CLUSTERING

- Unsupervised learning problem
- Unlabelled data $\mathcal{D} = \{\underline{x}_i\}_{i=1}^N$
- Goal: To divide $\mathcal{D}$ into clusters (e.g. to create labels)

K-means clustering

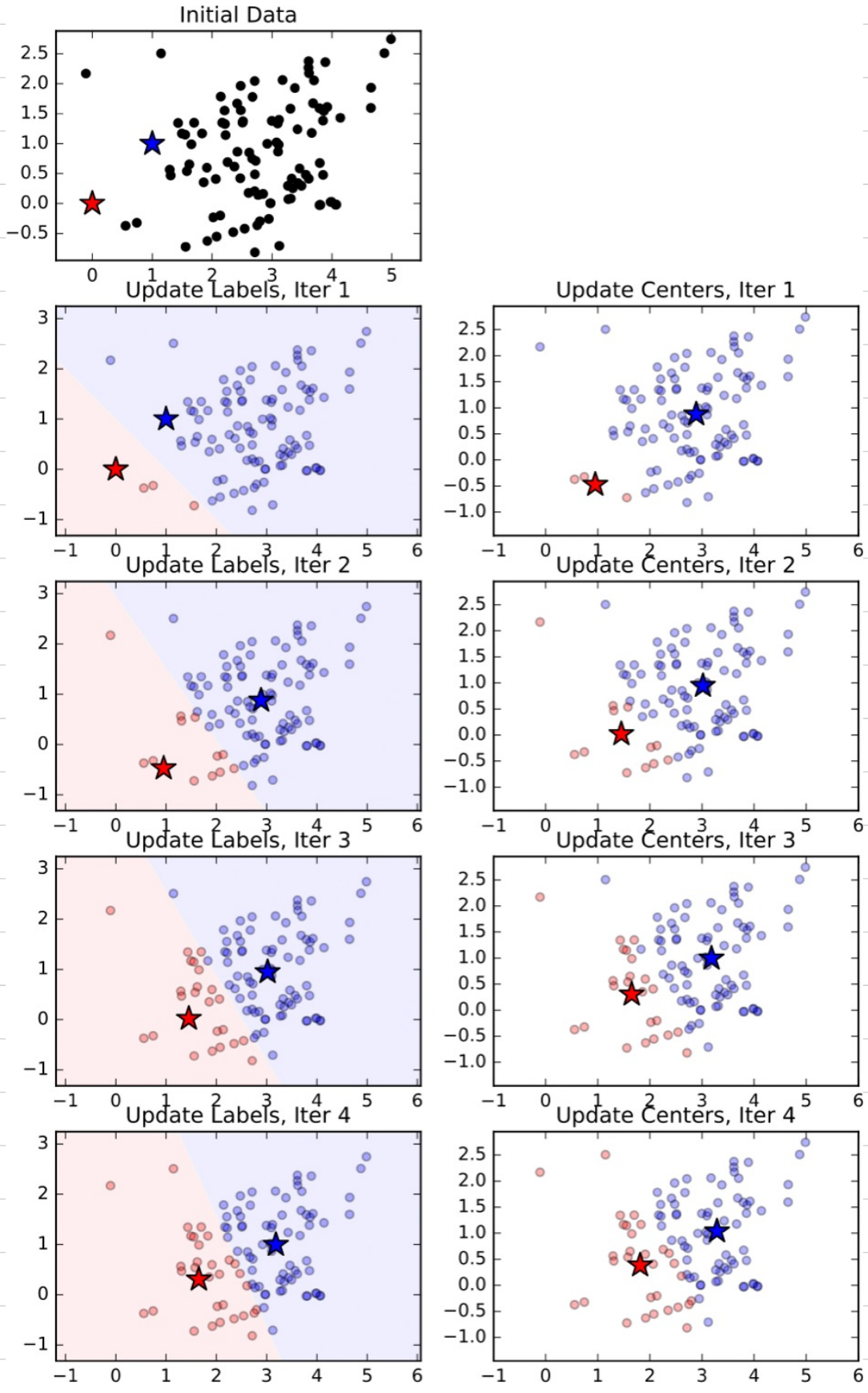
- Use Euclidean distance as metric
- Minimize

$$J_K(\{\underline{\mu}_\ell\}_{\ell=1}^K, \{z_i\}_{i=1}^N) = \sum_{i=1}^N \|\underline{x}_i - \underline{\mu}_{z_i}\|^2$$

  - $\underline{\mu}_\ell$ is "mean" of cluster ℓ, $\ell = 1, \ldots, K$
  - z_i is index of cluster given to datapoint $\underline{x}_i$
    z_i takes values in $\{1, 2, \ldots, K\}$
  - Exact minimization of J_K difficult
- Iterative algorithm to approximate minimization
  - Initialize $\{\underline{\mu}_\ell\}_{\ell=1}^K$ to K random points
  - Assign $\underline{x}_i$ to cluster index with closest mean
    i.e. $z_i = \arg\min_{\ell=1,\ldots,K} \|\underline{x}_i - \underline{\mu}_\ell\|^2$
  - After all data assigned, recompute $\{\underline{\mu}_\ell\}_{\ell=1}^K$

$$\underline{\mu}_\ell = \frac{1}{N_\ell} \sum_{i: z_i = \ell} \underline{x}_i, \quad \ell = 1, 2, \ldots, K$$

  - Repeat until convergence ($\underline{\mu}_\ell$'s don't change)


Initial Data
Update Labels, Iter 1
Update Centers, Iter 1
Update Labels, Iter 2
Update Centers, Iter 2
Update Labels, Iter 3
Update Centers, Iter 3
Update Labels, Iter 4
Update Centers, Iter 4

K-Medoids Algorithm

- Replace Euclidean distance $\|\underline{x}_i - \underline{\mu}_\ell\|$ with more general dissimilarity measure $d(\underline{x}_i, \underline{\mu}_\ell)$
- Replace "mean" with "centroid" (or "medoid")

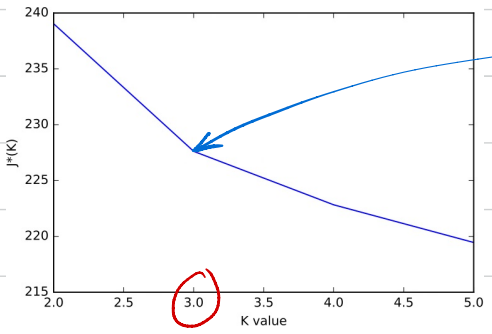
How to pick K?

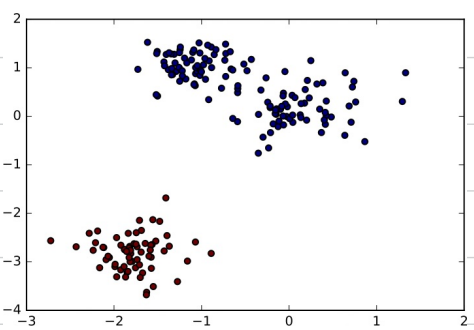
- Sometimes K is known beforehand, e.g., for handwritten digit recognition $K = 10$
- More generally, need to estimate K from data

Use K-means objective $J^*(K)$

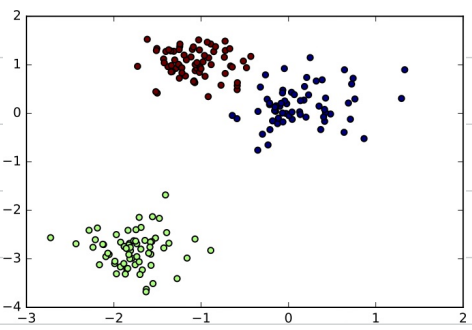
$$J^*(K) = \min_{\{z_i\}_{i=1}^N, \{\underline{\mu}_\ell\}_{\ell=1}^K} \sum_{i=1}^N \|\underline{x}_i - \underline{\mu}_{z_i}\|^2$$

- Estimate $J^*(K)$ by running K-means algorithm with many starting points and finding the smallest resultant metric
- Plot $J^*(K)$ v. K and look for "knee"

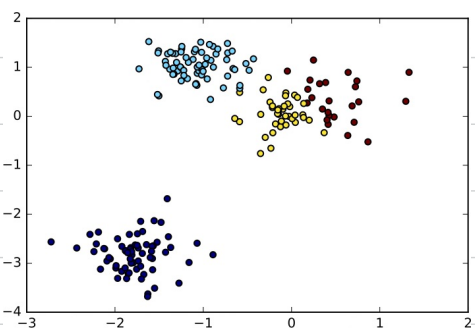




K = 2



K = 3



K = 4

Applications of Clustering

Obvious application: creating labelled data from unlabelled data for classification

Vector Quantization (VQ)

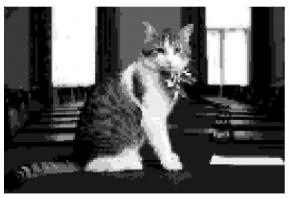
- Lossy compression of images
- chop image up into equal size blocks
  e.g. 5 pixels × 5 pixels = 25-dim vector $\underline{x}_i$
- Do K-means clustering of $\underline{x}_i$'s for some K
- Encode image by storing the K means $\{\underline{\mu}_\ell\}_{\ell=1}^K$ and cluster indices $\{z_i\}_{i=1}^N$ for the $\underline{x}_i$'s
- To decompress image use the mean $\underline{\mu}_{z_i}$ in place of $\underline{x}_i$

0.04 bits/pixel →

K = 2

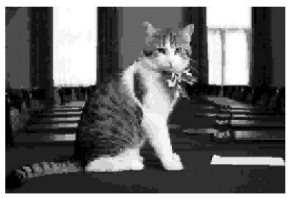
K = 5

← 0.09 bits/pixel



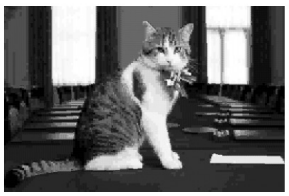
0.13 bits/pixel →

K = 10



K = 20

← 0.17 bits/pixel



0.23 bits/pixel →

K = 50

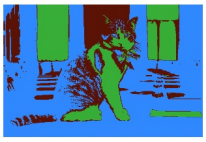
Original

← 8 bits/pixel

Image Segmentation

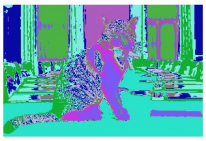
- Similar to VQ
- $\underline{x}_i = [R\ G\ B]$ Values for each pixel i
- Cluster $\underline{x}_i$'s using k-means, and replace pixel with mean $\underline{\mu}_{z_i}$

K = 2

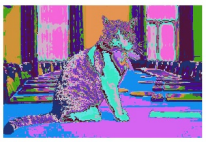


K = 3

K = 5



K = 10

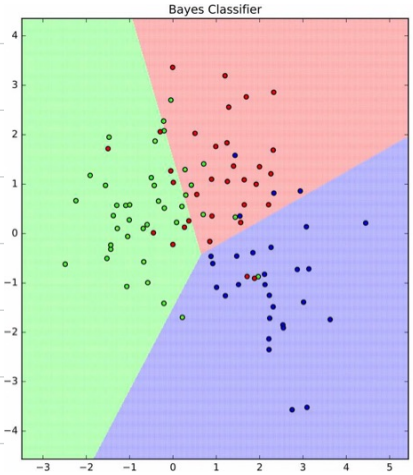


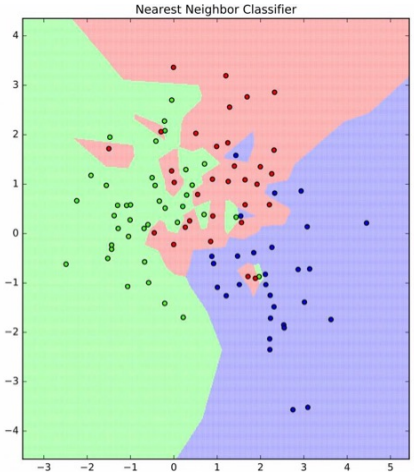
K = 20

original

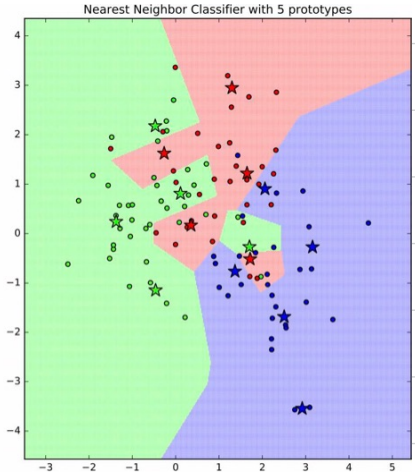
K-Means for NN Classification

- NN requires computation of N distances for each new $\underline{x}$ that we want to classify using $\mathcal{T} = \{(\underline{x}_i, y_i)\}_{i=1}^{N}$
- For each label ℓ, use K-means to cluster the training data for label ℓ ($K \ll N$)
- Replace training data for each class with the K means $\underline{\mu}_1, \ldots, \underline{\mu}_K$ for that class.
- Now we need to only compute KM distances for each new $\underline{x}$, which may be $\ll N$ if N is large





N = 100, M = 3



K = 5