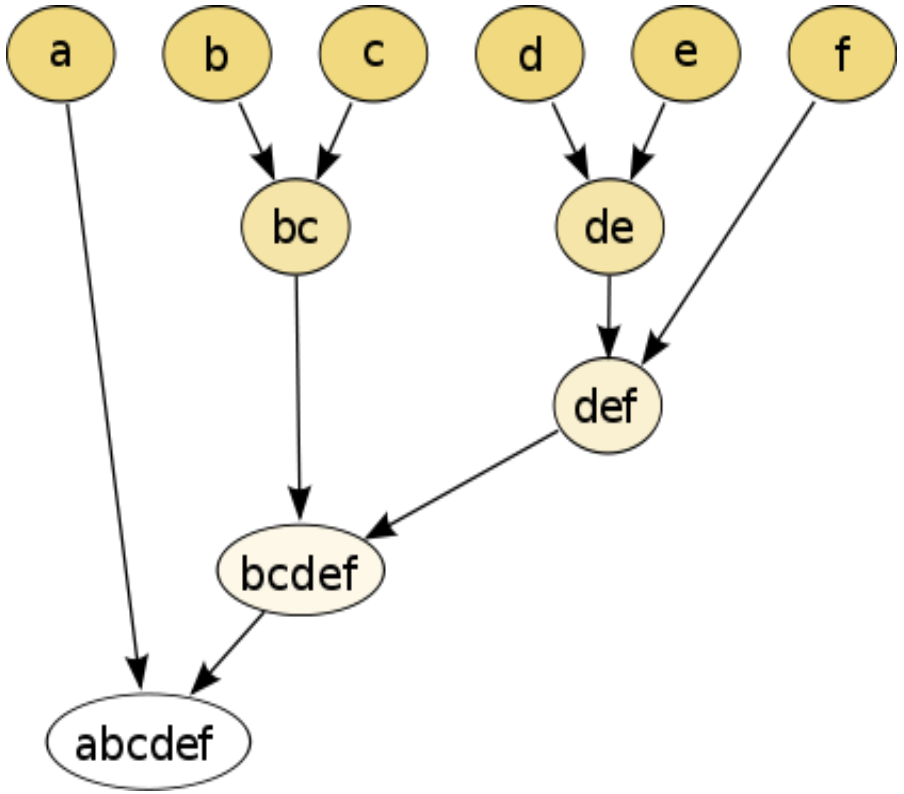
- **Average linkage**:
  average dissimilarities between two clusters
- **Complete-linkage:**
  farthest points between the two clusters
- **Single-linkage:**
  smallest dissimilarity between the two clusters



Wiki example

Raw 2-dim data





1. Take the two most similar
2. points and merge them into a new (super) point – we now have n-1 points

B-rule allows us to have a dissimilarity measure for the n-1 points

3. Continue until all points are grouped.

Remarks

- Hierarchical algorithms are based on common sense rather than formalized theory. We call points either the objects to be clustered or the clusters of objects generated by the algorithm.

- The family of clusters built by such an ascending algorithm forms a so-called hierarchy. This family has the property of having the whole set and also every one of the object separately. By cutting the tree of a hierarchical clustering tree, we get a partition in which the number of clusters increases as the cut-point approaches the initial points. A hierarchy allows us to obtain a set of n nested partitions containing 1 to n groups/clusters.

Word2vec https://www.tensorflow.org/tutorials/word2vec

Word2Vec is a class of algorithms (two-layer neural networks) that produce ”word embeddings” in an Euclidean space

e.g. each word is represented by a 300 dimensional vector

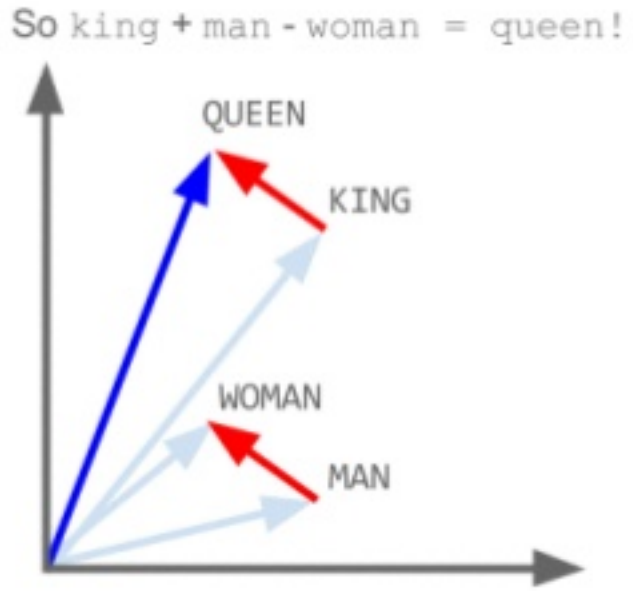
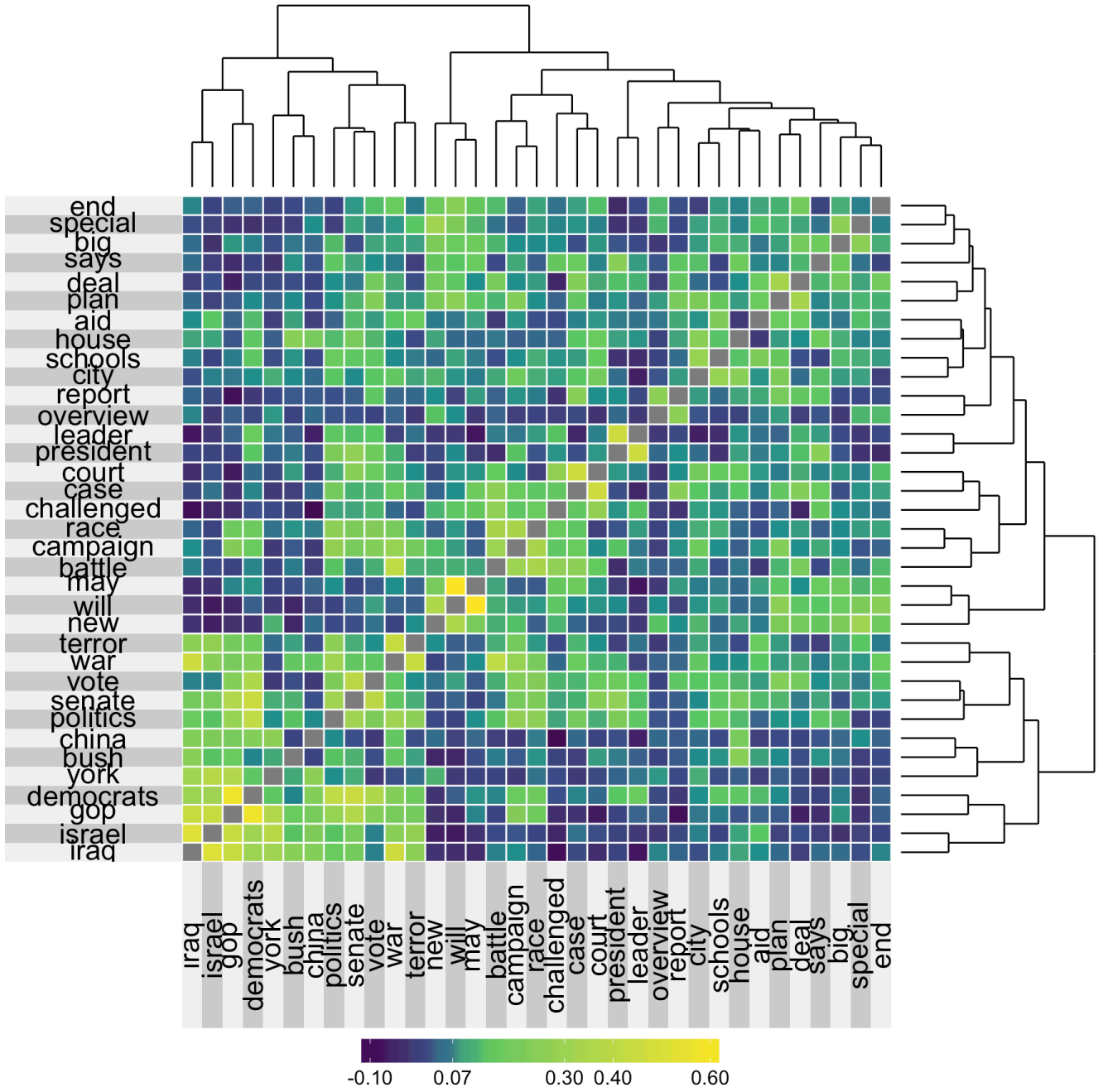


Image source: http://book.paddlepaddle.org/04.word2vec/index.en.html

Hierarchical clustering of word vectors

Visualizing a subset of the word vector *correlations* between the most popular words from the Google News Corpus.

Cluster using hierarchical clustering using the “complete *linkage* (or farthest neighbor clustering)” method.



GoogleNews word2vec data:
https://code.google.com/archive/p/word2vec/

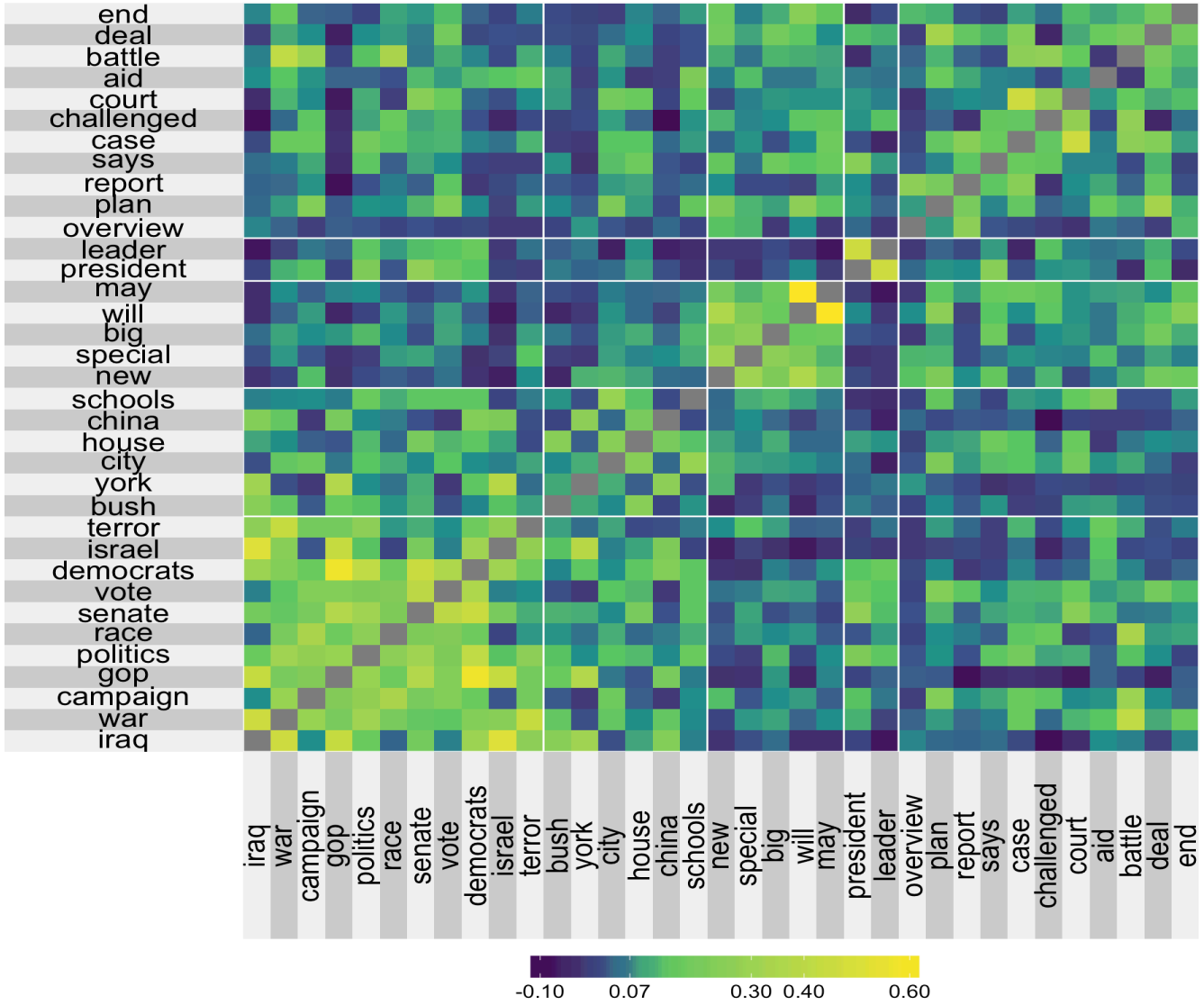
Superheat R-package (Barter and Yu, 2017):
https://arxiv.org/pdf/1512.01524.pdf

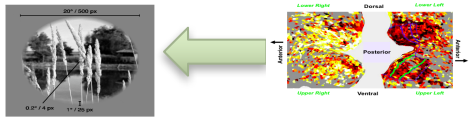
Compare with K-means

Must specify number of clusters, e.g. k = 5

Clusters changed a bit…

Other info is needed to say which is better…





Movie reconstruction

Nishimoto, Vu, Naselaris, Benjamini, Yu and Gallant (2011)

Presented clip

Clip reconstructed from brain activity

You are ready to peak in the reconstruction box…

Movie reconstruction using fMRI signals

Neuroscience

S. Nishimoto

J. Gallant

A. Vu

T. Naselaris

Statistics

Y. Benjamini

B. Yu

Current Biology 21, 1641–1646, October 11, 2011 ©2011 Elsevier Ltd All rights reserved DOI 10.1016/j.cub.2011.08.031

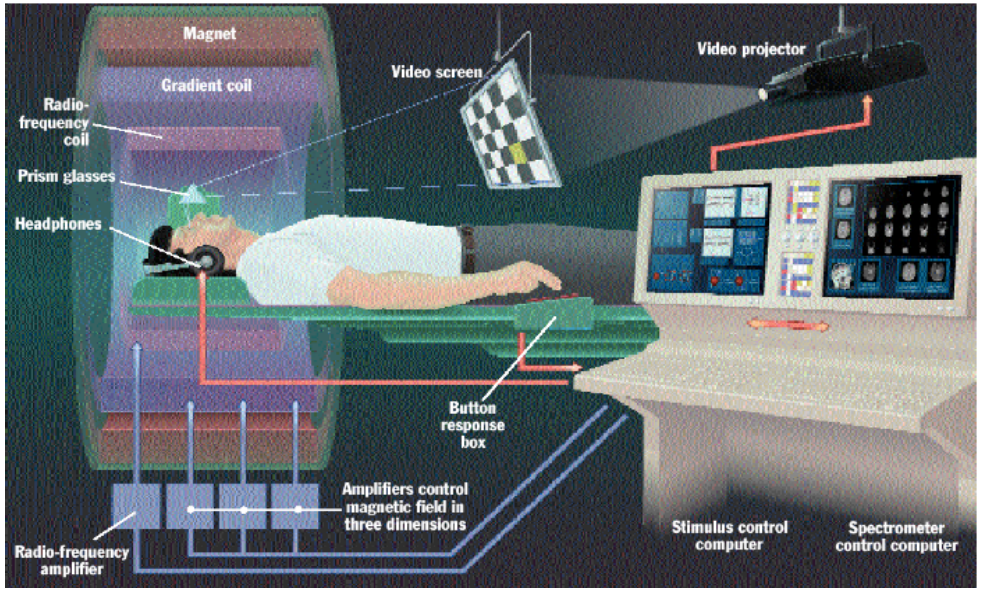
Report

Reconstructing Visual Experiences from Brain Activity Evoked by Natural Movies

Data collection
(the Gallant Lab)

- **Non invasive and indirect recording technique**
- Low temporal resolution, a few seconds
- High spatial resolution
  (**voxel = 1x1x1 mm cube**)
  - 10,000 voxels in early visual areas
  - each voxel covers > 100,000 neurons

- Data: input movie clips
  and corresponding fMRI
  brain signals of subjects



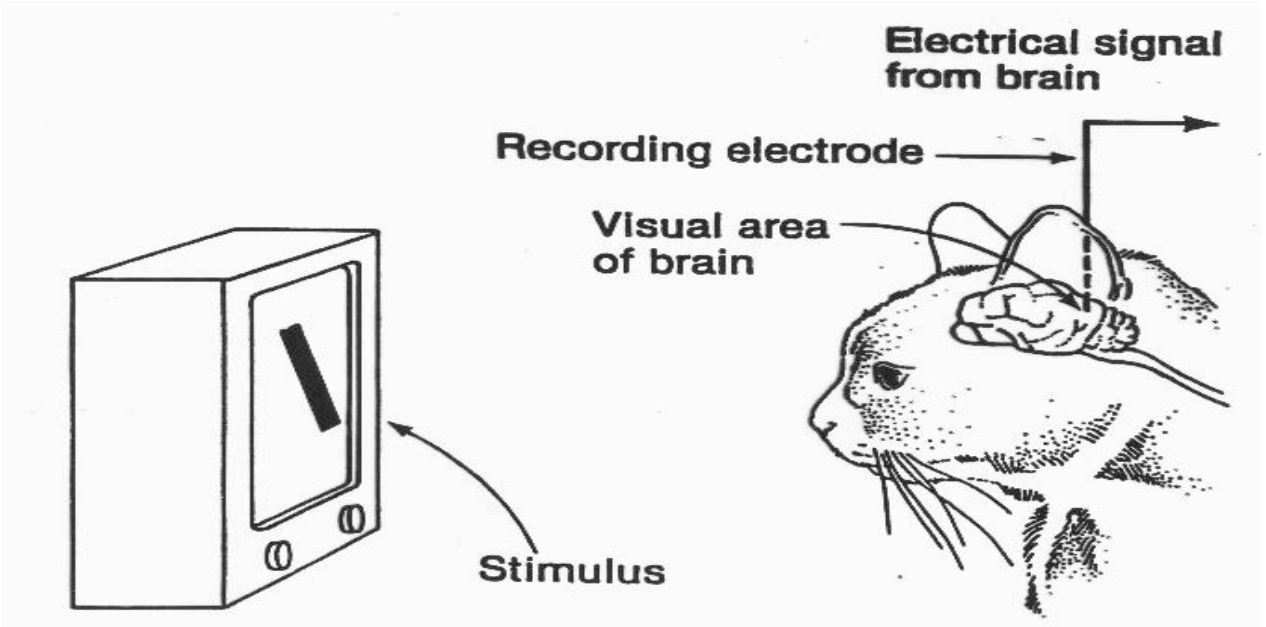
“Kepler”=Hubel and Wiesel (1959)

They discovered, in neuron cells of the primary cortex area V1,

orientation and location selectivity, and

excitatory and inhibitory regions .

Soundification of neuron spike data



Hubel and Wiesel (Nobel Prize,1981)

"The signal message that the eye sends to the brain can be regarded as a secret code to which only the brain possesses the key and can interpret the message. Hubel and Wiesel have succeeded in breaking the code."

-- Presentation Speech by Professor David Ottoson of the Karolinska Institute at the Award Ceremony for

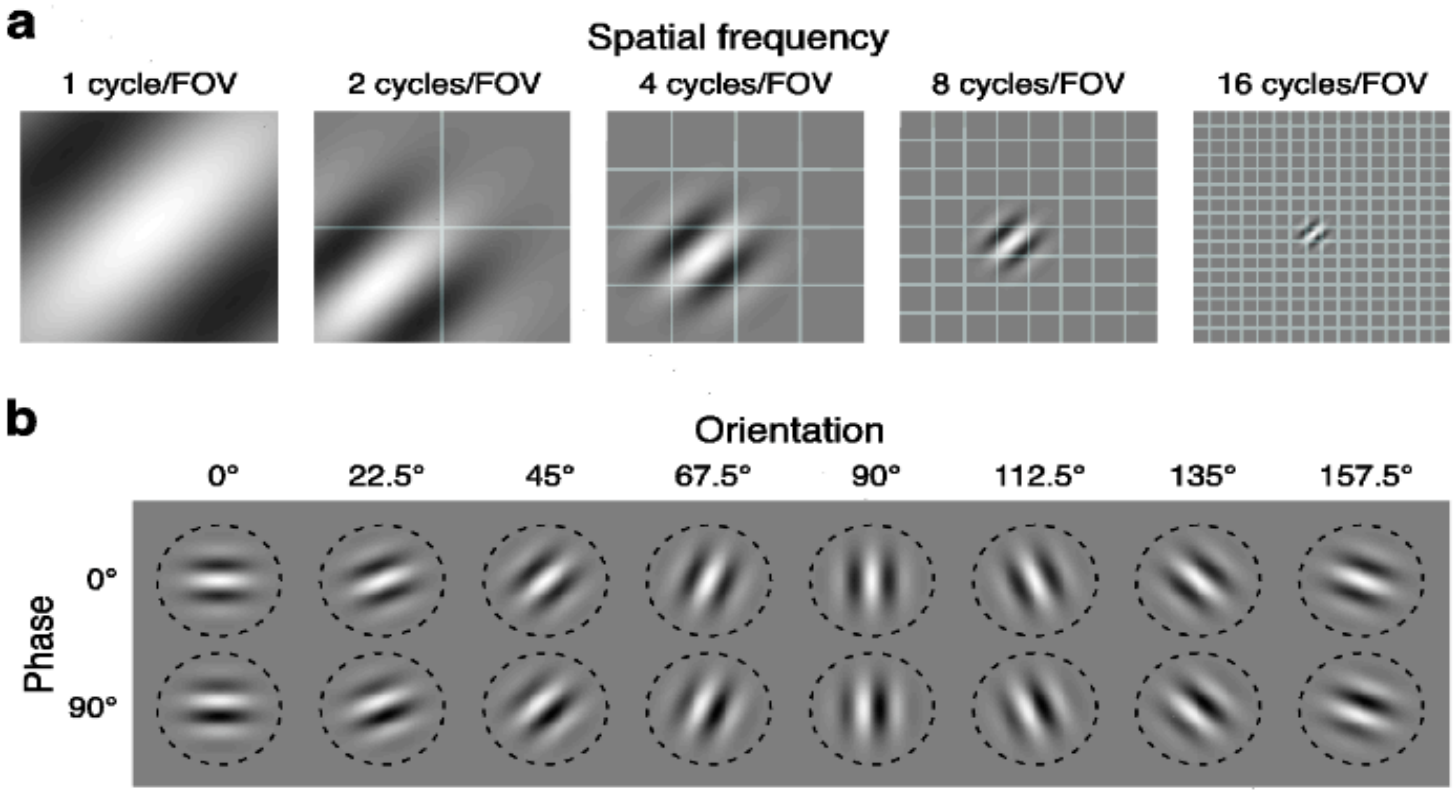
The Nobel Prize in Physiology or Medicine 1981

Roger W. Sperry, David H. Hubel, Torsten N. Wiesel

2-D Gabor filters –

math. representation of edge detectors, 100% reproducible

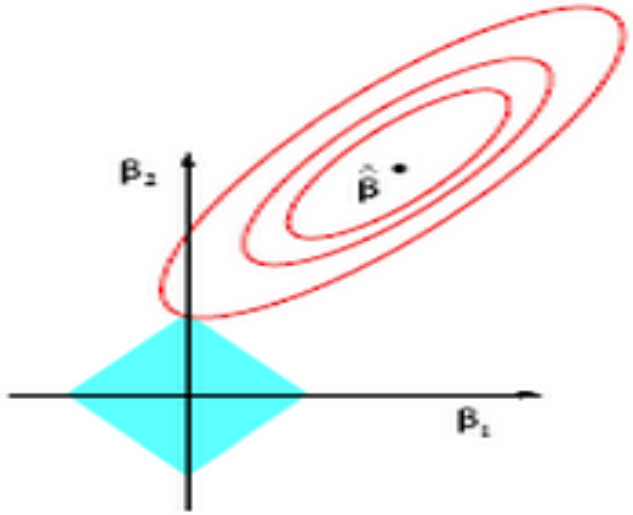
- 2-D Gabor filters corresponding to particular spatial frequencies, locations, orientations (Hubel and Wiesel, 1959,…)



Movie reconstruction algorithm

- Lasso predictive model based on **3-D Gabor features** to predict fMRI signal on a voxel from movies
  (**neuroscience knowledge**; similar to first layer of deep learning)

Lasso = L1 constrained Least Squares
Amount of L1 constraint chosen using CV



- This model and a large movie clip database were used for movie reconstruction

On our way to reconstruct movies…

Using Lasso, for each voxel, we have a good linear prediction rule for fMRI responses corresponding to any movie clips.

What else do we have?

A large movie clip database from movie trailers, YouTube, etc…



….

An “external” memory to approximately “replace” our internal memory?

Given an observed fMRI vector Y over selected "informative" voxels

Lasso-model

clip 1  ----------> predicted fMRI vector Y1

clip 2  ----------> predicted fMRI vector Y2

clip 3  ----------> predicted fMRI vector Y3

:

:

Rank clip 1, clip 2, clip 3, … depending on how close Y1, …, Y3… are to Y in terms of a "weighted LS metric."

Movie reconstruction based on fMRI signals

Take top 100 movie clips and average to give the reconstructed movie clip.

Movie reconstruction results for 3 subjects

Summary

- Silhouette – a graphical method for K selection
- EM – parametric clustering
- Hierarchical clustering (example based on word2vec)
- Movie reconstruction explained

Extra slides on EM (not required for the course)

EM (expectation-maximization)

A concise and clear tutorial on EM is by Sean Borman who seems to have taken it from some unpublished notes by Stuart Russell.

In this tutorial, given an initial estimator θ_n (note n is not the sample size here but the iteration number as in the tutorial by Bowman) of the parameter, Jensen's inequality is used to give a lower bound on the log likelihood function using this initial estimator. This lower bound function takes the same value at θ_n as the log-likelihood L(θ_n).

EM (expectation-maximization)

Maximizing this lower bound function gives the EM algorithm:

- E-step: calculating the lower bound function which is an expectation over the hidden or latent variable. For exponential families, this step amounts to calculating the expected value of the hidden variable given the incom- plete data X and the current parameter estimator θ_n.

- M-step: maximizing the lower bound function to obtain the updated θ_{n+1}.

EM (expectation-maximization)

Because of the construction of the lower bound, the log likelihood value at the updated value θ_{n+1} is lower bounded by the lower bound functiton at at the updated value θ_{n+1}, which is larger than the lower bound function at the initial value θ_n, which is equal to the log-likelihood at the initial value θ_n.

Hence the EM algorithm always increases the log-likelihood and the algorithm converges to a local maximum of the log-likelihood function.