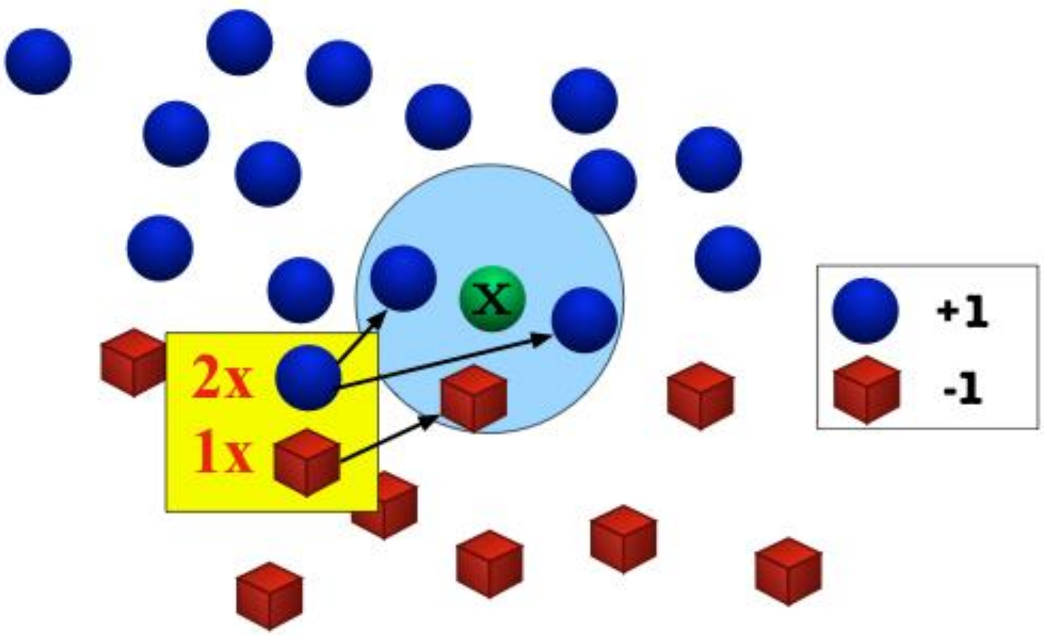
CSE 176 Introduction to Machine Learning

Lecture 5: K-means and K-modes Clustering

Some materials from Yuri Boykov

Recap: K nearest neighbor algorithm

- ❑Nearest neighbor often instable (noise)
- ❑For a test input ***x***, assign the most common label amongst its k most similar training inputs

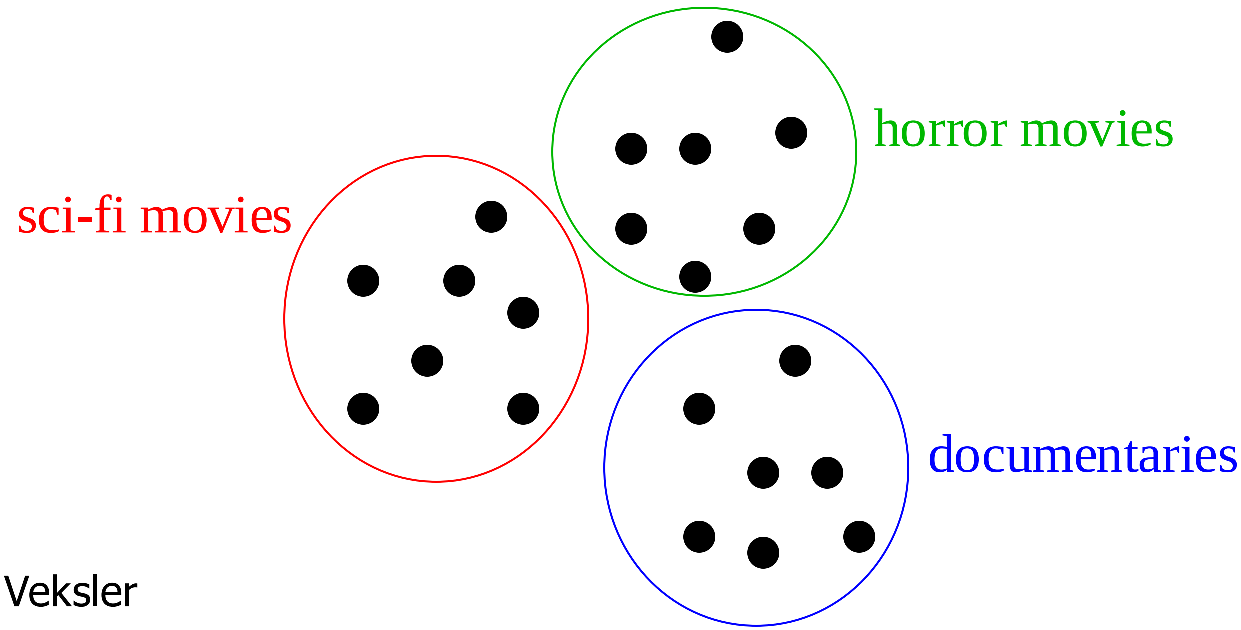


Recap: Choosing K

- How should we choose K?
  - Select K with highest test accuracy
- Split data into training, validation and test sets
  - Training set: compute nearest neighbour
  - Validation set: optimize hyperparameters such as K
  - Test set: measure performance

General Grouping or Clustering

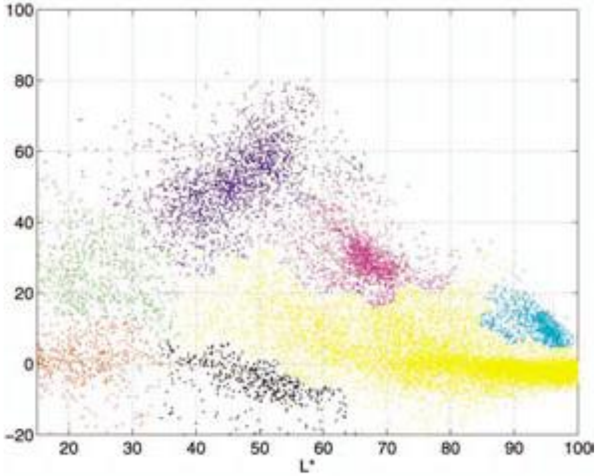
- Have data points (samples, a.k.a. feature vectors, examples, etc.) $\boldsymbol{f}_1,\ldots,\boldsymbol{f}_p,\ldots$
- Cluster similar points into groups
    - points are **not** pre-labeled
    - think of clustering as 'discovering' labels



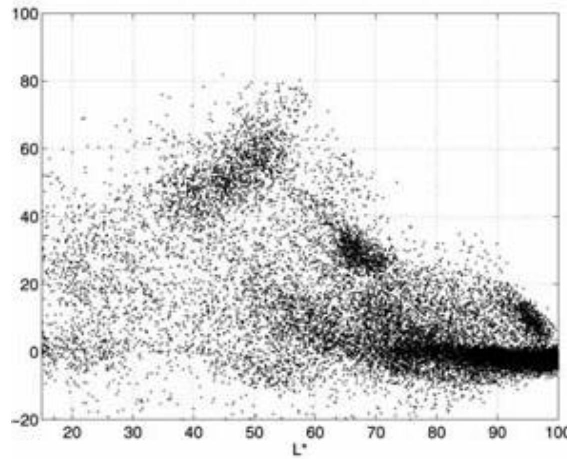
slides from Olga Veksler

Data Clustering

decision boundaries for ND features
could be arbitrarily complex (surfaces)



?



Example: break data points (e.g. RGB or RGBXY space) into a few clusters

Clustering methods

- ❑K-means
- ❑Distortion clustering
- ❑Probabilistic clustering, EM, GMM
- ❑Parametric vs non-parametric formulations
- ❑Kernel and spectral methods
- ❑Graph clustering
- ❑Mean-shift

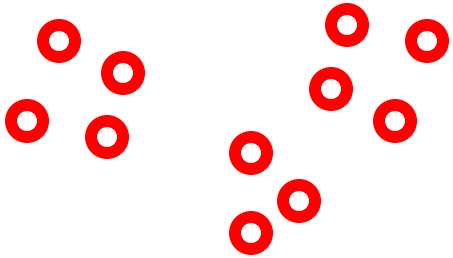
Topics today

- K-means clustering
- K-modes clustering

K-means Algorithm (Lloyd’s, 1957)

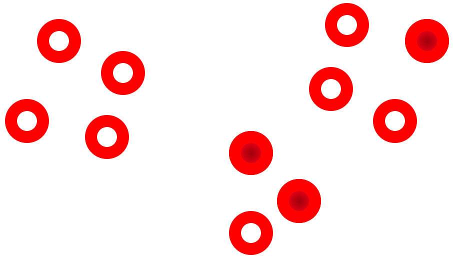
K-means Clustering: Algorithm

- Initialization step
  1. pick ***K*** cluster centers randomly (e.g. from data points)



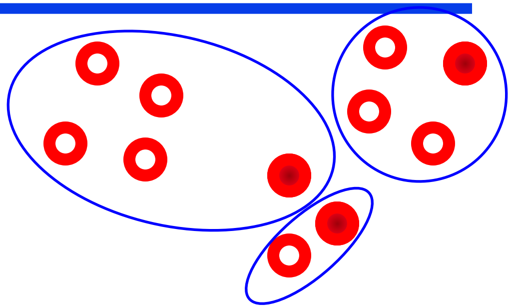
K-means Clustering: Algorithm

- Initialization step
  1. pick ***K*** cluster centers randomly (e.g. from data points)



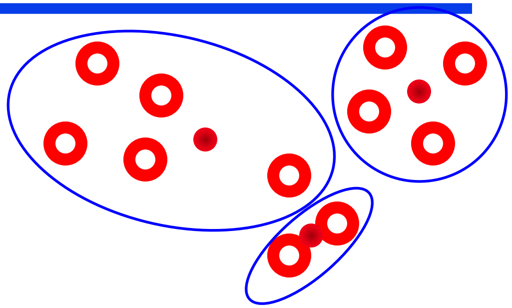
K-means Clustering: Algorithm

- Initialization step
  1. pick ***K*** cluster centers randomly
  2. assign each sample to its closest center



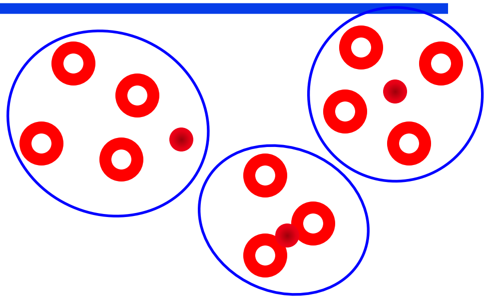
K-means Clustering: Algorithm

- Initialization step
  1. pick ***K*** cluster centers randomly
  2. assign each sample to its closest center



- Iteration steps
  1. compute centers as cluster means $\mu_k = \frac{1}{|S^k|}\sum_{p \in S^k} f_p$

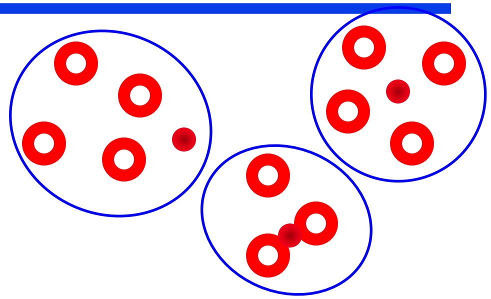
K-means Clustering: Algorithm



- Initialization step
  1. pick ***K*** cluster centers randomly
  2. assign each sample to its closest center
- Iteration steps
  1. compute centers as cluster means $\mu_k = \frac{1}{|S^k|} \sum_{p \in S^k} f_p$
  2. re-assign each sample to the closest mean

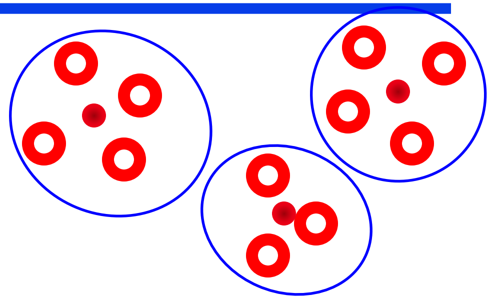
K-means Clustering: Algorithm

- Initialization step
  1. pick ***K*** cluster centers randomly
  2. assign each sample to its closest center



- Iteration steps
  1. compute centers as cluster means $\mu_k = \frac{1}{|S^k|} \sum_{p \in S^k} f_p$
  2. re-assign each sample to the closest mean
- Iterate until clusters stop changing

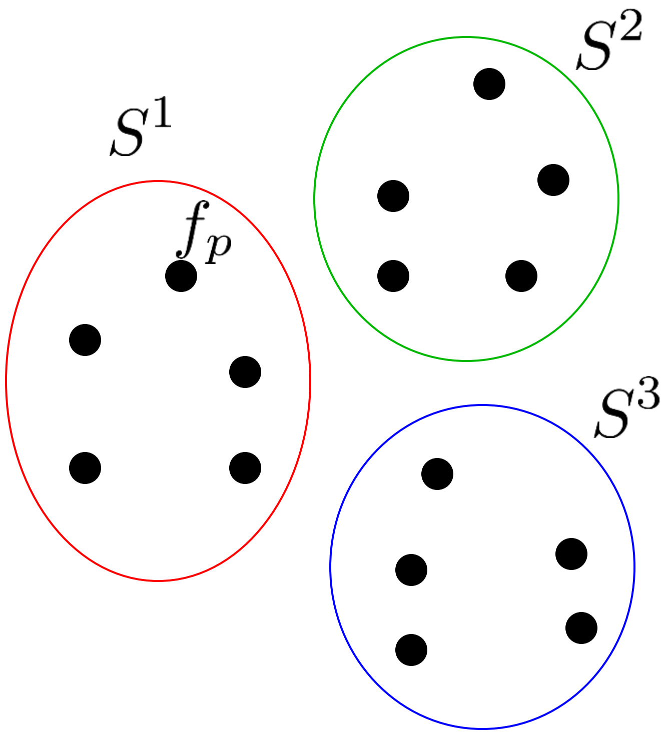
K-means Clustering: Algorithm



- Initialization step
  1. pick ***K*** cluster centers randomly
  2. assign each sample to its closest center
- Iteration steps
  1. compute centers as cluster means $\mu_k = \frac{1}{|S^k|} \sum_{p \in S^k} f_p$
  2. re-assign each sample to the closest mean
- Iterate until clusters stop changing

K-means Objective

K-means objective



features

$$\{f_p \,|\, p \in \Omega\}$$

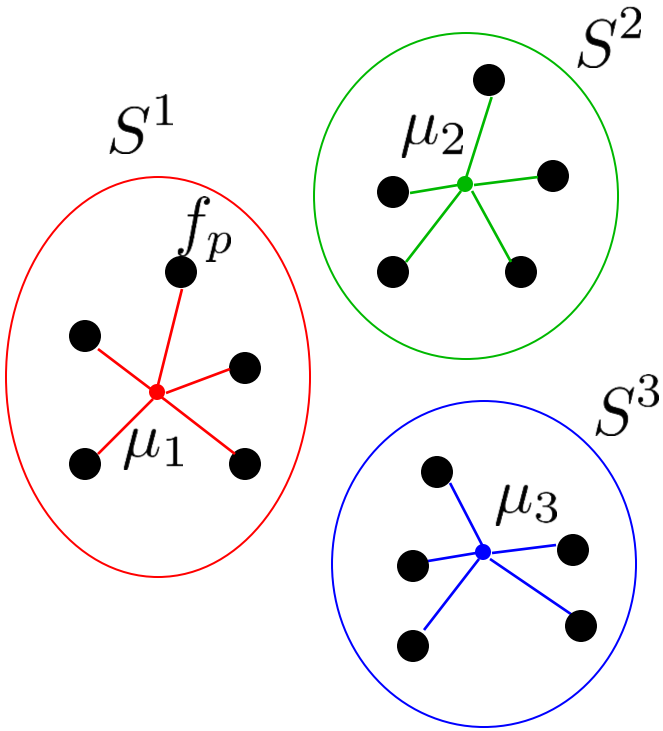
input

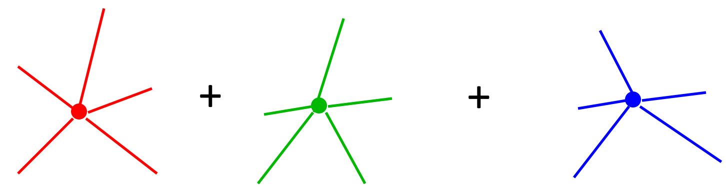
K subsets of Ω

$$S = \{S^1, \ldots, S^K\}$$

output

K-means objective



$E(S,\mu) =$ 

$$= \sum_{k=1}^{K} \sum_{p \in S^k} \|f_p - \mu_k\|^2$$

(SSD)

μ_k : extra parameters (means)

Squared distance as log-likelihood

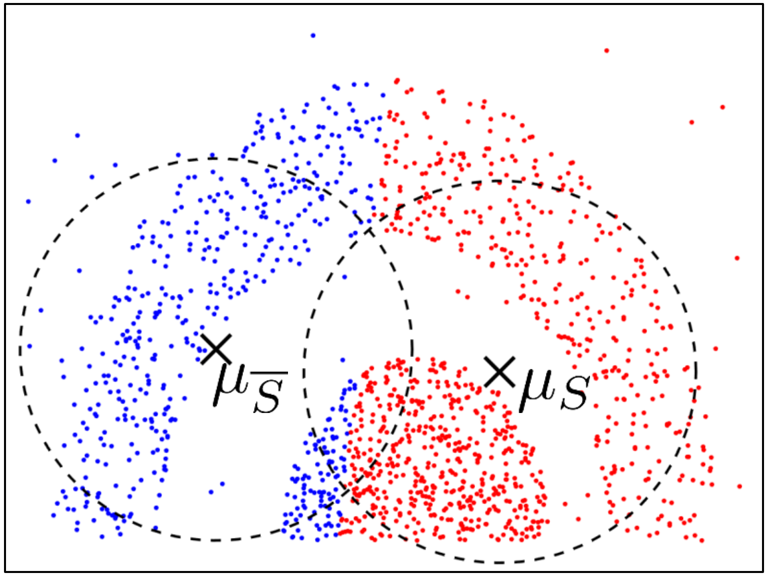
Assume K=2, $\Omega = S \cup \bar{S}$

$$\boxed{\sum_{p\in S} \|f_p - \mu_S\|^2 + \sum_{p\in \overline{S}} \|f_p - \mu_{\overline{S}}\|^2}$$

$$= \quad -\sum_{p\in \mathbf{S}} \ln \underline{\mathcal{N}(f_p|\mu_{\mathbf{S}})} \; - \sum_{p\in \bar{\mathbf{S}}} \ln \mathcal{N}(f_p|\mu_{\bar{\mathbf{S}}})$$

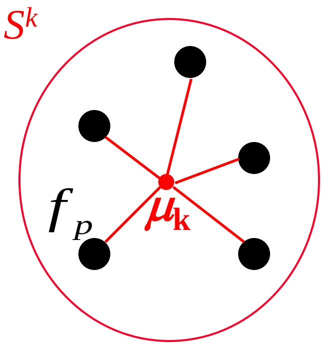
single Gaussian

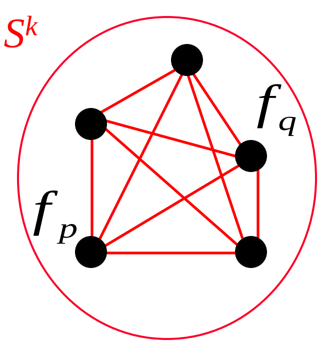
single Gaussian of ***fixed*** covariance



$$\theta_S = \{\mu_S\}$$

K-means as variance clustering criteria





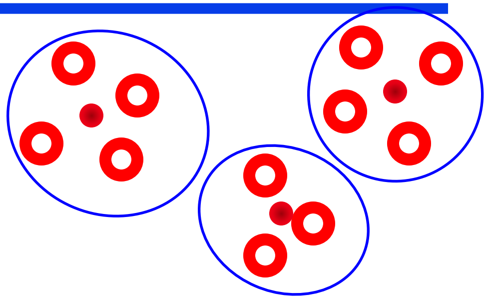
both formulas can be written as

$$E(S) \quad = \quad \sum_{k=1}^{K} |S^k| \; \mathbf{var}(S^k)$$

sample variance: $\mathrm{var}(S^k) = \frac{1}{|S^k|} \sum_{p \in S^k} \| f_p - \mu_k \|^2 = \frac{1}{2|S^k|^2} \sum_{pq \in S^k} \| f_p - f_q \|^2$

K-means Clustering: Algorithm

- Initialization step
  1. pick ***K*** cluster centers randomly
  2. assign each sample to its closest center



- Iteration steps
  1. compute centers as cluster means $\mu_k = \frac{1}{|S^k|} \sum_{p \in S^k} f_p$
  2. re-assign each sample to the closest mean
- Iterate until clusters stop changing

Lloyd's algorithm (1957)

- Each step decreases the value of the objective function

$$E(S,\mu) \;=\; \sum_{k=1}^{K} \sum_{p \in S^k} \|f_p - \mu_k\|^2$$

optimization variables

$$S = (S^1, \ldots, S^K)$$
$$\mu = (\mu_1, \ldots, \mu_K)$$

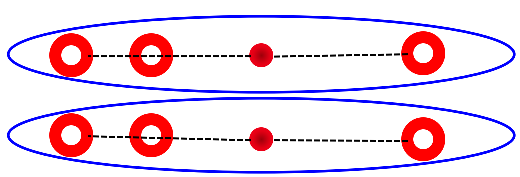
block-coordinate descent: step 1 optimizes $\{\mu_k\}$ for fixed $\{S_k\}$, step 2 optimizes $\{S_k\}$ for fixed $\{\mu_k\}$

K-means: Approximate Optimization

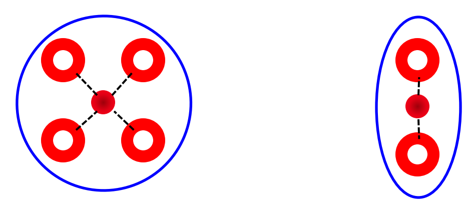
- K-means is fast and (sometimes) works well in practice
- But can get stuck in a local minimum of objective $\boldsymbol{E_K}$
  - not surprising, since the exact optimization of its objective is NP-hard

initialization

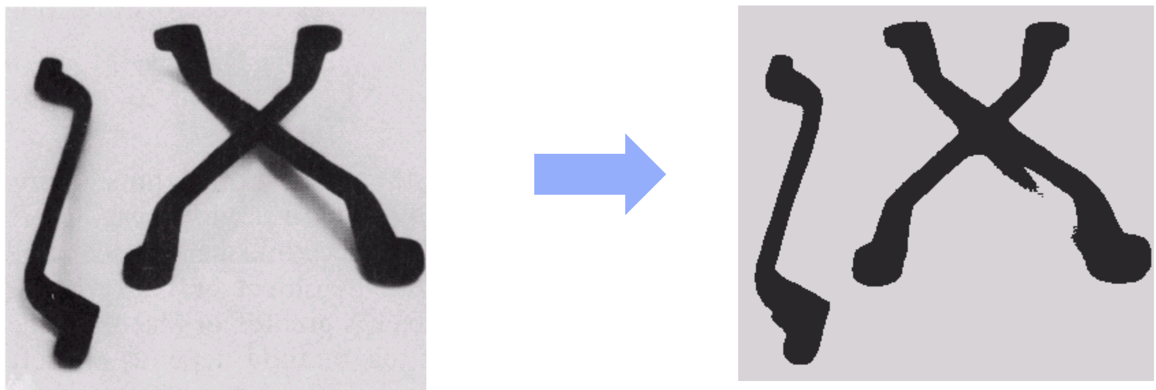
converged to local min



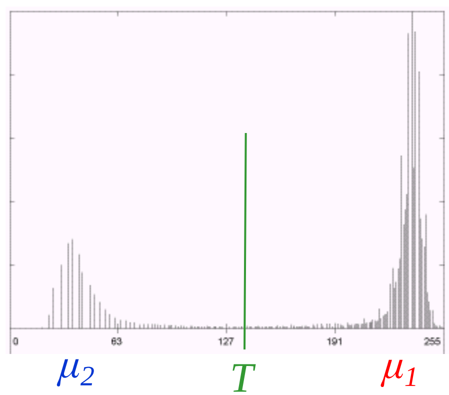
global minimum



K-means clustering examples: Segmentation



$$I_p \in R^1$$



here K-means finds compact clusters of pixels' intensities

In this case K-means (K=2) implicitly finds a good threshold (between 2 clusters)

K-means for colors (RGB features): Segmentation?

***k* = 3**

(**mean** color is used to show each segment/cluster)

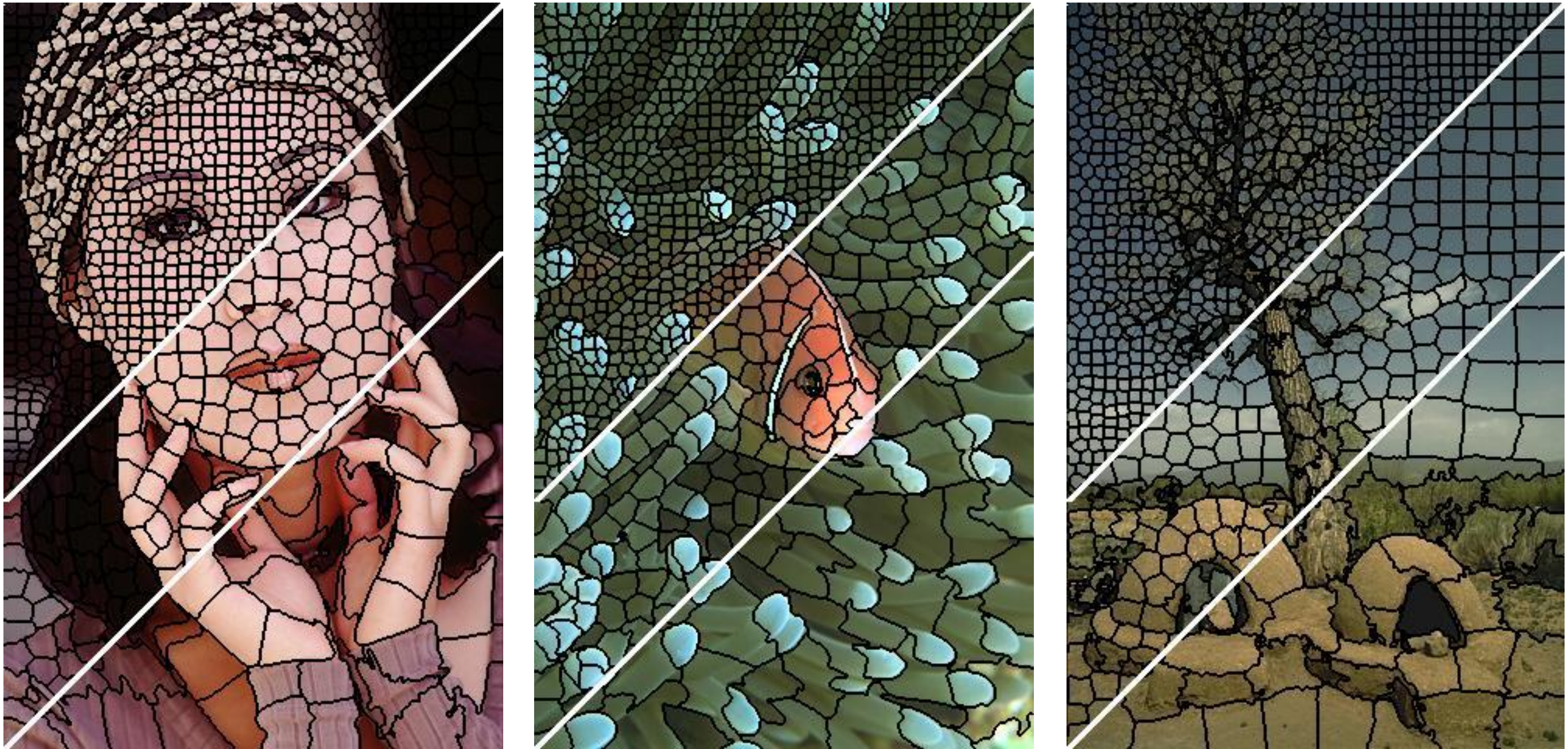
***k* = 5**

***k* = 10**

K-means clustering examples: Superpixels

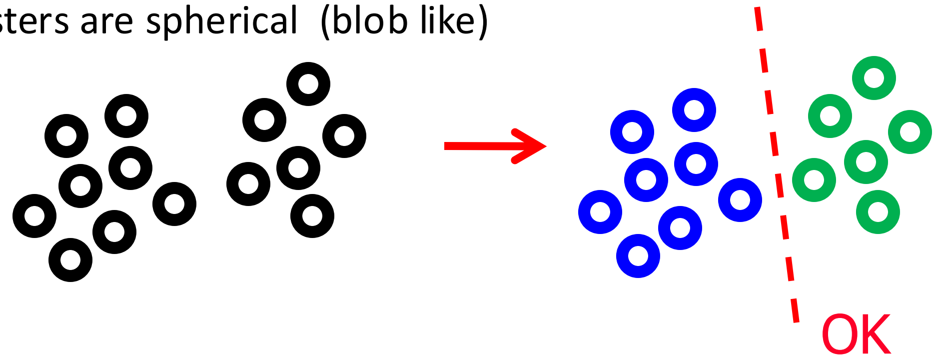
- Apply K-means to RGBXY features

[SLIC superpixels, Achanta et al., PAMI 2011]

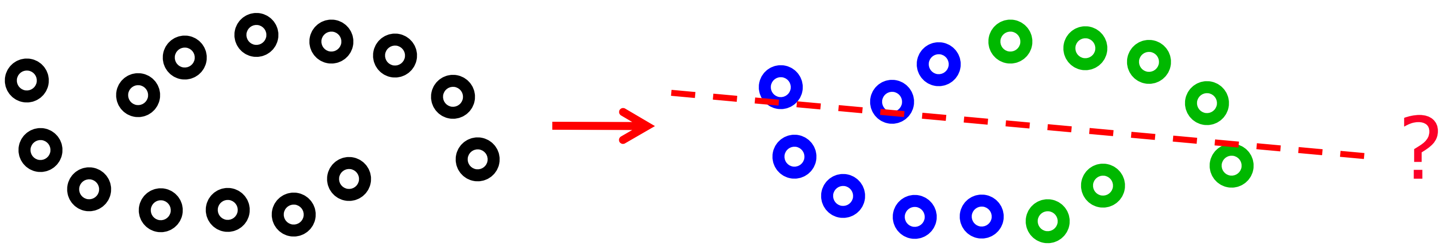


K-means Properties

- Works best when clusters are spherical (blob like)



- Fails for non-compact clusters

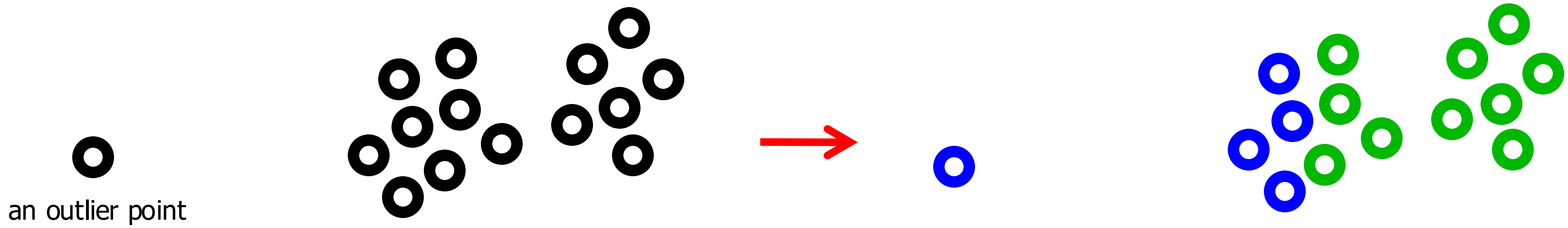


K-means produces <u>linear decision boundaries</u> between features f_p **(why?)**

Thus, K-means does not work if two clusters can not be separated by a line/plane, *i.e.* if the data is linearly non-separable.

K-means Properties

- Sensitive to outliers

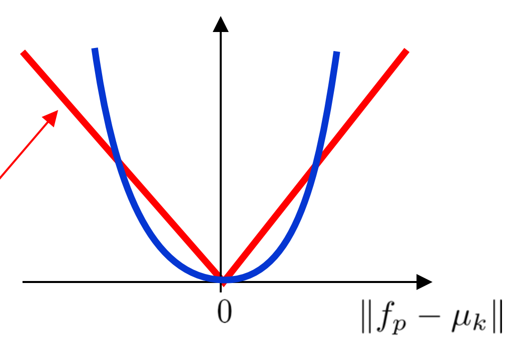


Explanation: squared distance error grows too fast making any outlier extremely costly. This also explains non-robustness of a "sample mean" statistic.

$$SSE = \sum_{k=1}^{K} \sum_{p \in S^k} \| f_p - \mu_k \|^2$$

Possible solution: replace squared distances by <u>absolute distances</u> that grow at a slower pace.

$$SAE = \sum_{k=1}^{K} \sum_{p \in S^k} \| f_p - \mu_k \|$$



Interestingly, in this case the optimal value of μ_k is the "median" of set S^k instead of its "mean"

K-means Summary

Good

- Principled (objective function) approach to clustering
- Simple to implement (the approximate iterative optimization)
- Fast

Not so good

- Only a local minimum is found (sensitive to initialization)
- May fail for non-blob like clusters
- Maybe sensitive to outliers
- How to choose ***K*** ?

Can add **sparsity/complexity** term making K an additional variable

$$E(S,\mu,K) \;=\; \sum_{k=1}^{K}\sum_{p\in S^k}\|f_p-\mu_k\|^2 \;+\; \gamma\,|K|$$

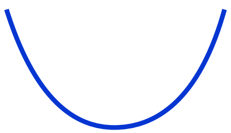
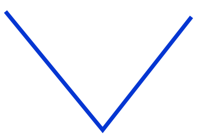
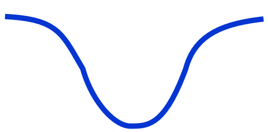
Akaike Information Criterion (AIC) or
Bayesian Information Criterion (BIC)

(generalization)

Distortion Clustering

can use different "distortion" measures

$$E(S,\mu) \;=\; \sum_{k=1}^{K}\sum_{p\in S_k} \|f_p-\mu_k\|_d$$

	examples of distortion measure $\|\cdot\|_d$		interpretation of parameters μ_k
	$\|\cdot\|_d \;=\; \|\cdot\|^2$	squared L_2 norm	**K-means**
	$\|\cdot\|_d \;=\; \|\cdot\|$	absolute L_2 norm	**K-medians**
	$\|\cdot\|_d \;=\; 1-\exp(-\|\cdot\|^2)$		**K-modes**

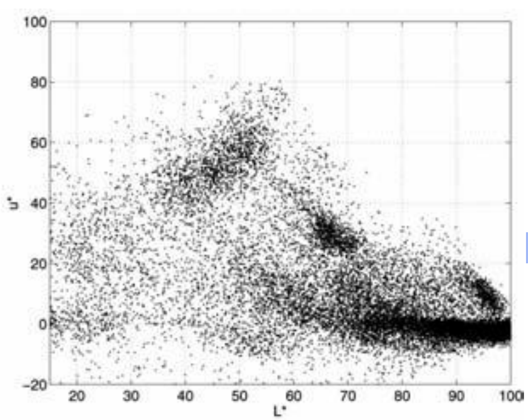
NOTE: besides changing the distortion measure, there are different generalizations of K-means requiring **other interpretations of SSE objective**

K-modes clustering

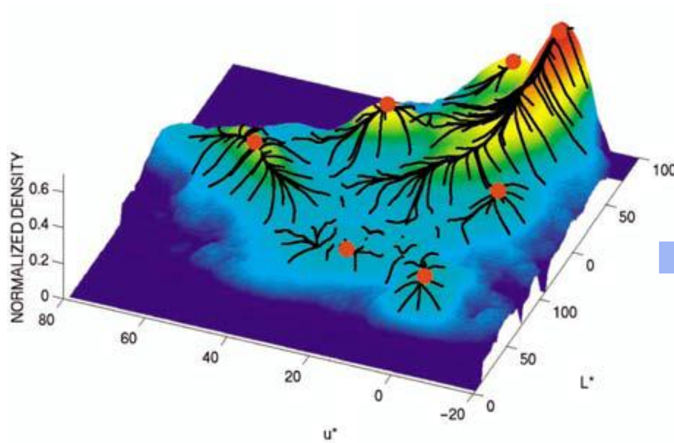
From “**means**” towards “**modes**” clustering:

Kernel-based ***mode clustering***

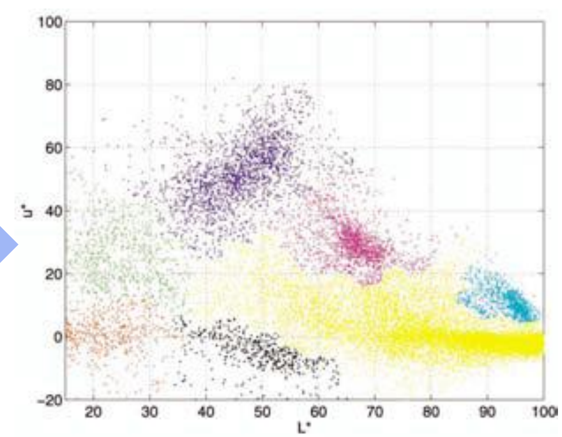
- Formulate clustering as *histogram partitioning*
  - look for **modes** in data histograms
  - assign points to modes



data points

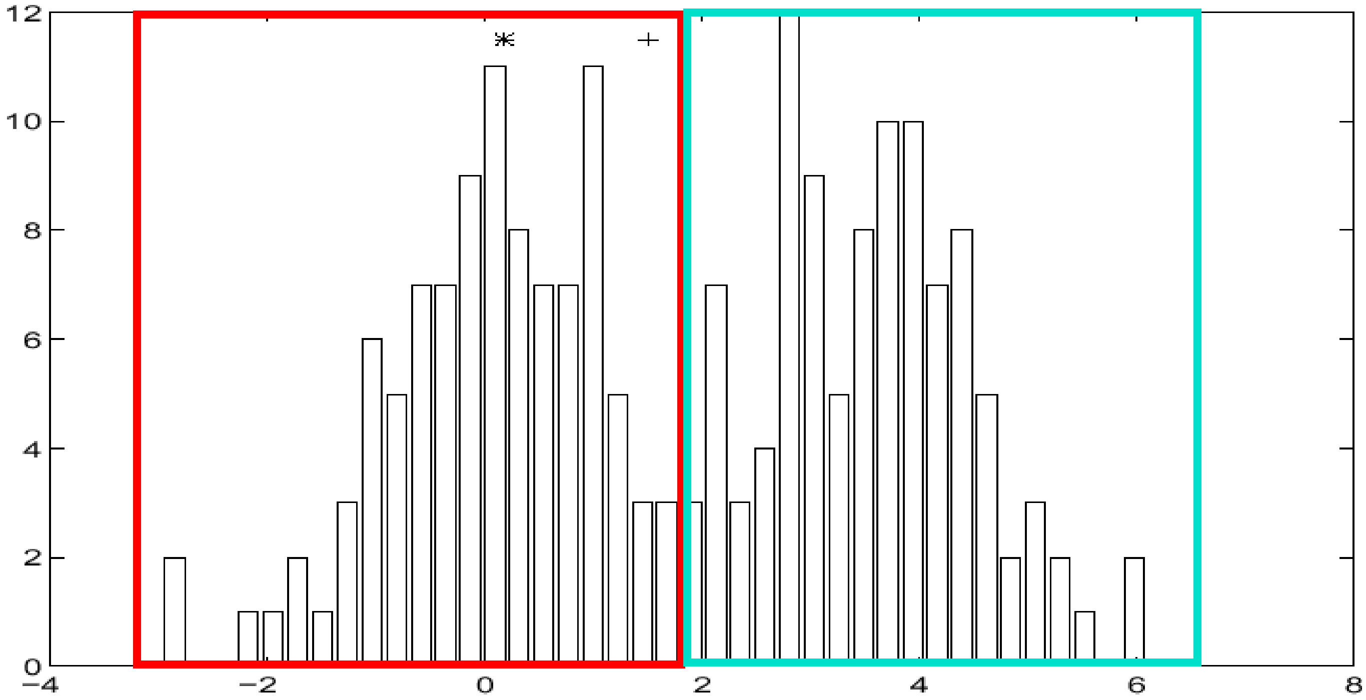


data histogram and its modes



clustering

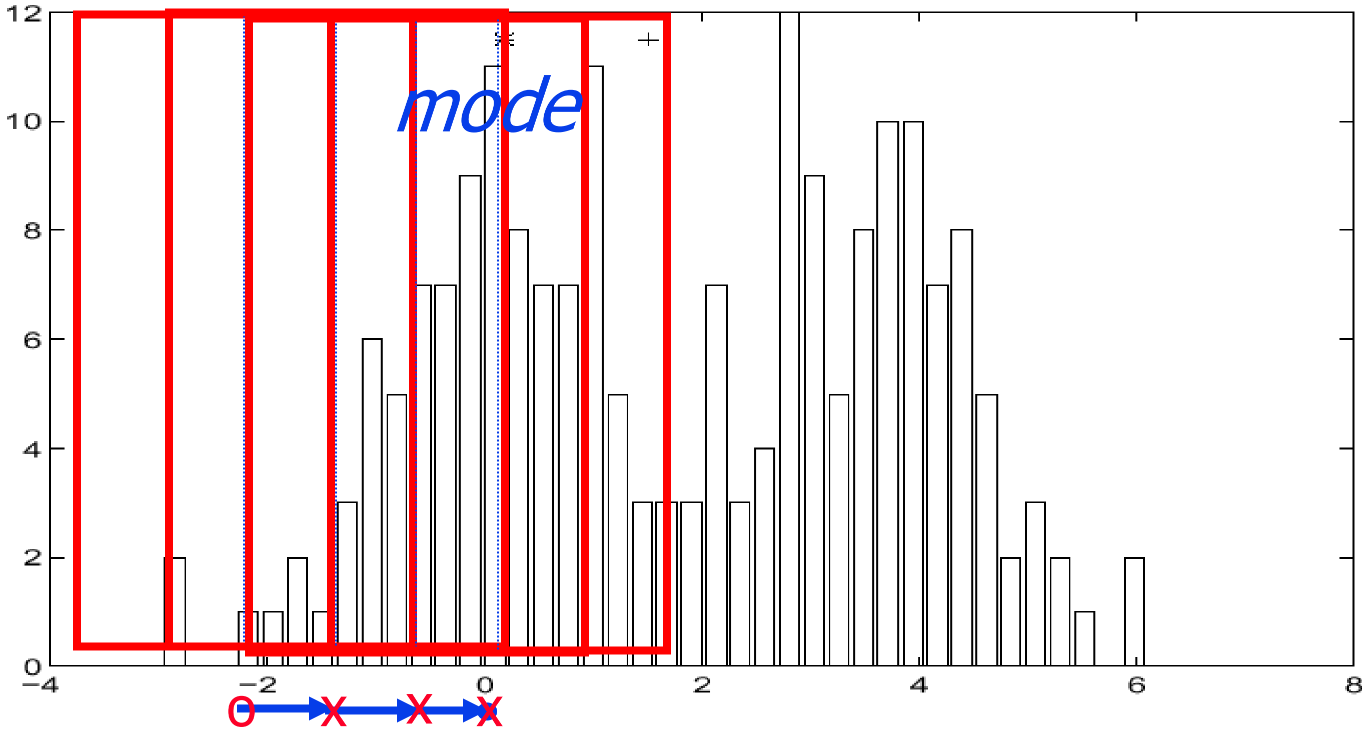
Finding Modes in a Histogram



- How Many Modes Are There?
  - Easy to see, not too obvious how to compute

Mean Shift

[Fukunaga and Hostetler 1975, Cheng 1995, Comaniciu & Meer 2002]



Iterative Mode Search

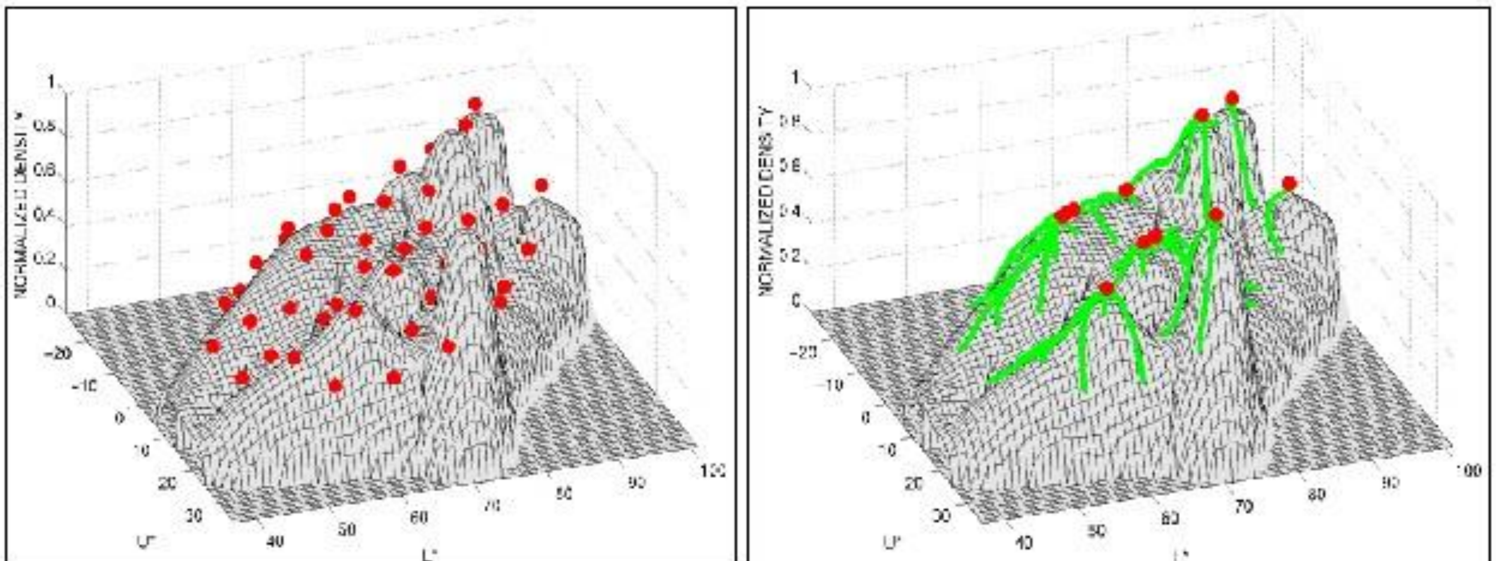
1. Initialize random seed, and fixed window
2. Calculate center of gravity 'x' of the window (the"mean")
3. Translate the search window to the mean
4. Repeat Step 2 until convergence

Mean Shift

[Fukunaga and Hostetler 1975, Cheng 1995, Comaniciu & Meer 2002]

Multimodal Distributions

- Parallel processing of an initial tessellation.
- Pruning of mode candidates.
- Classification based on the basin of attraction.



Mean shift trajectories

Mean Shift as K-modes

[Salah, Mitche, Ben-Ayed 2010]

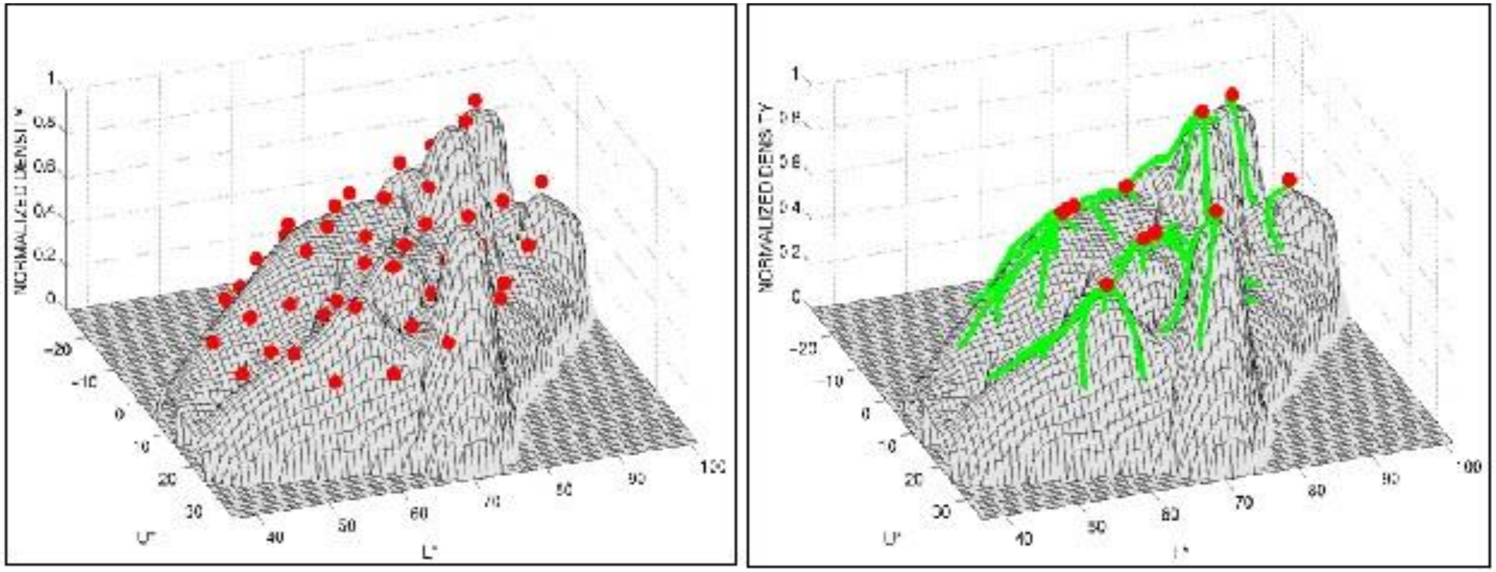
$$\sum_{k=1}^{K}\sum_{p\in S^k}\left\|f_p-\mu_k\right\|_d \qquad \|\cdot\|_d \;:$$

quadratic (*K-means*)

absolute (*K-medians*)

bounded (*K-modes*)

Mean-shift segmentation relates to distortion clustering with a bounded loss (**K-modes**)

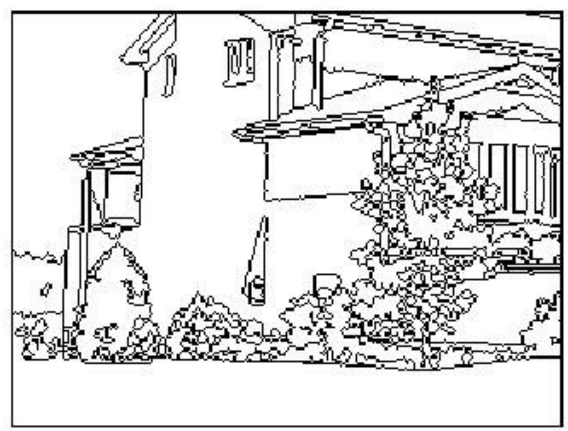


Mean shift trajectories

Mean-shift results for segmentation

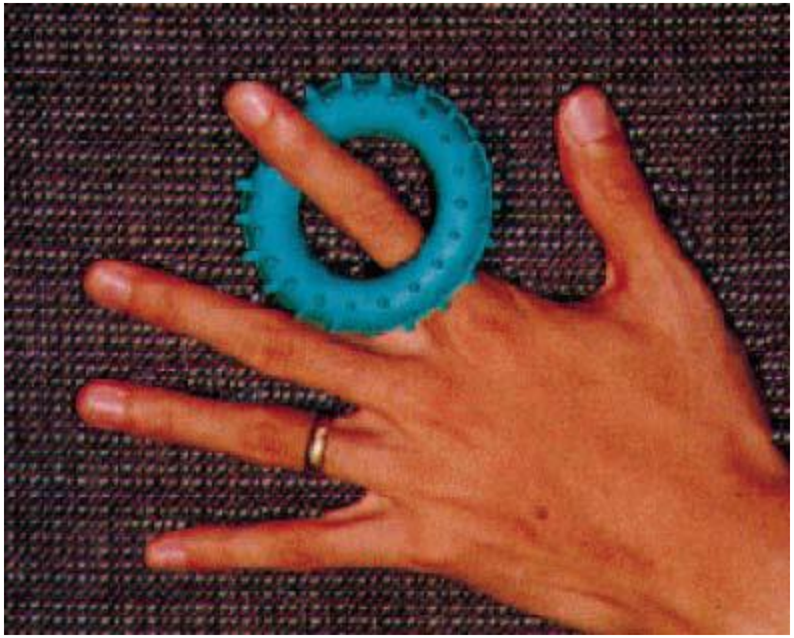
RGB+XY clustering
[Comaniciu & Meer 2002]

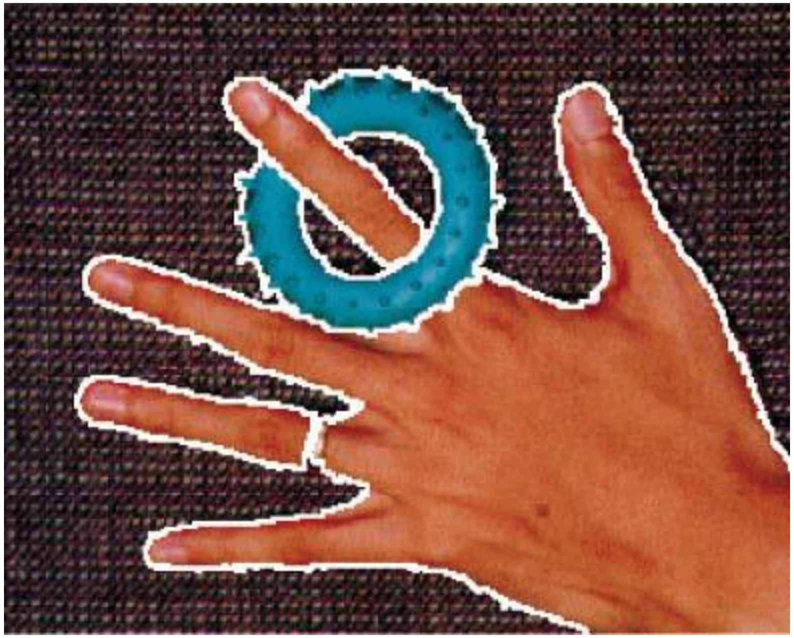
Figure 2: The *house* image, 255 × 192 pixels, 9603 colors.



Mean-shift results for segmentation

RGB+XY clustering
[Comaniciu & Meer 2002]





Mean-shift results for segmentation

RGB+XY clustering
[Comaniciu & Meer 2002]

works well for segments with near-consistent color

What have we learned today

- K-means clustering
- K-modes clustering via mean-shift