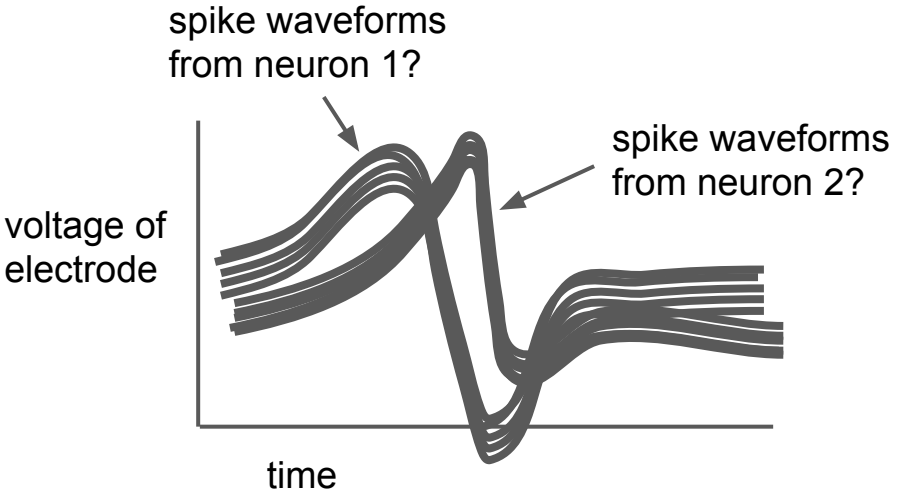
Last time:

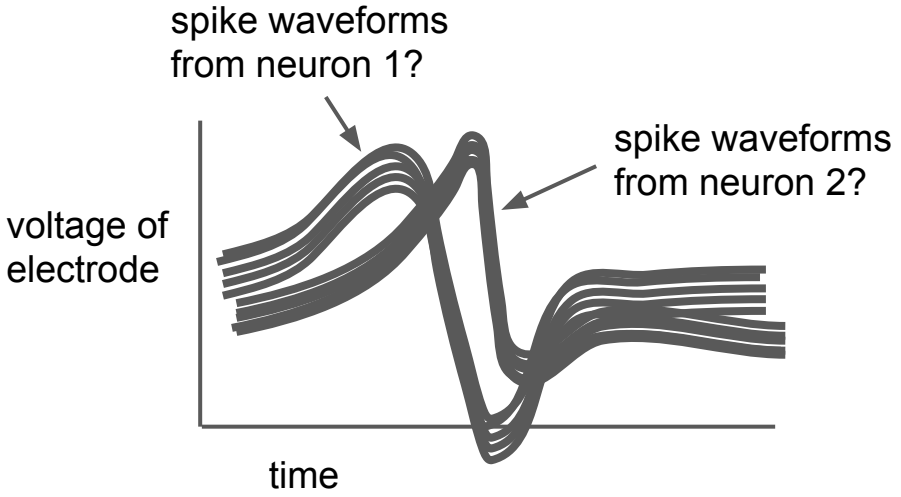
Last time:

- identifying temporal templates with PCA

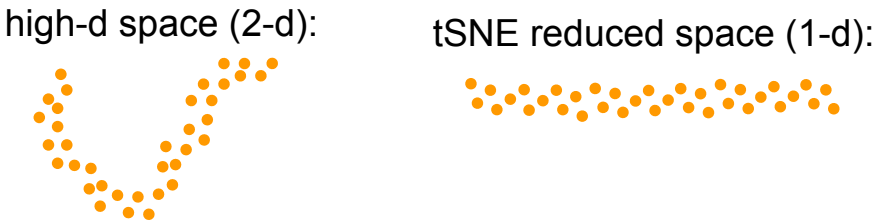


Last time:

- identifying temporal templates with PCA



- using tSNE for nonlinear dimensionality reduction



Today:

- finish up on-your-own script with tSNE, increasing the number of clusters

- Clustering: k-means algorithm

Finish Colab notebook from last time (Section 3): varying number of clusters and applying tSNE.

On your own. 15 minutes!

Clustering

Clustering characterizes the “modes” in the data.

Clustering characterizes the “modes” in the data.

- idea: identify “regions” of data points that are separated from other regions

Clustering characterizes the “modes” in the data.

- idea: identify “regions” of data points that are separated from other regions
- unsupervised

Clustering characterizes the “modes” in the data.

- idea: identify “regions” of data points that are separated from other regions
- unsupervised
- helpful for unlabeled data…
  - natural behavior
  - spike sorting
  - neural activity

Clustering characterizes the “modes” in the data.

- idea: identify “regions” of data points that are separated from other regions
- unsupervised
- helpful for unlabeled data…
  - natural behavior
  - spike sorting
  - neural activity
- main problem: what is a cluster?
  → many different definitions of cluster
  → many different algorithms

Clustering characterizes the “modes” in the data.

- idea: identify “regions” of data points that are separated from other regions
- unsupervised
- helpful for unlabeled data…
  - natural behavior
  - spike sorting
  - neural activity
- main problem: what is a cluster?
  → many different definitions of cluster
  → many different algorithms
- second problem: how many clusters?

k-means algorithm

k-means algorithm

- groups *n* data points into *k* clusters

k-means algorithm

- groups *n* data points into *k* clusters
  - each cluster has a mean ← parameters

k-means algorithm

- groups *n* data points into *k* clusters
  - each cluster has a mean ← parameters
  - each data point belongs to cluster with nearest mean

k-means algorithm

- groups *n* data points into *k* clusters
  - each cluster has a mean ← parameters
  - each data point belongs to cluster with nearest mean

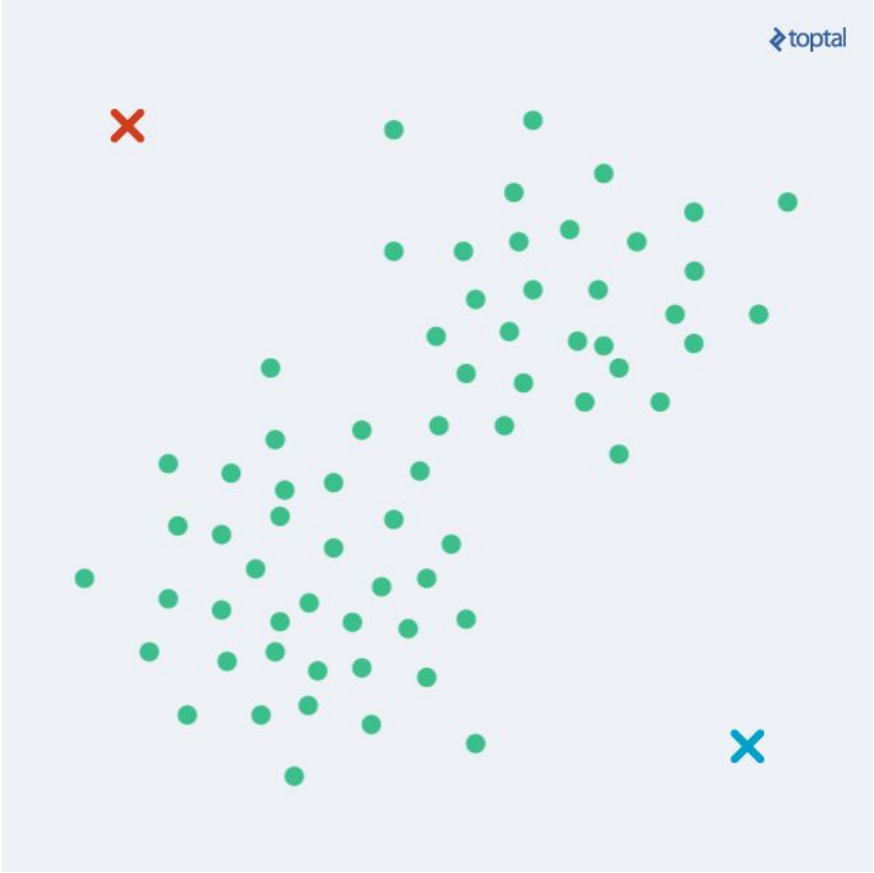
- the *k* means are identified via an alternating procedure:
  - start with initial means as random

k-means algorithm

- groups *n* data points into *k* clusters
  - each cluster has a mean ← parameters
  - each data point belongs to cluster with nearest mean

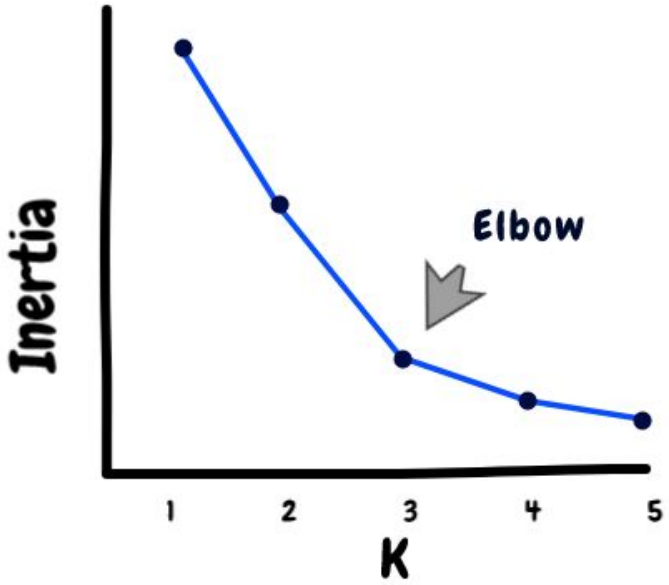
- the *k* means are identified via an alternating procedure:
  - start with initial means as random
  - repeat:
    - step 1: assign data points to cluster
    - step 2: re-compute means based on cluster's updated members

k-means algorithm



How to choose the number of clusters, *k*?

- plot “intertia” --- sum of within-cluster distances between the cluster mean and cluster’s members



Note: Always visualize the chosen clusters with PCA/tSNE!

k means algorithm in Python

Work on Colab notebooks together.