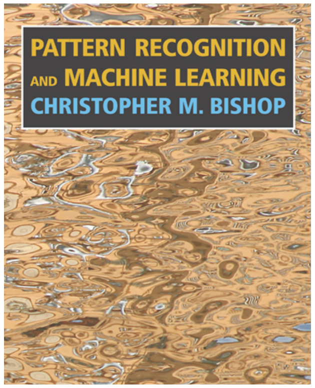
Spike Sorting

Carnegie Mellon

18-698 / 42-632
Neural Signal Processing
Spring 2013
Prof. Byron Yu

Topics we will cover in PRML



Chap. 4: Classification. Naive Bayes.
Neuroscience application: discrete neural decoding

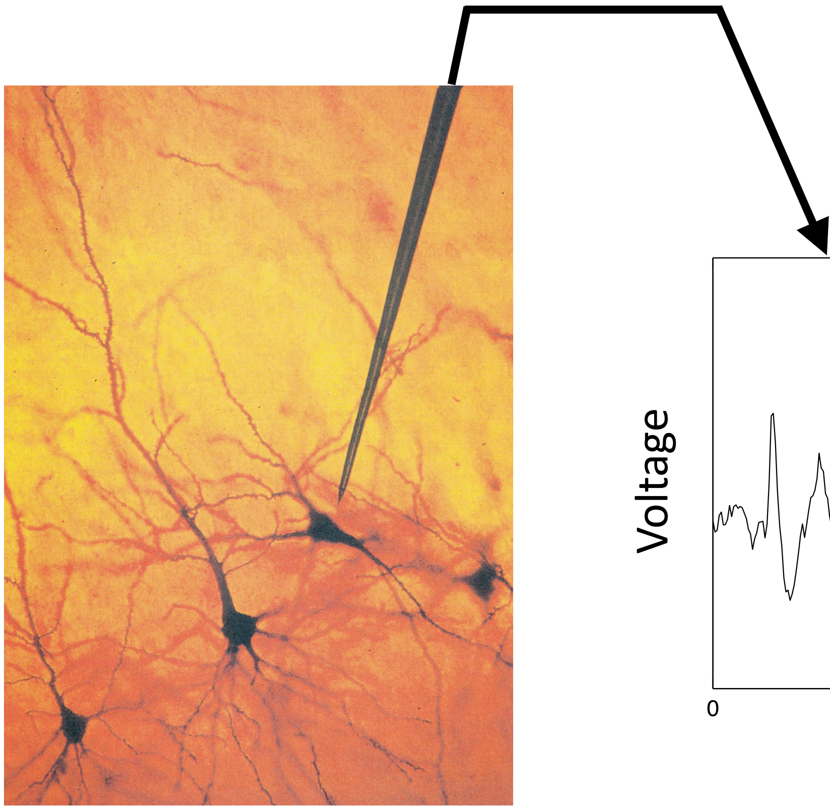
Chap. 8: Graphical models.

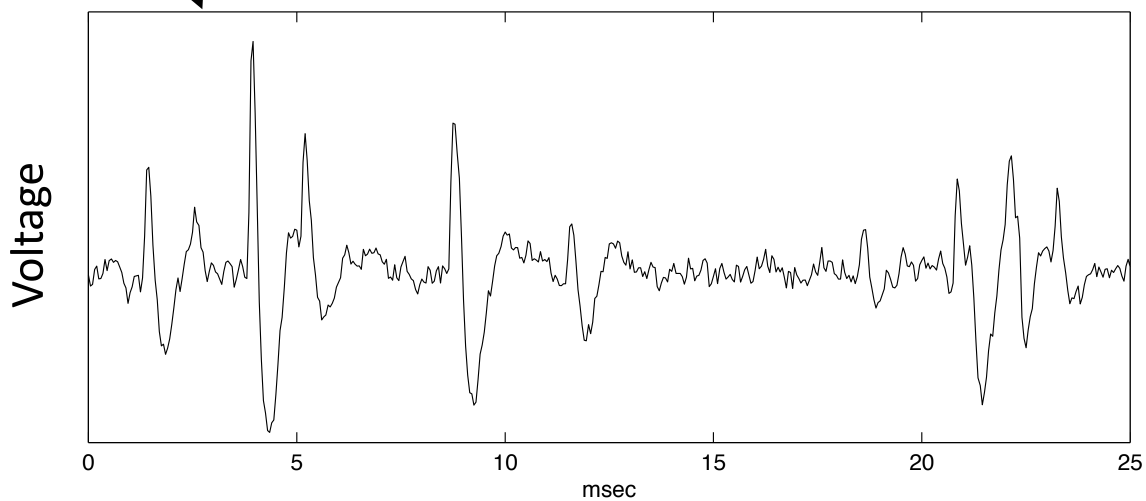
Chap. 9: Mixture models. Expectation-maximization.
Neuroscience application: spike sorting

Chap. 12: Principal components analysis. Factor analysis.
Neuroscience applications: spike sorting, discrete neural decoding

Chap. 13: Hidden Markov model. Kalman filter.
Neuroscience application: continuous neural decoding

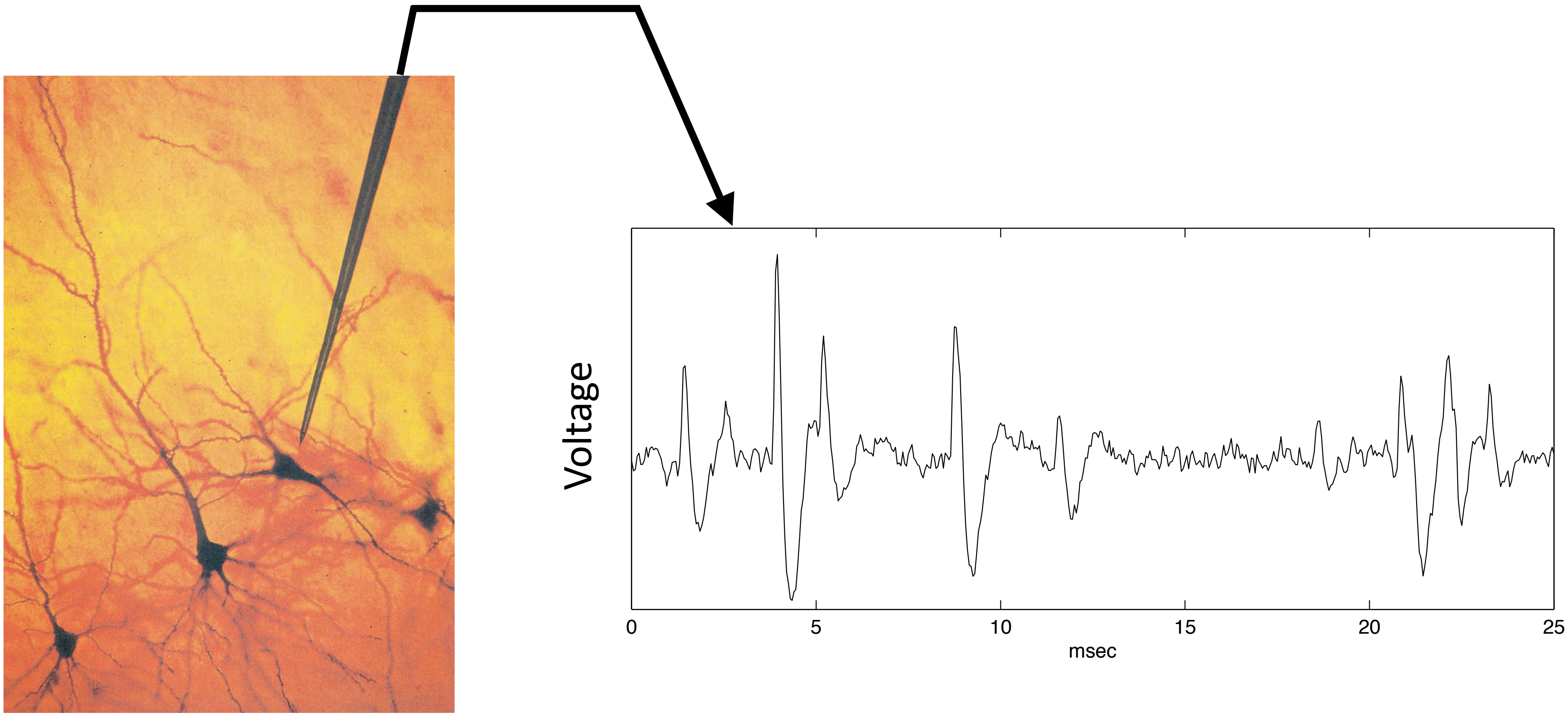
What is spike sorting?





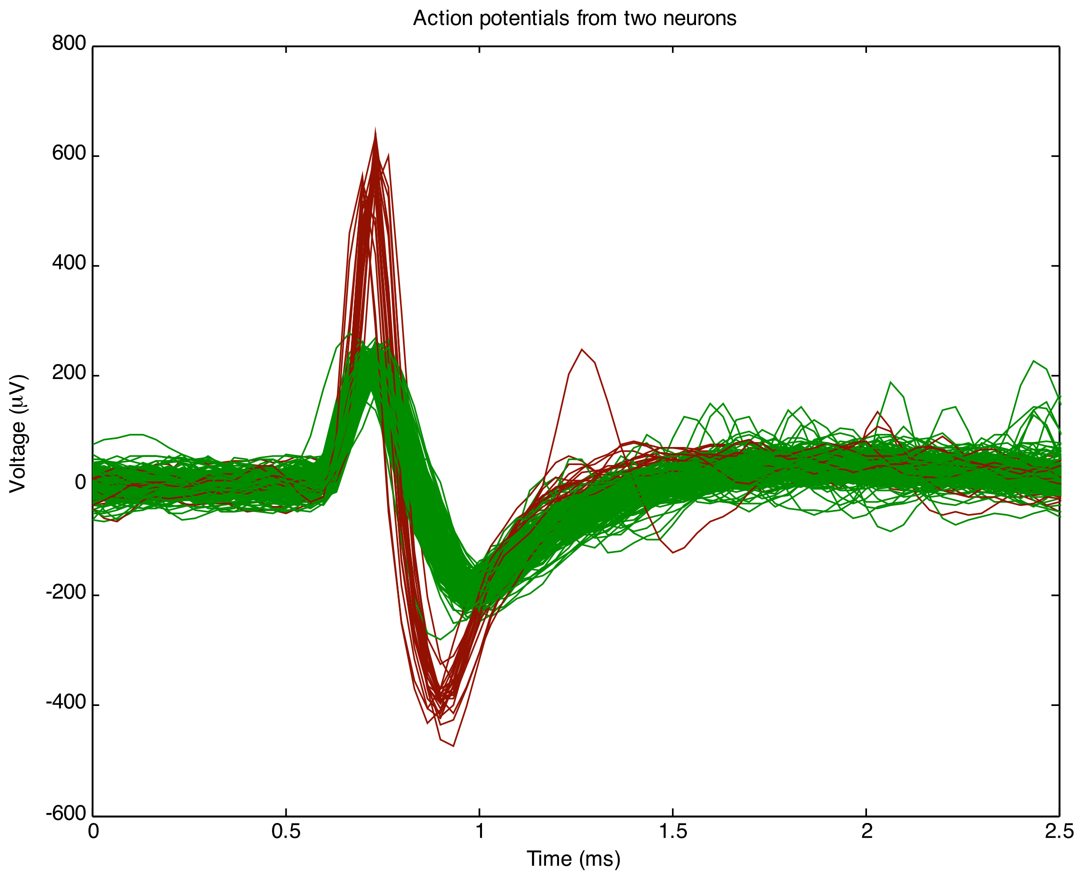
- At what times did action potentials (APs) occur?
- Do APs arise from more than one neuron?
- If so, how many neurons?
- How do we assign an AP to which neuron it came from?

What is spike sorting?



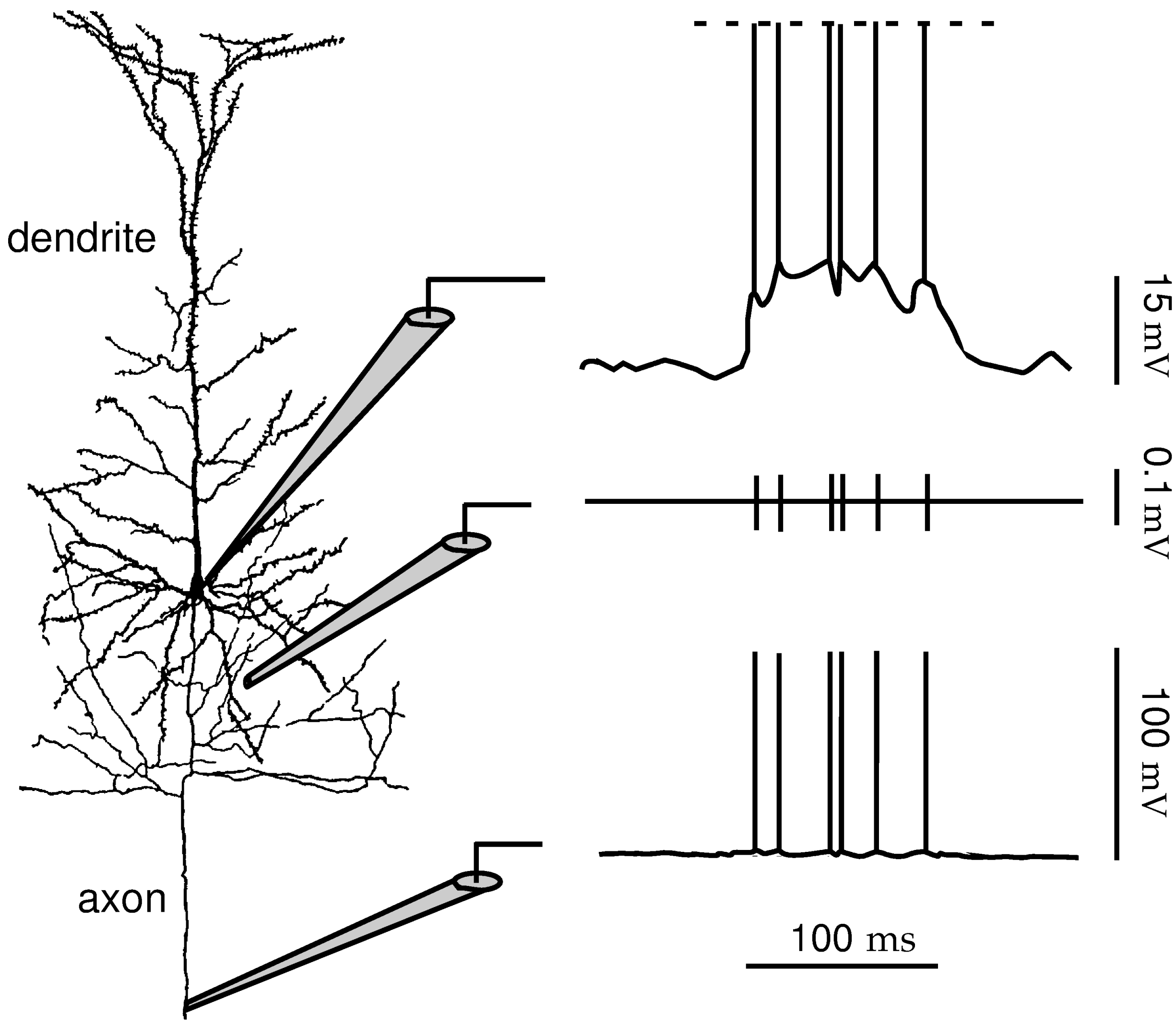
- An electrode typically “hears” APs from more than one neuron.
- Each neuron encodes information differently => we must separate signals.
- Note presence of “background noise”, which makes spike sorting even more challenging.

How can we distinguish spikes from different neurons?



- APs from different neurons usually have different characteristic shapes.
- But I thought you said that AP shape doesn't convey information!

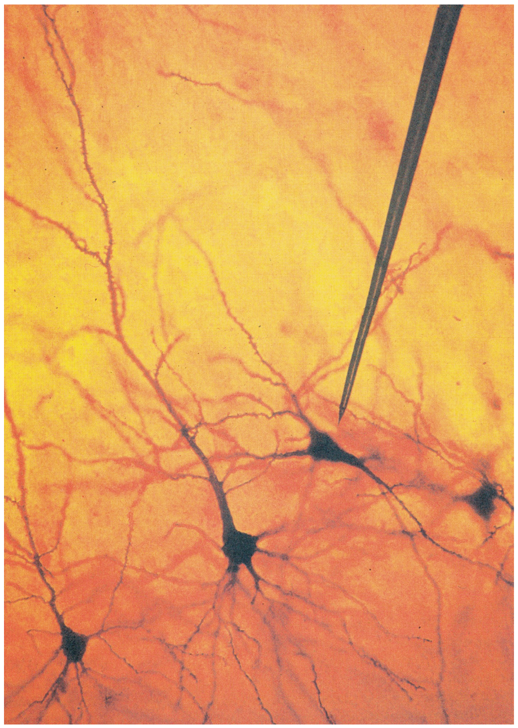
Intra-cellular vs. extra-cellular recordings

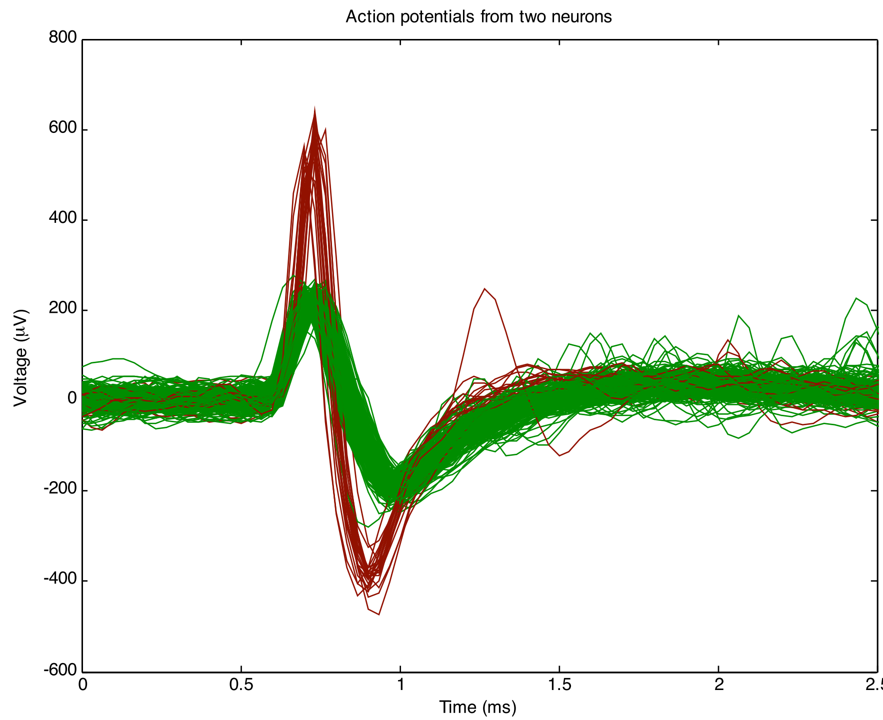


Intra-cellular recording: AP shape <u>similar</u> for different neurons

Extra-cellular recording: AP shape <u>different</u> for different neurons. Why?

Why do APs have different shapes for different neurons?





- Different neurons are located at different distances from electrode.
- This affects the amplitude and shape of AP recorded extracellularly.
- For those who have taken communications theory, think of multiple transmitters (neurons) and a single receiver (electrode).

Spike sorting algorithms

1. Threshold detection

2. Hoop-based spike discrimination

3. Feature analysis, then clustering

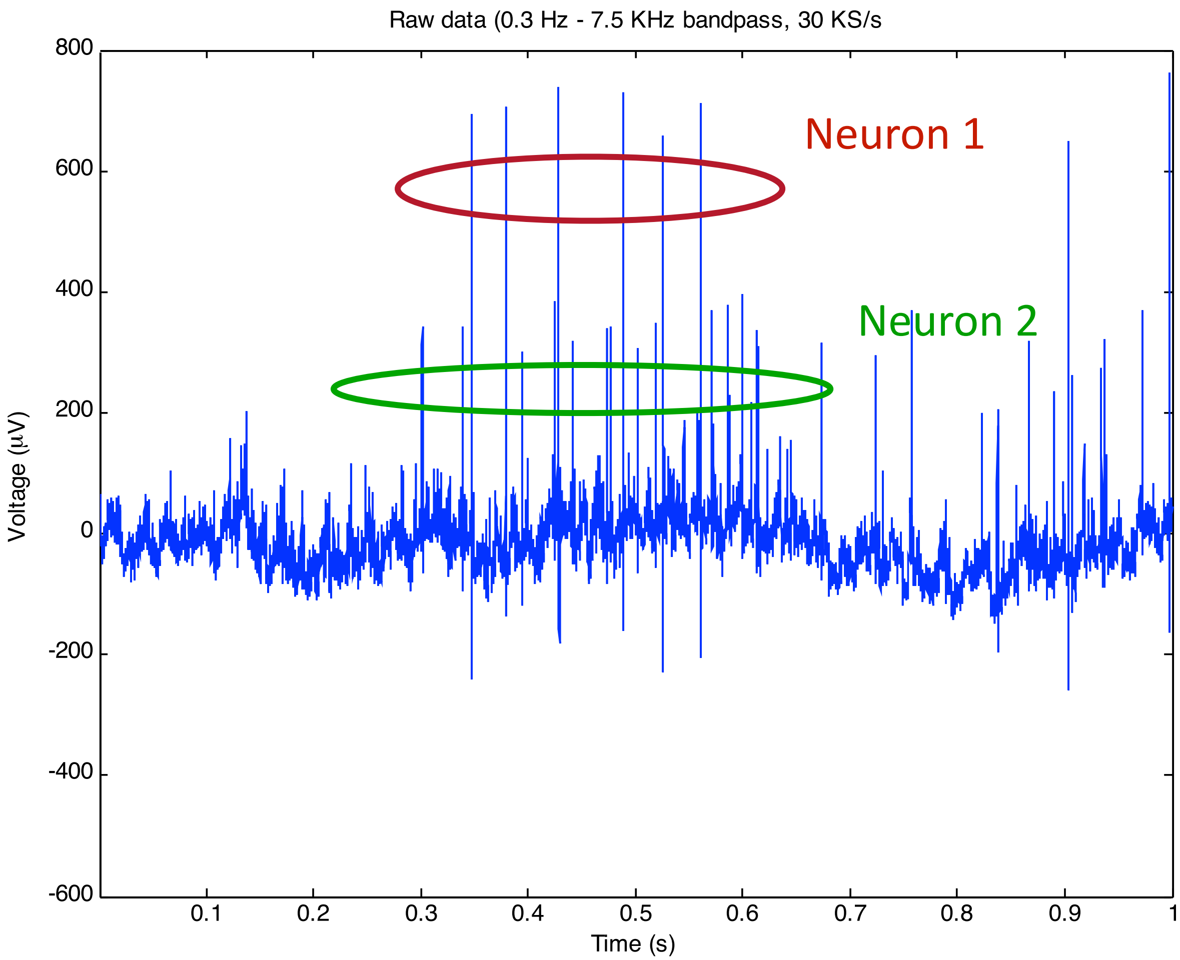
PRML Chapter 9

4. Principal components analysis, then clustering

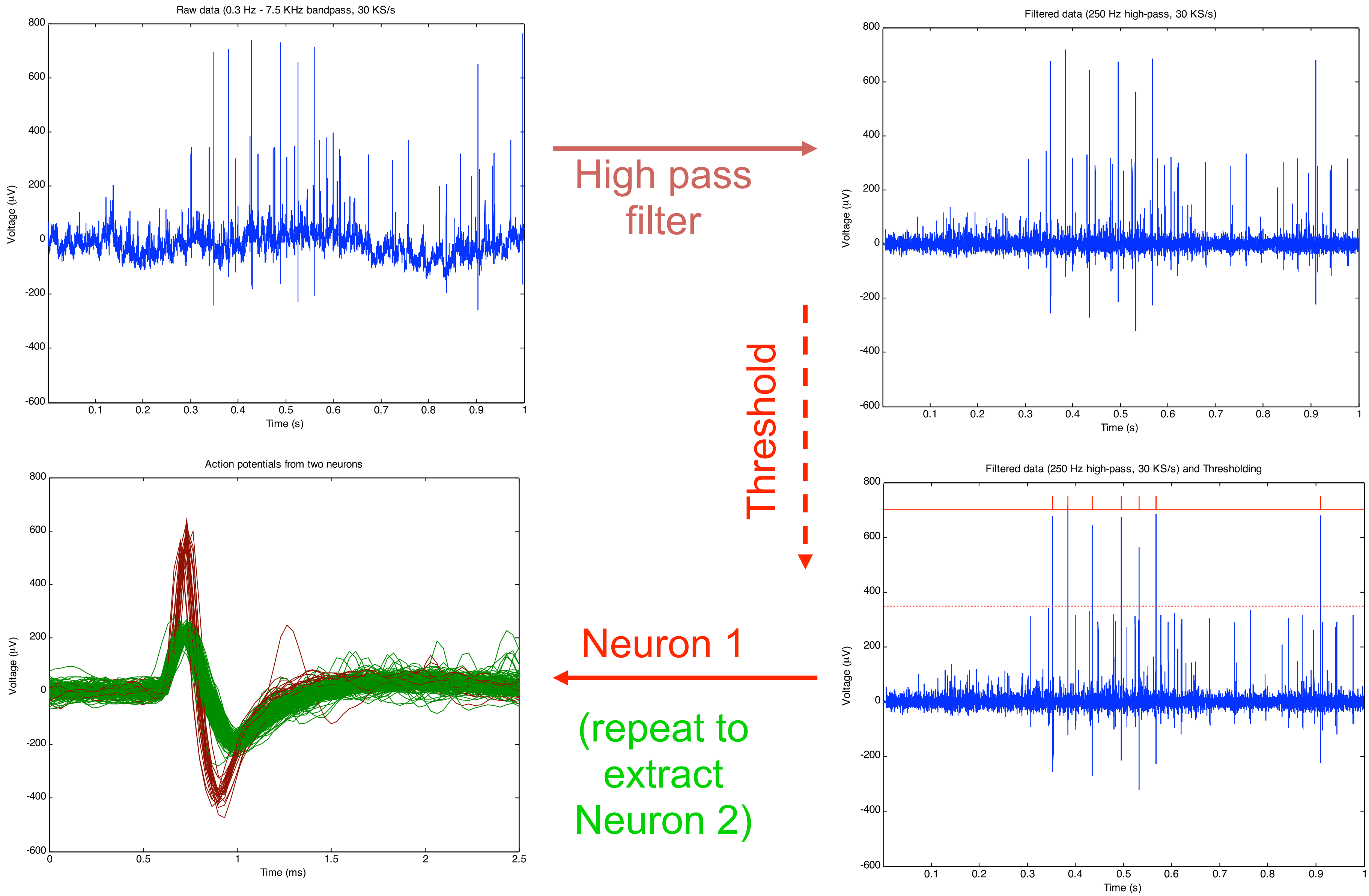
PRML Chapter 12

Threshold detection

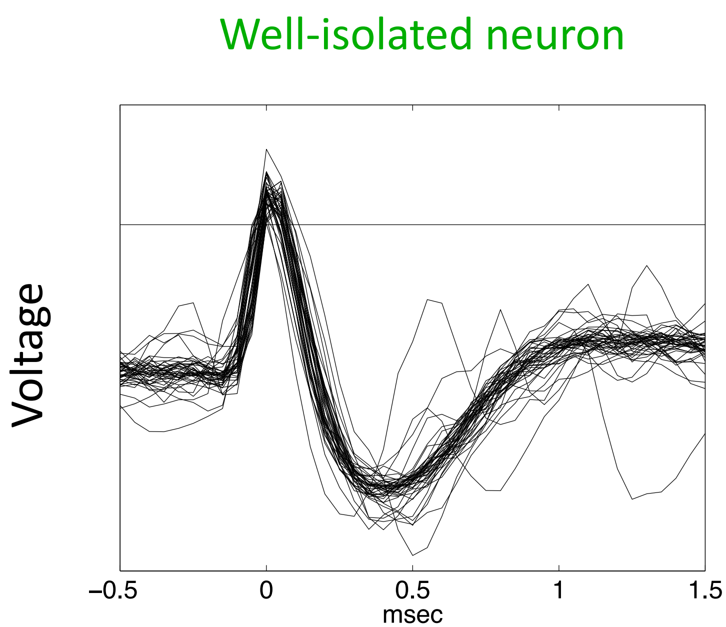
- A simple spike-sorting algorithm based on spike amplitude.

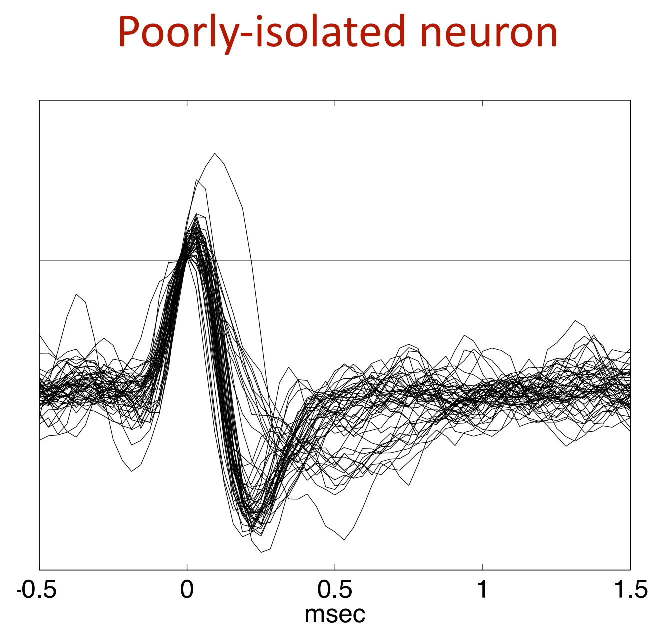


Threshold detection



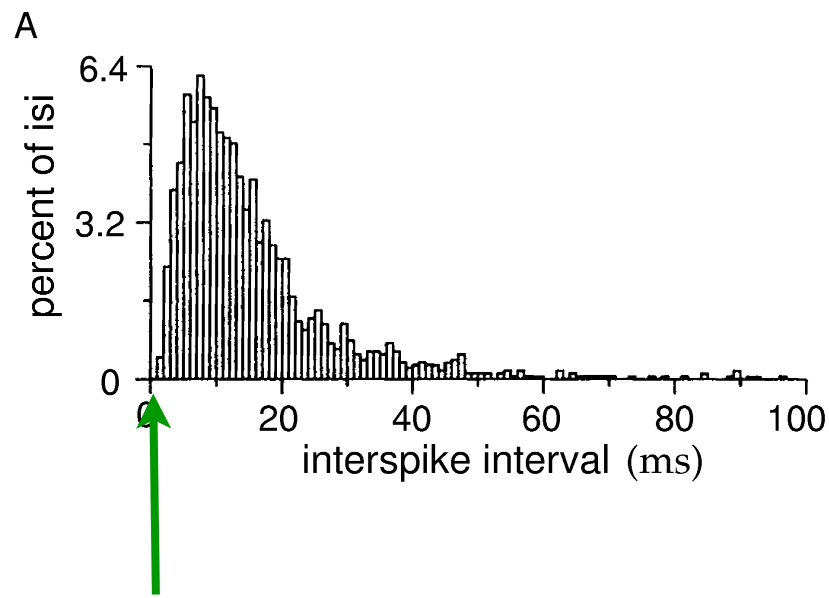
Sort quality: spike waveforms





There is no choice of threshold that can fix this.

Sort quality: ISI histogram



A well-isolated neuron should have no ISIs less than the absolute refractory period (~1 ms)

Threshold detection

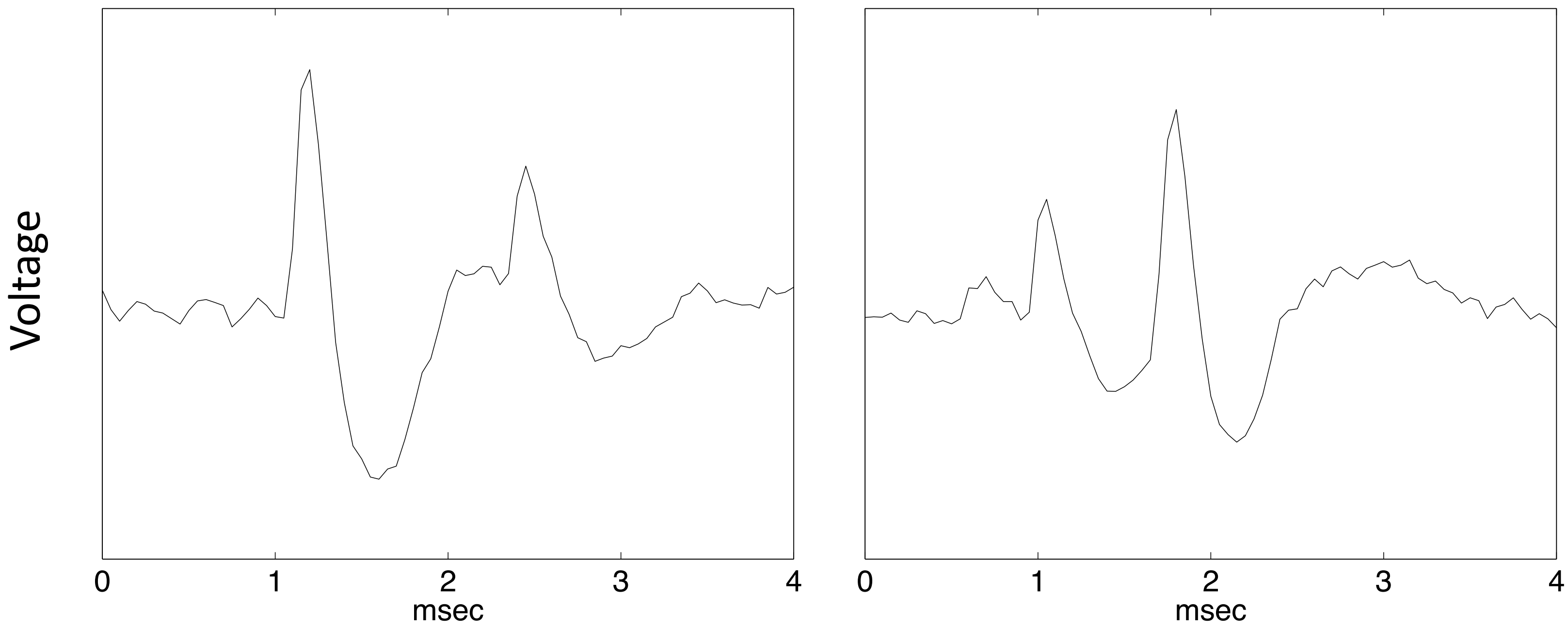
Pro:

- Simplicity

Cons:

- Sort quality often poor
- Bias toward neurons with large APs

Threshold detection prone to errors due to AP overlap



AP amplitude can change dramatically depending on location and size of adjacent APs.

False positives: two low-amplitude spikes combine to cross threshold.

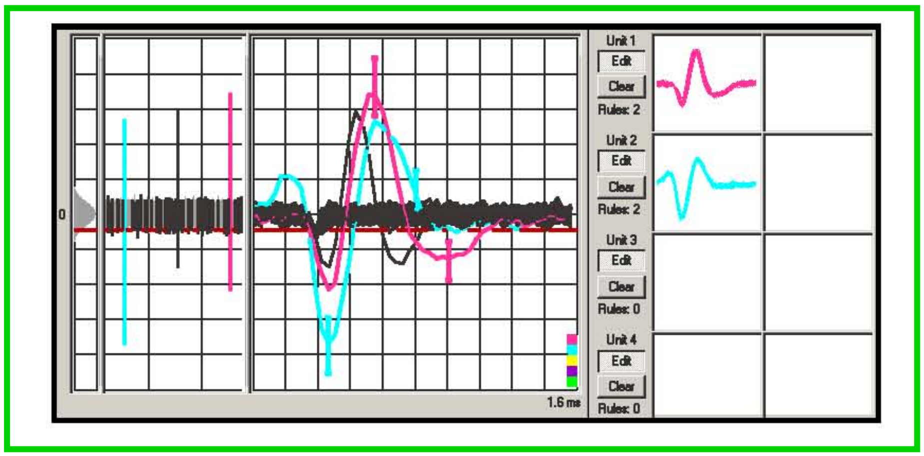
False negatives: peak of one AP lines up with trough of another AP, thus "canceling" each other.

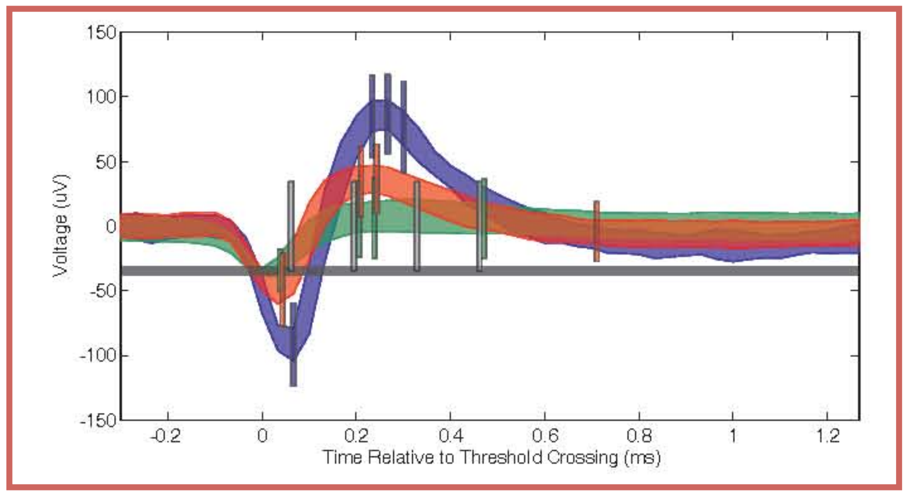
Spike sorting algorithms

1. Threshold detection
2. Hoop-based spike discrimination
3. Feature analysis, then clustering
4. Principal components analysis, then clustering

Hoop-based spike discrimination

- Threshold-based approach too simple → can't differentiate similar waveforms, which are quite common given stereotyped APs.
- Using a set of voltage-time "hoops" provides considerably more differentiation.
- Hoops can be set either manually (experimenter) or automatically (unsupervised learning algorithm).





Santhanam et al. (2004) IEEE EMBS.

Spike sorting algorithms

1. Threshold detection

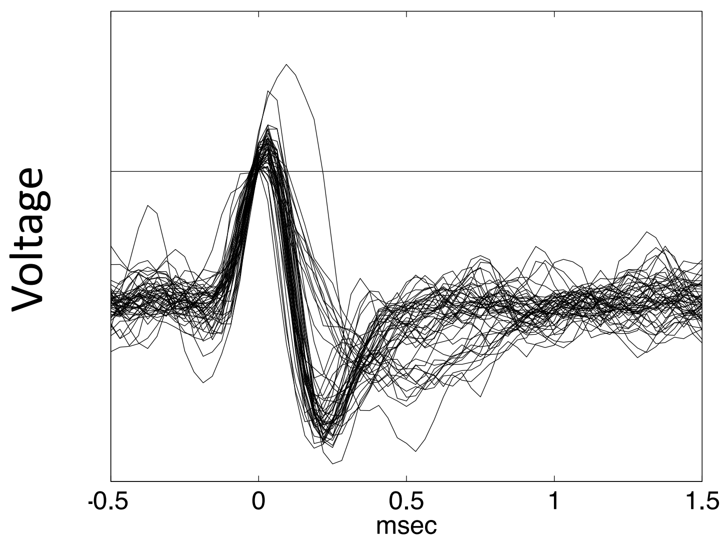
2. Hoop-based spike discrimination

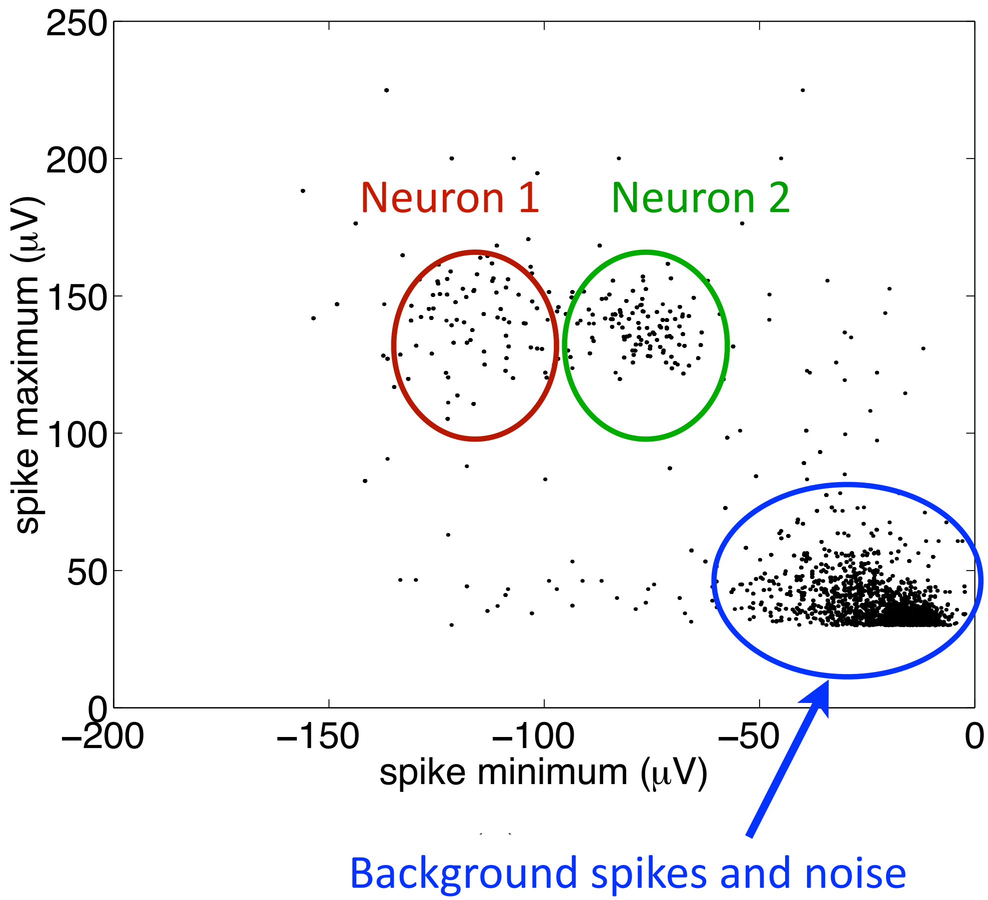
3. Feature analysis, then clustering

4. Principal components analysis, then clustering

Feature analysis

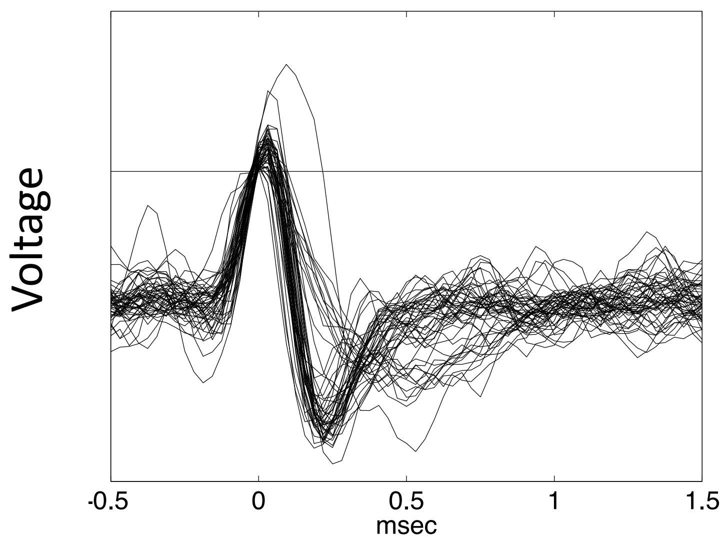
Each data point corresponds to an *entire* waveform snippet.

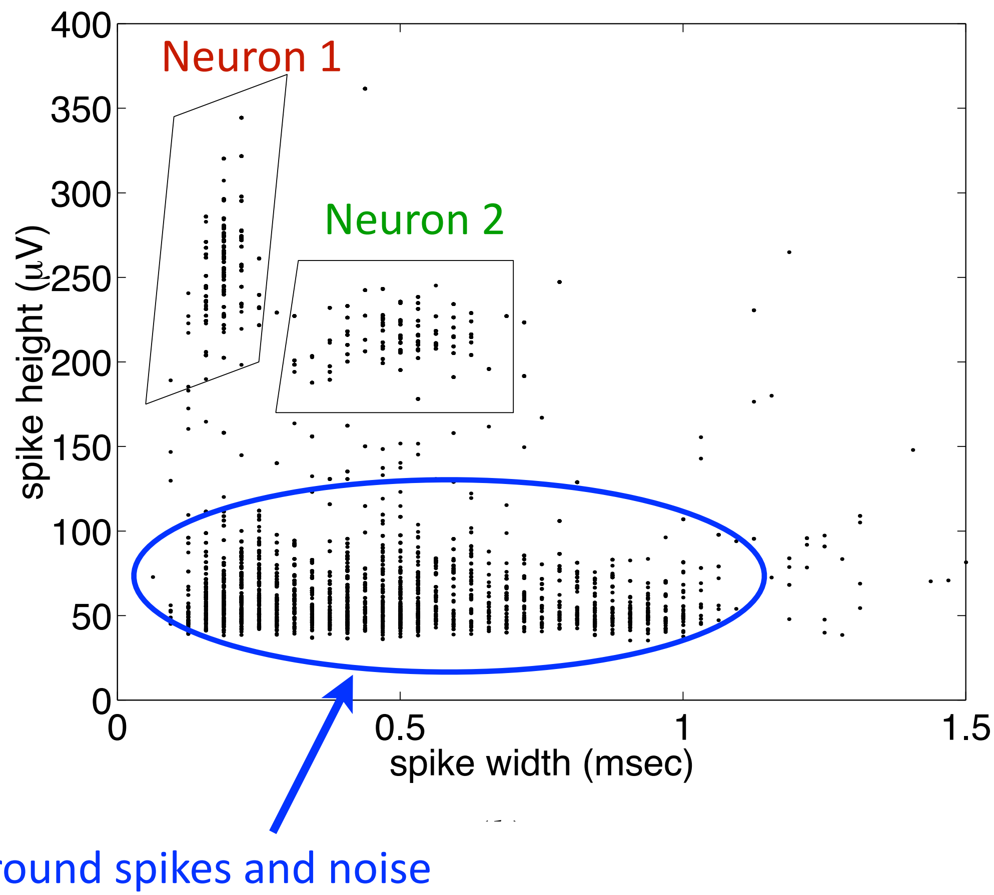




Appears to be 2 neurons with similar spike maximum, but different spike minimum.

Feature analysis





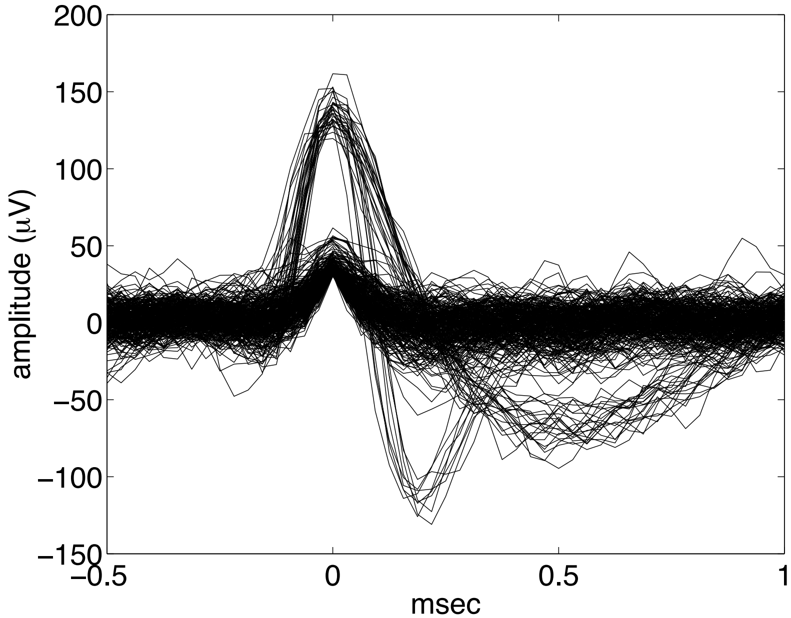
Can choose other features of spike waveform shape.

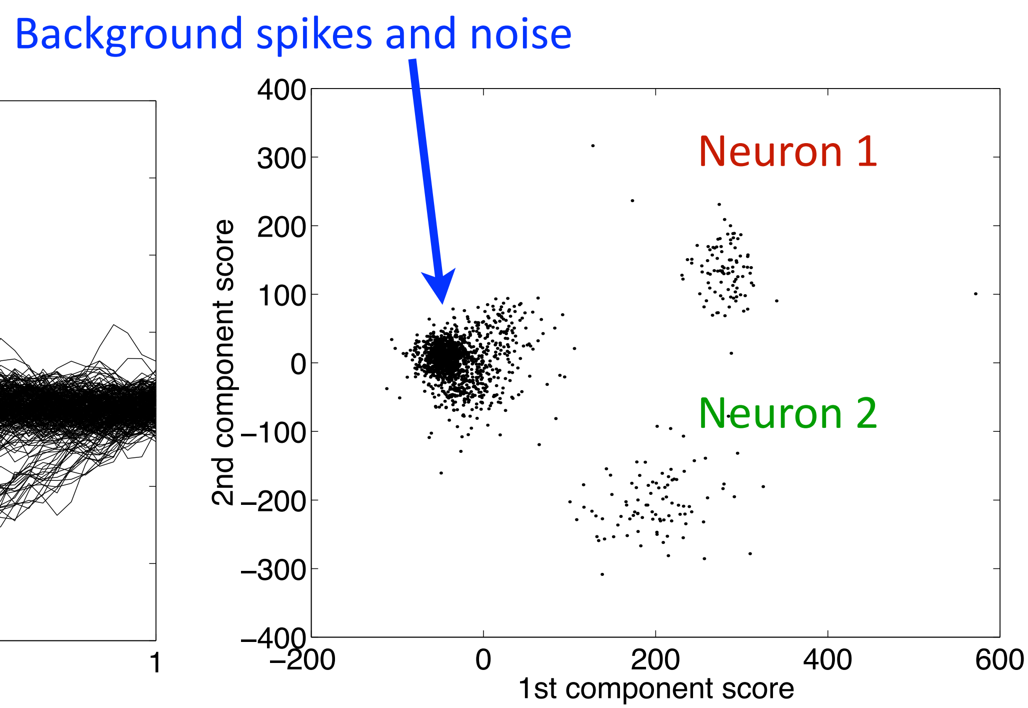
Spike sorting algorithms

1. Threshold detection
2. Hoop-based spike discrimination
3. Feature analysis, then clustering
4. Principal components analysis, then clustering

Principal components analysis (PCA)

- Choosing features is ad-hoc and can often yield poor cluster separation.
- PCA is an unsupervised technique that identifies directions of greatest data variance.
- We will talk about PCA in full detail in *PRML* Chapter 12.



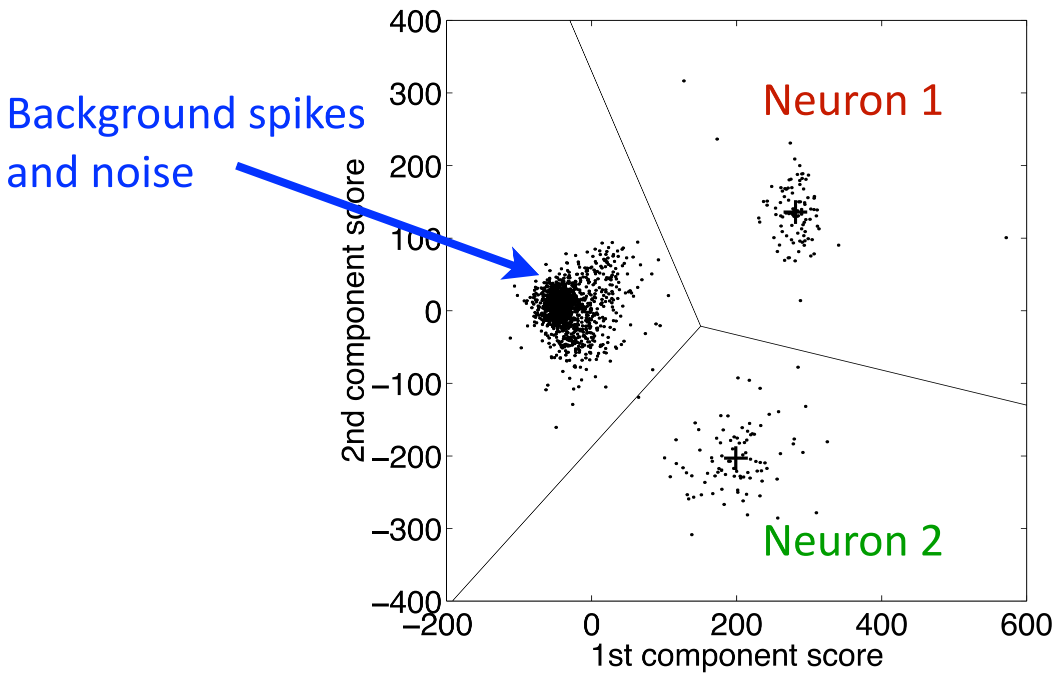


Clustering

How do we draw boundaries between different neurons in feature space or PCA space?

- Eyeballing (ad hoc)
- Nearest-neighbor (a.k.a., K-means algorithm)
- Mixture of Gaussians

PRML Chapter 9



What do these boundaries remind you of?

Classification vs. Clustering

They are very related.

Main difference:

During training,

- class labels are given in classification

- class labels are not given in clustering
(i.e., don't know how many clusters, nor which cluster each training data point belongs to)

Spike sorting algorithms

1. Threshold detection

2. Hoop-based spike discrimination

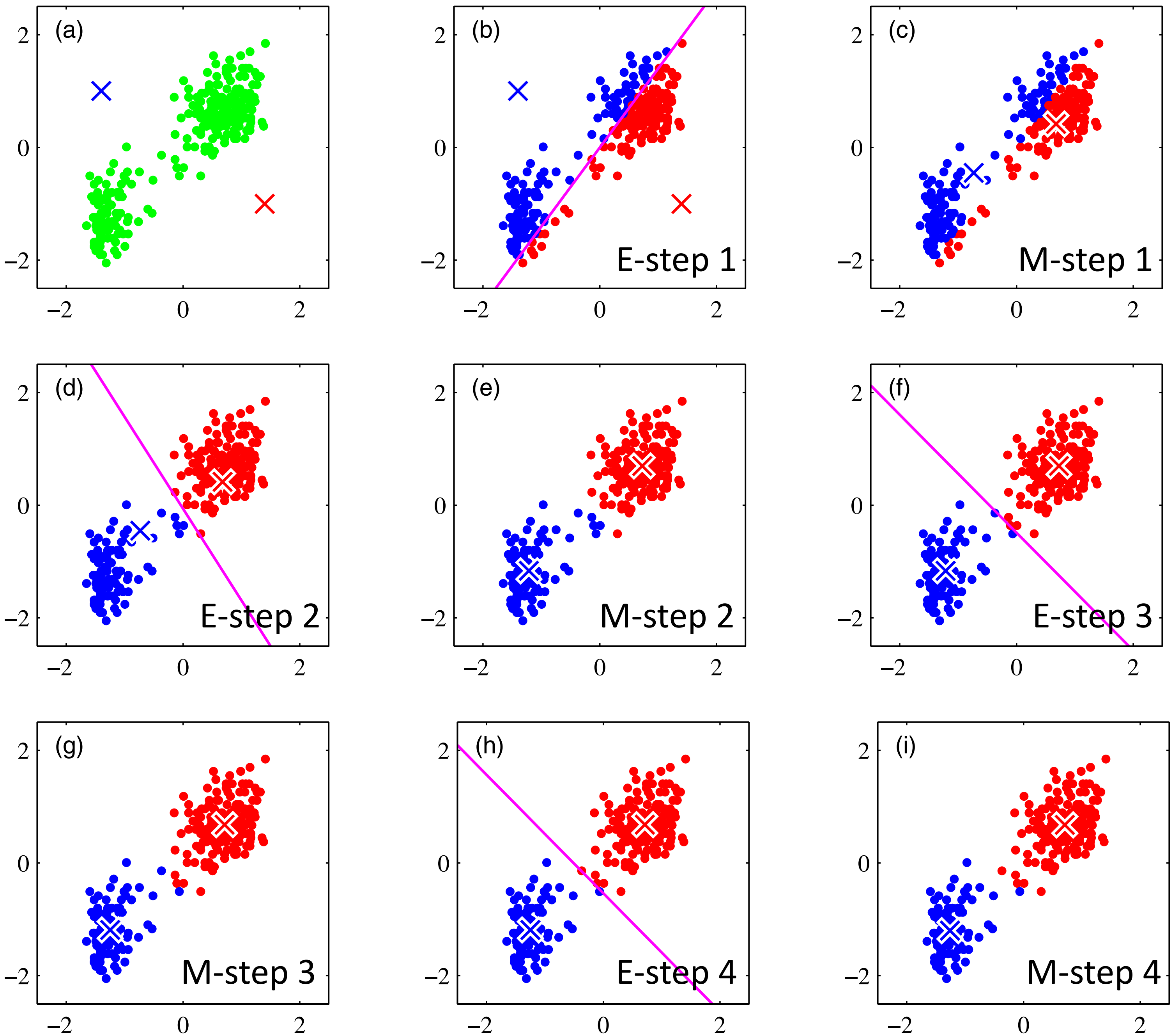
3. Feature analysis, then clustering *PRML* Chapter 9

4. Principal components analysis, then clustering

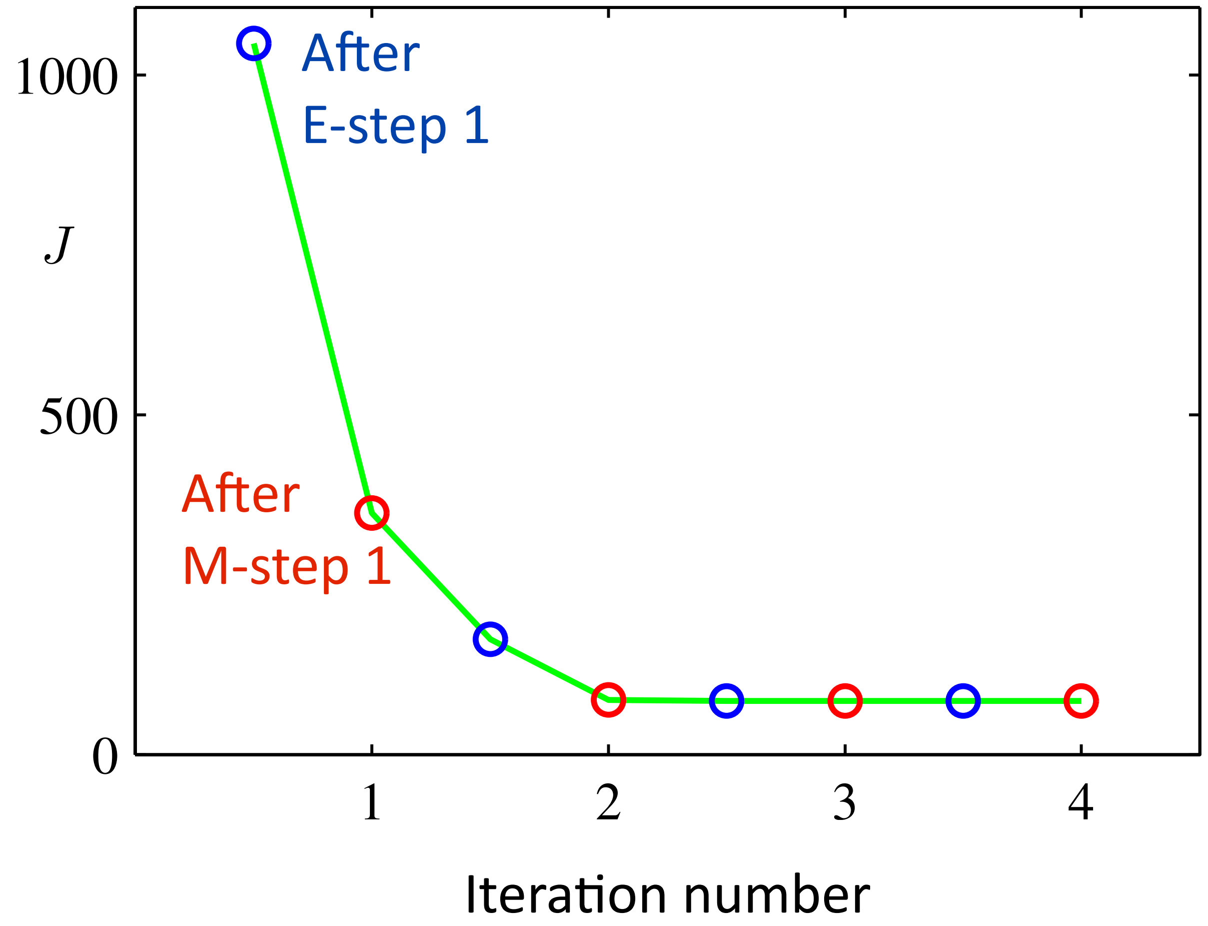
PRML Chapter 12

See “Mixture Models and EM” (Part 1) handout

K-means example

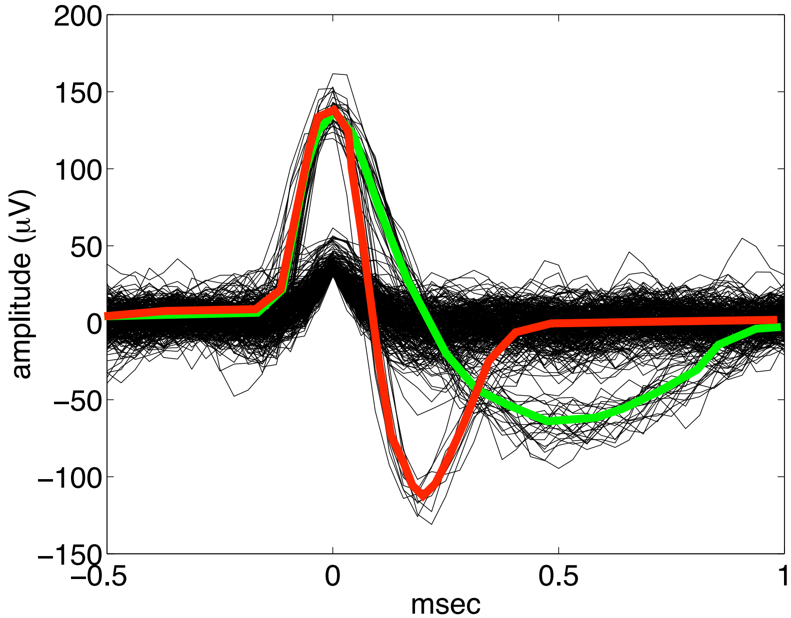


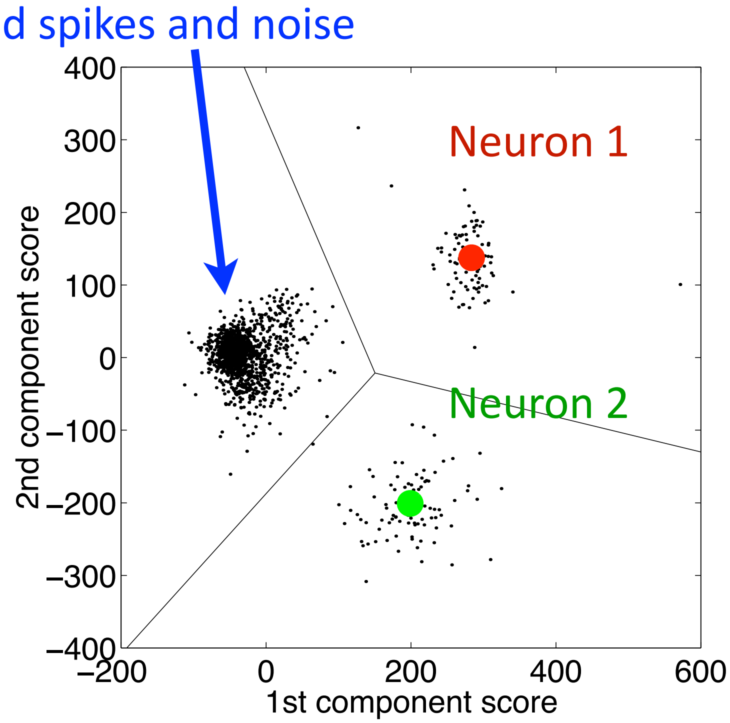
Cost function decreases during K-means



What do the cluster centers represent in spike sorting?

- Each data point corresponds to an *entire* waveform snippet
- Cluster center is an AP prototype for that neuron





Drawbacks of K-means

- Hard assignments to clusters. But what if a data point lies roughly midway between cluster centers?

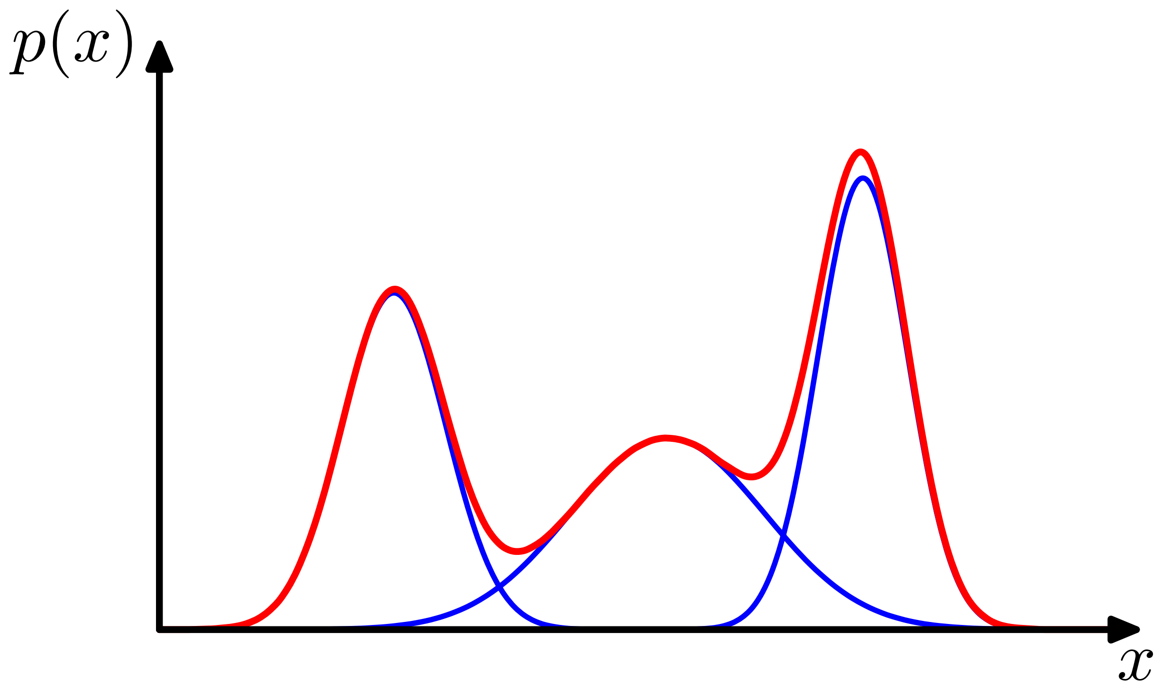
- No easy way to determine the number of clusters K.

This motivates the Gaussian mixture model (GMM):

- Soft assignments to clusters in a way that reflects level of uncertainty of cluster assignment.

- An optimal K can be found from the data. This is made possible by the fact that the GMM is a probabilistic model.

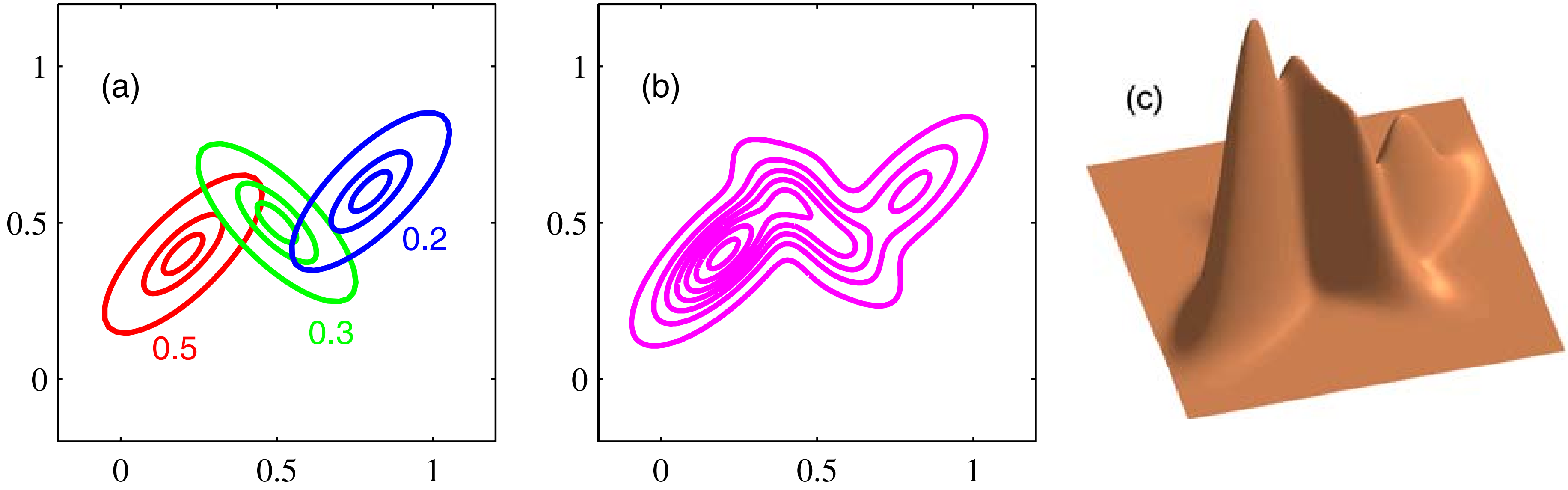
Mixture of 3 Gaussians in one dimension



$$p(\mathbf{x}) = \sum_{k=1}^{K} \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \Sigma_k)\, \pi_k$$

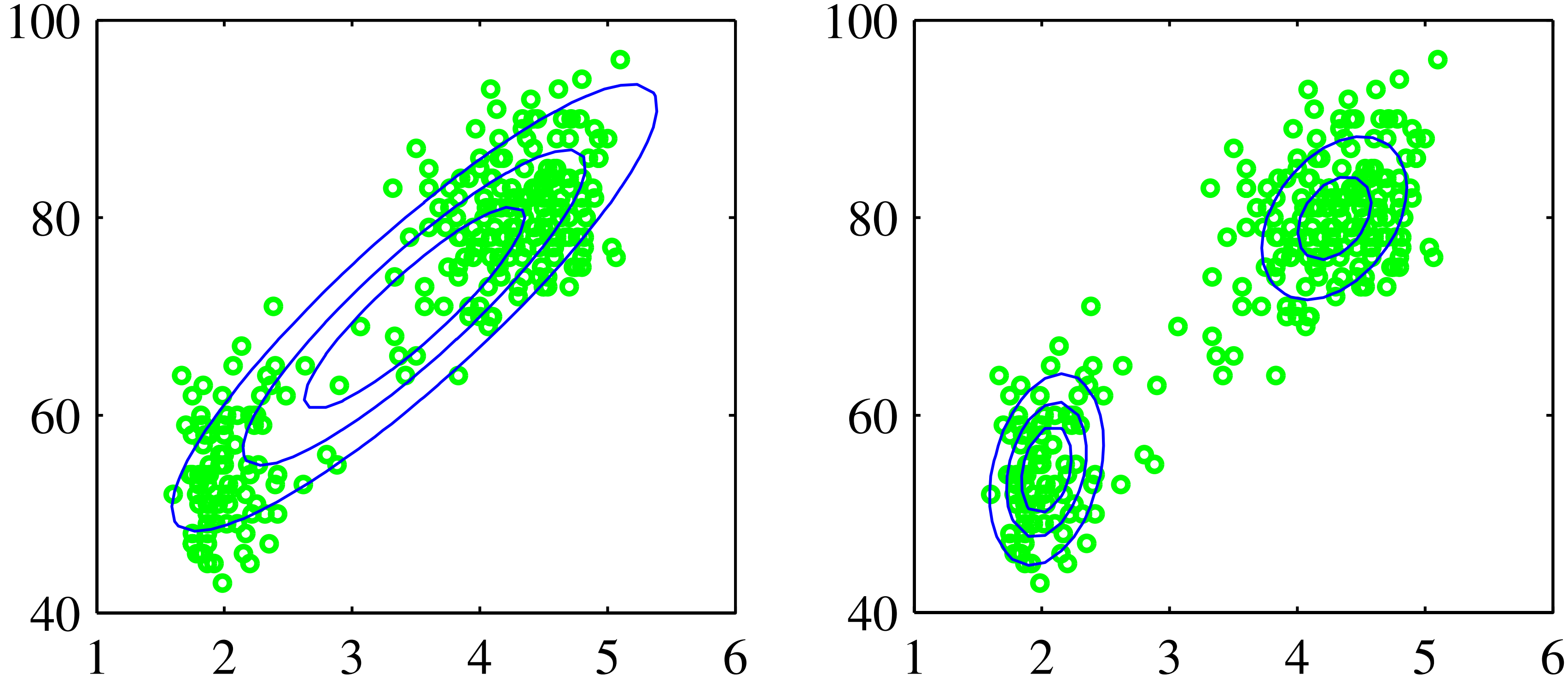
Weight of component *k*

Mixture of 3 Gaussians in two dimensions



Mixing Gaussian distributions is a powerful way of creating rather complex distributions.

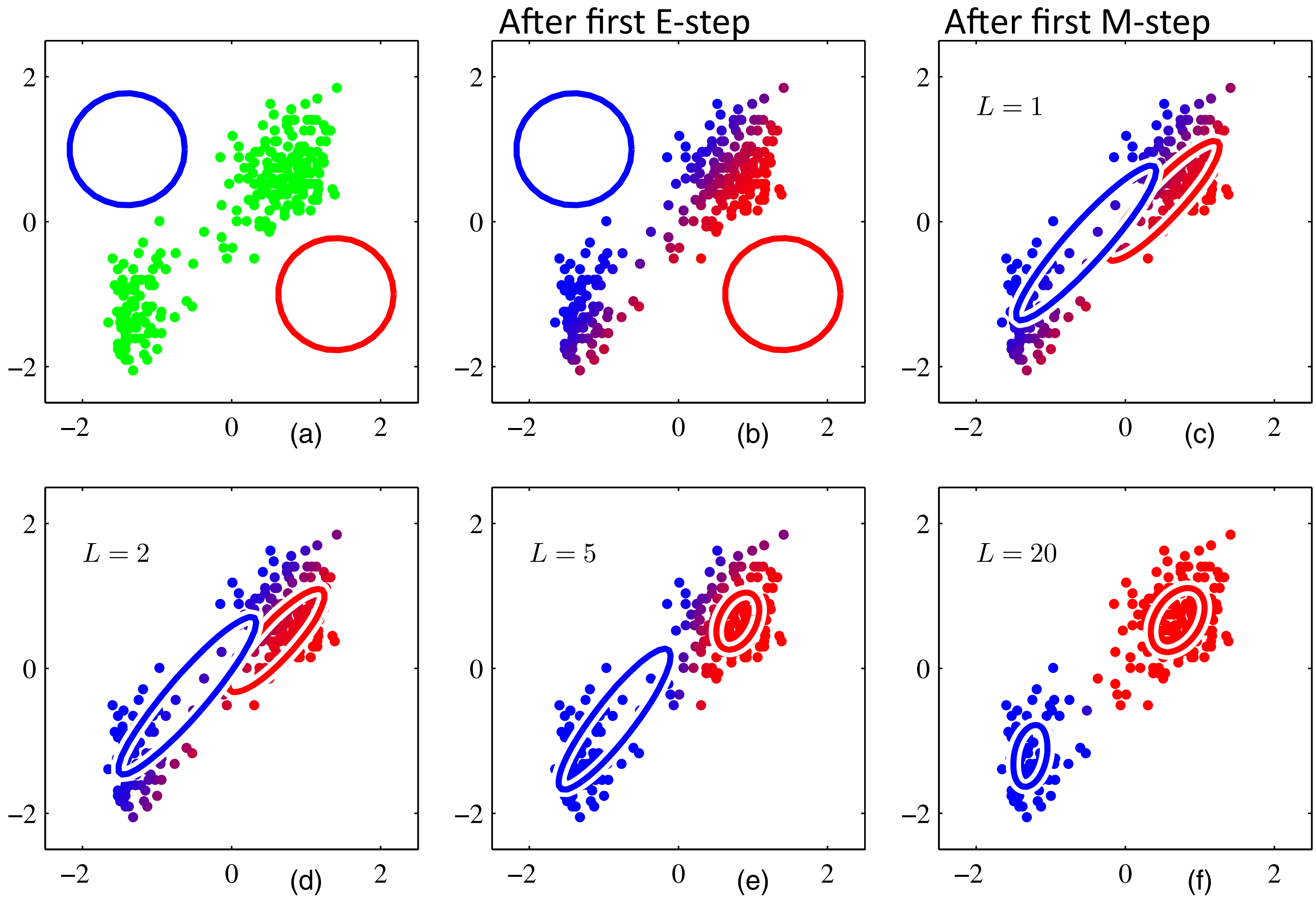
Mixture of Gaussians for clustering



- Goal: Assign each data point to a cluster
- As with K-means, we don't know a priori how many clusters, nor cluster shapes
- Approach: model each cluster with a Gaussian

See “Mixture Models and EM” (Part 2) handout

Example of EM for mixture of Gaussians



See “Mixture Models and EM” (Part 3) handout