Machine Learning and Category Representation

Jakob Verbeek

November 25, 2011

Course website:

http://lear.inrialpes.fr/~verbeek/MLCR.11.12.php

Plan for the course

- Class 1, November 25 2011
  - Cordelia Schmid: Local invariant features
  - Jakob Verbeek: Clustering with k-means, mixture of Gaussians

- Class 2, December 2 2011
  - Cordelia Schmid: Local features 2 + Instance-level recognition
  - Jakob Verbeek: EM for mixture of Gaussian clustering + classification
  - Student presentation 1: Scale and affine invariant interest point detectors, Mikolajczyk, Schmid, IJCV 2004.

- Class 3, December 9 2011
  - Jakob Verbeek: Linear classifiers
  - Cordelia Schmid: Bag-of-features models for category classification
  - Student presentation 2: Visual categorization with bags of keypoints Csurka, Dance, Fan, Willamowski, Bray, ECCV 2004

Plan for the course

- Class 4, December 16 2011
  - Jakob Verbeek: Non-linear kernels + Fisher vector image representation
  - Cordelia Schmid: Category level localization
  - Student presentation 3: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories.
  - Student presentation 4: Video Google: A Text Retrieval Approach to Object Matching in Videos
- Class 5, January 6 2012
  - Cordelia Schmid: TBA
  - Student presentation 5: Object Detection with Discriminatively Trained Part Based Models.
  - Student presentation 6: Learning realistic human actions from movies Laptev, Marszalek, Schmid, Rozenfeld, CVPR 2008.
- Class 6, January 13 2012
  - Jakob Verbeek: TBA
  - Student presentation 7: High-dimensional signature compression for large-scale image classification
  - Student presentation 8: Segmentation as Selective Search for Object Recognition, van de Sande, Uijlings, Gevers, Smeuldersm, ICCV 2011.

Visual recognition - Objectives

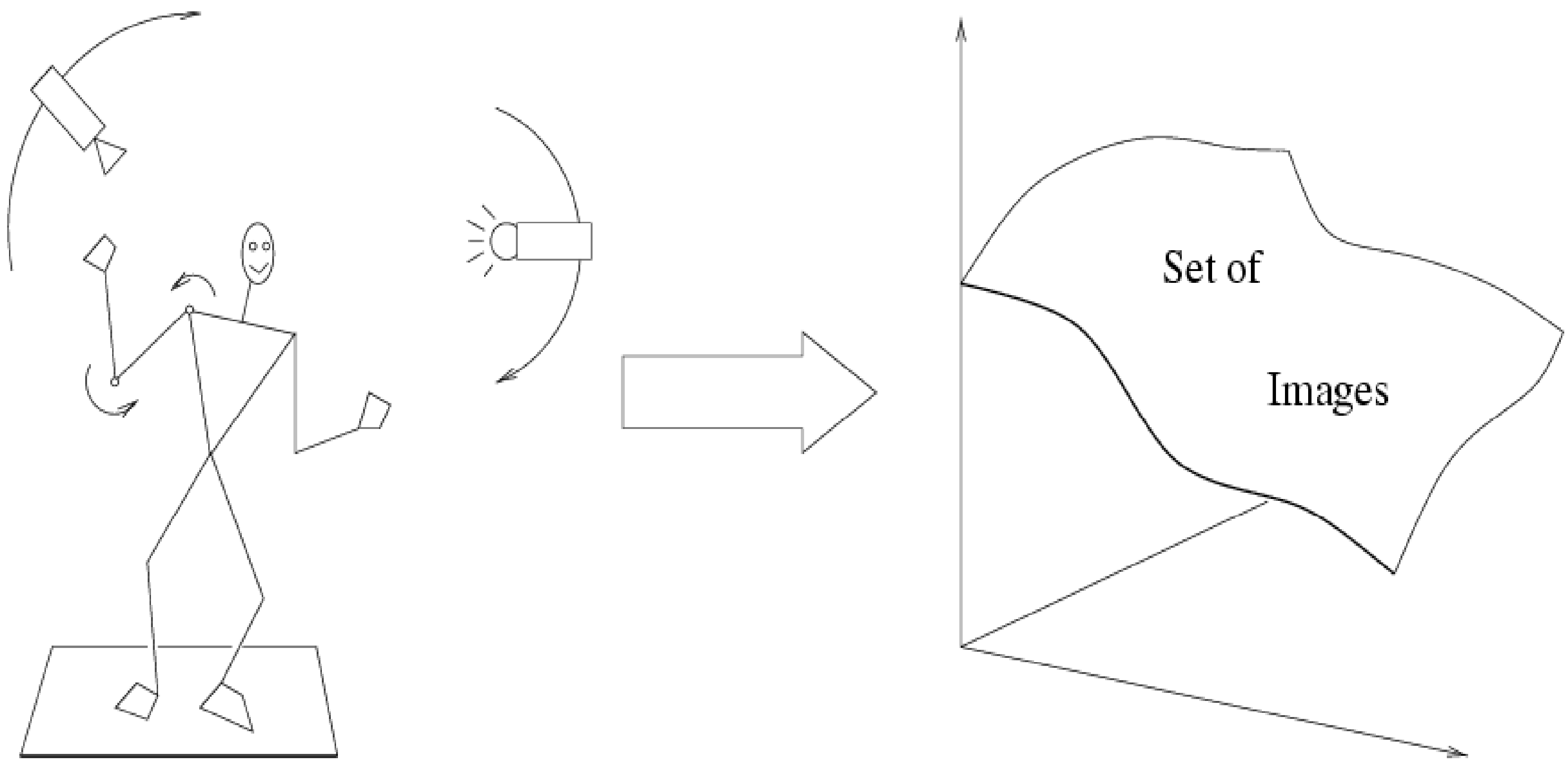
- Image classification: assigning label to the image

Car: present
Cow: present
Bike: not present
Horse: not present
...

- Object localization: define the location and the category

Category label
+ location

Difficulties: within object variations



Variability in appearance of the same object:
Viewpoint, illumination, occlusion,
articulation of deformable objects, ...

Difficulties: within-class variations





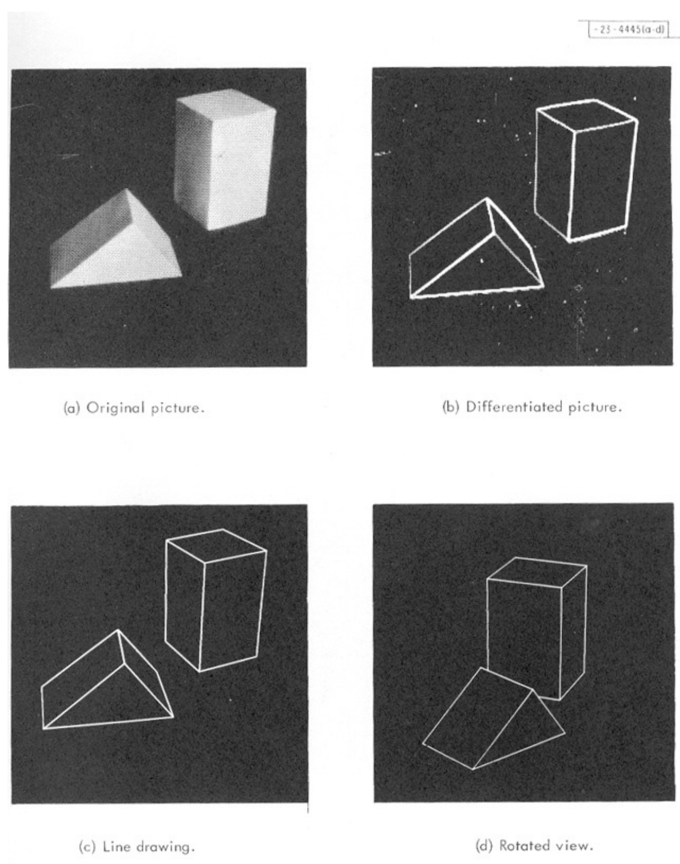

Visual category recognition

- Robust image description
  - Appropriate descriptors for objects and categories

- Statistical modeling and machine learning
  - Automatic modeling from category instances
    - scene types
    - object categories
    - human actions

Why machine learning?

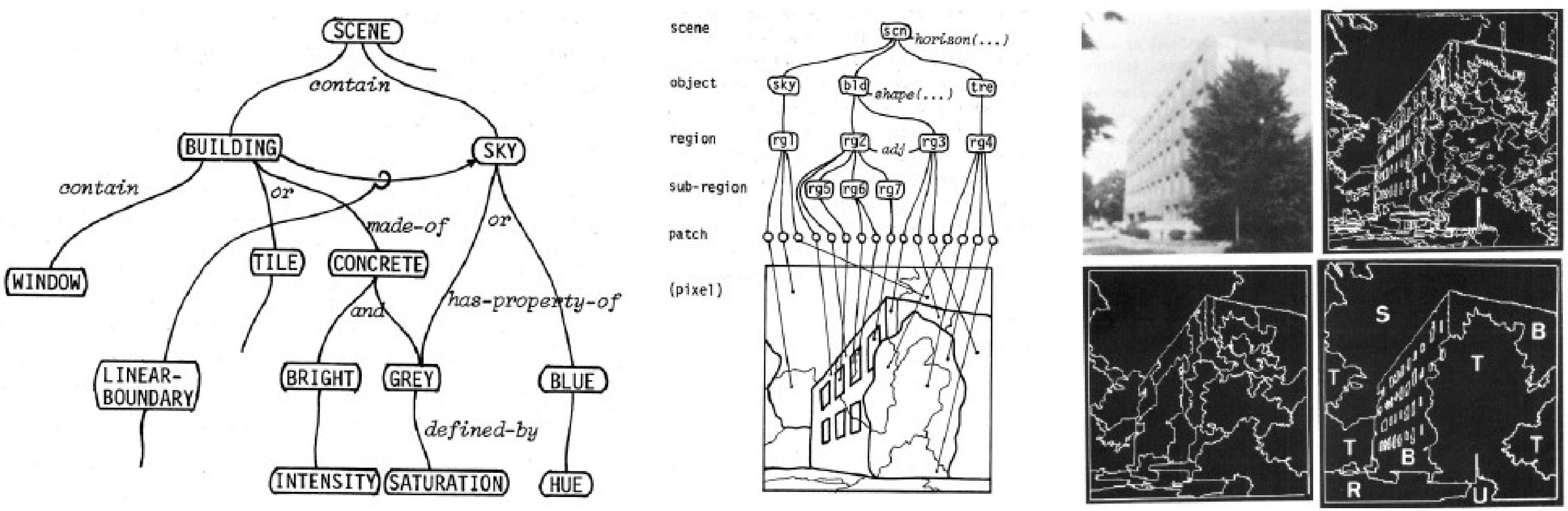
- Early approaches: simple features + handcrafted models
- Can handle only few images, simples tasks



L. G. Roberts, *Machine Perception of Three Dimensional Solids,* Ph.D. thesis, MIT Department of Electrical Engineering, 1963.

Why machine learning?

- Early approaches: manual programming of rules
- Tedious, limited and does not take into account the data



(a) Bottom-up process (b) Top-down process (c) Result

Figure 3. A system developed in 1978 by Ohta, Kanade and Sakai [33, 32] for knowledge-based interpretation of outdoor natural scenes. The system is able to label an image (c) into semantic classes: S-sky, T-tree, R-road, B-building, U-unknown.

*Y. Ohta, T. Kanade, and T. Sakai, "*An Analysis System for Scenes Containing objects with Substructures," *International Joint Conference on Pattern Recognition*, 1978.

Why machine learning?

- Today lots of data, complex tasks

Internet images,
personal photo albums

Movies, news, sports

Why machine learning?

- Today lots of data, complex tasks

Internet images,
personal photo albums

Movies, news, sports

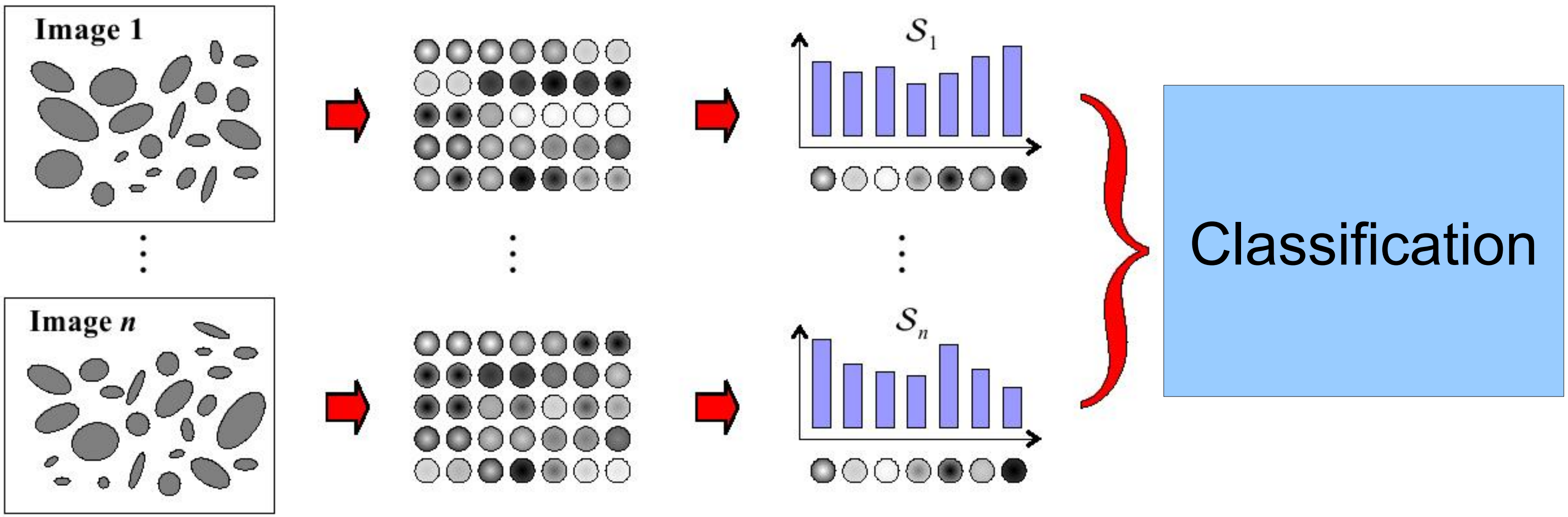
- Instead of trying to define rules manually, learn them automatically from examples

Bag-of-words image classification

- Excellent results in the presence of
  - background clutter, occlusion, lighting, viewpoint,...

bikes books building cars people phones trees

Bag-of-features for image classification



Extract regions

Compute descriptors

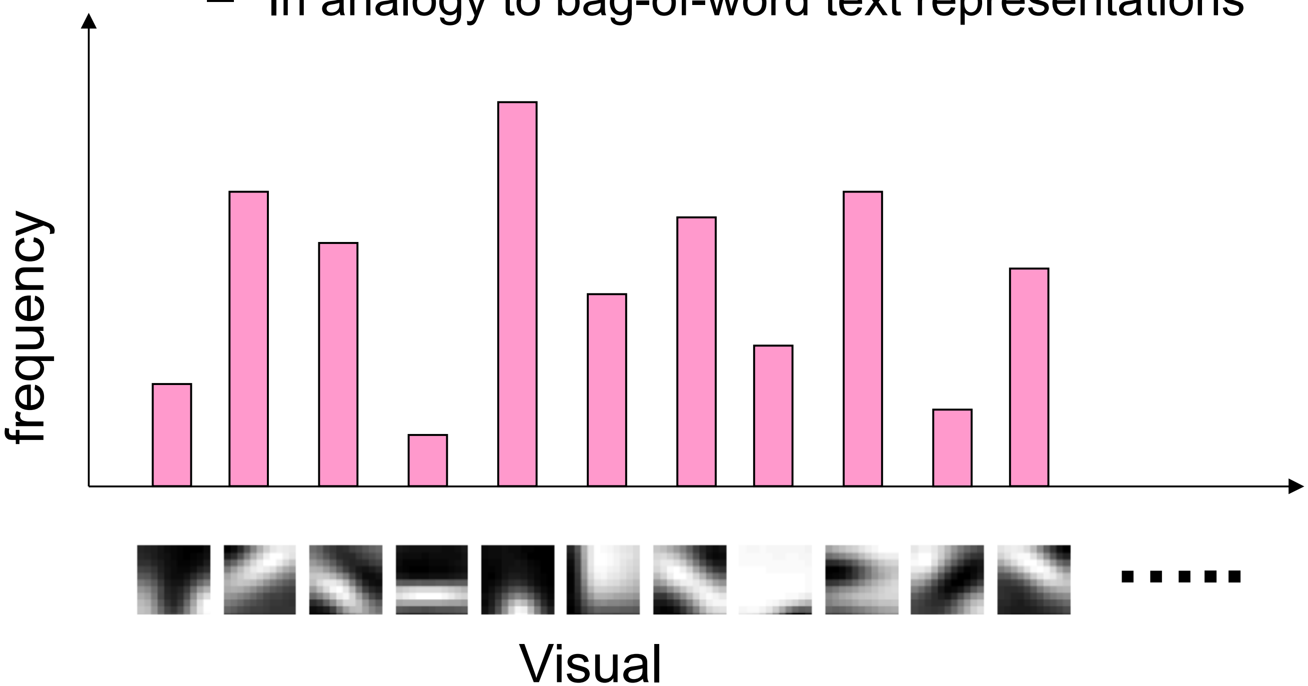
Find clusters and frequencies

From local descriptors to Bag-of-Words

1) Detect local regions in image (eg. interest point detector)
2) Compute local descriptors (eg. SIFT)

- Image now represented by a set of *N* local descriptors

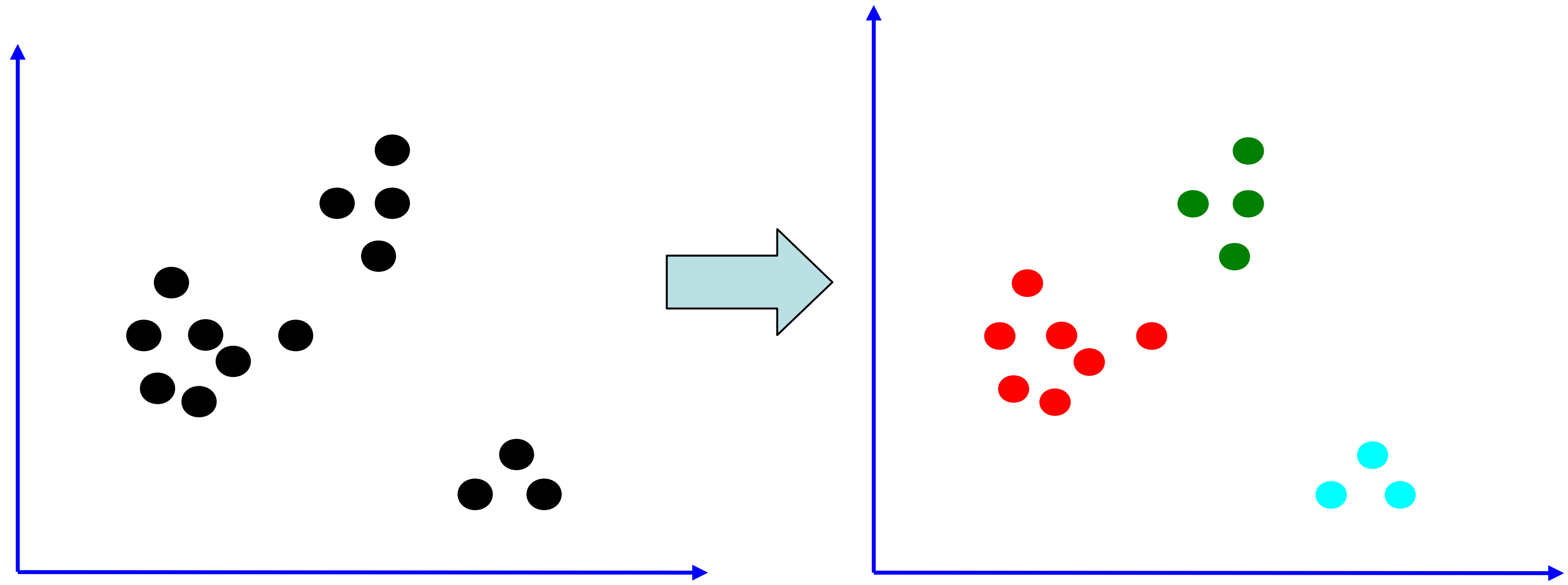
- Map each local descriptor to one out of *K* “visual words”
- Image now represented by visual word histogram of length *K*
  - In analogy to bag-of-word text representations



Clustering

- Finding a group structure in the data
  - Data in one cluster similar to each other
  - Data in different clusters dissimilar

- Map each data point to a discrete cluster index
  - "flat" methods find *k* groups
  - "hierarchical" methods define a tree structure over the data

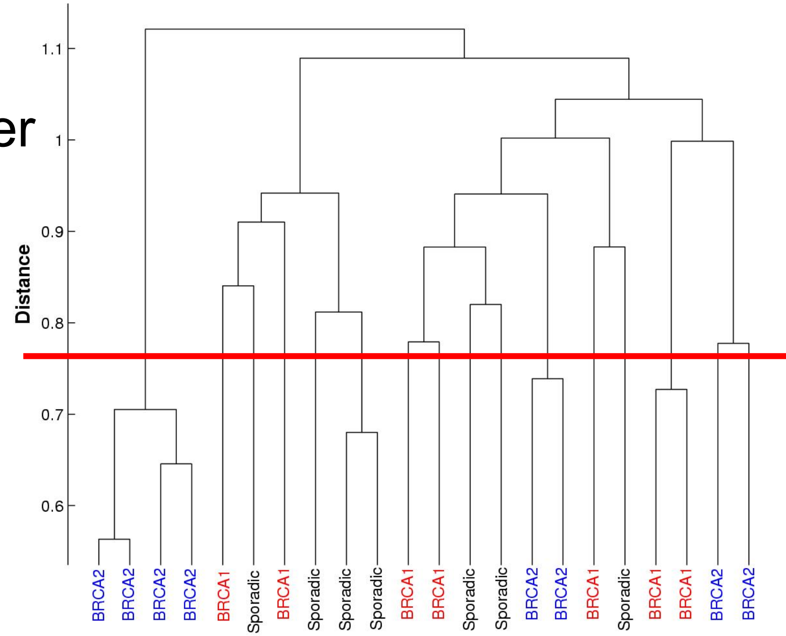


Hierarchical Clustering

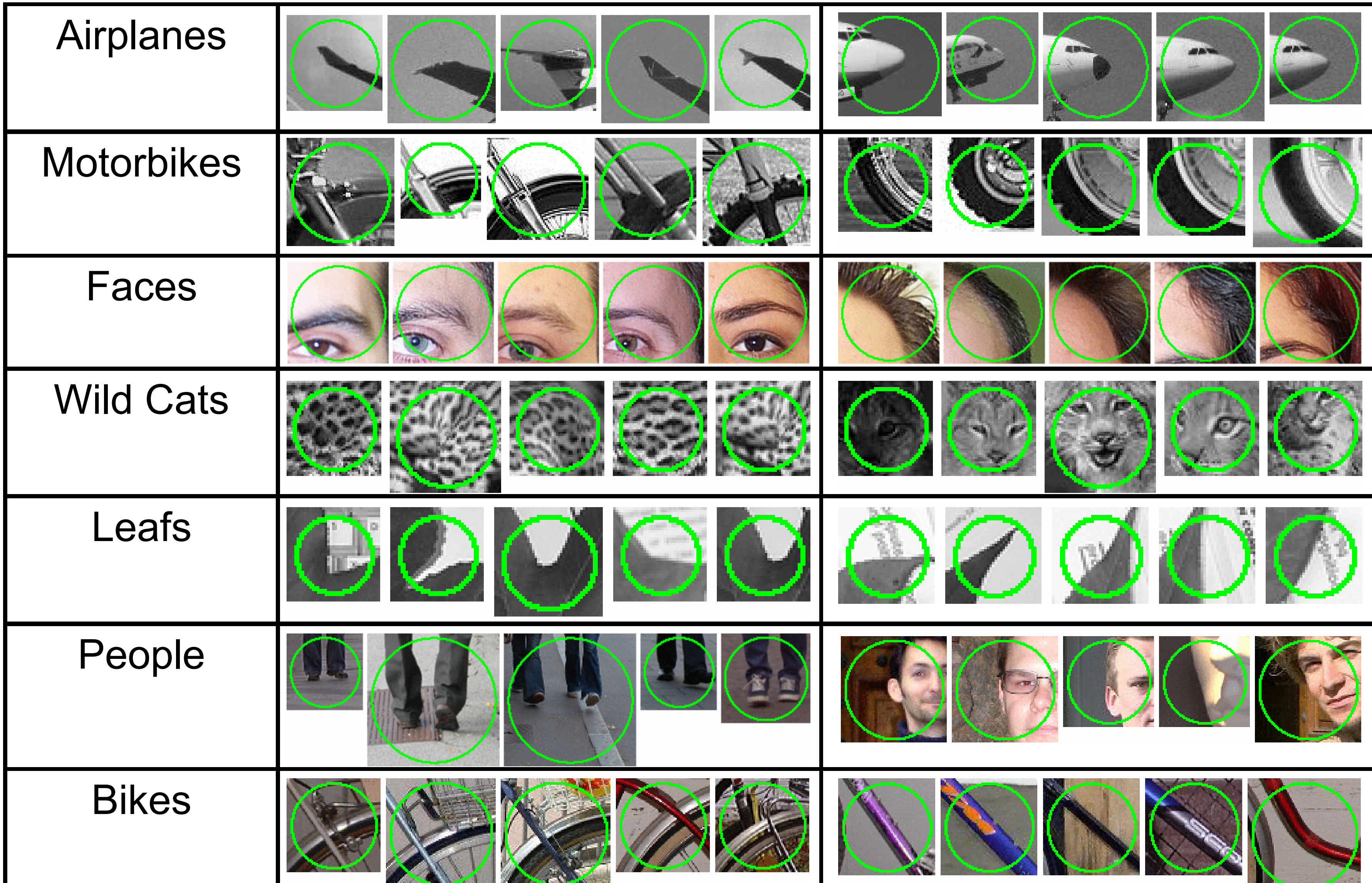
- Data set is partitioned into a tree structure

- Top-down construction
  - Start all data in one cluster: root node
  - Apply "flat" clustering into k groups
  - Recursively cluster the data in each group

- Bottom-up construction
  - Start with all points in separate cluster
  - Recursively merge "closest" clusters
  - Distance between clusters A and B
    - Min, max, or mean distance between x in A, and y in B

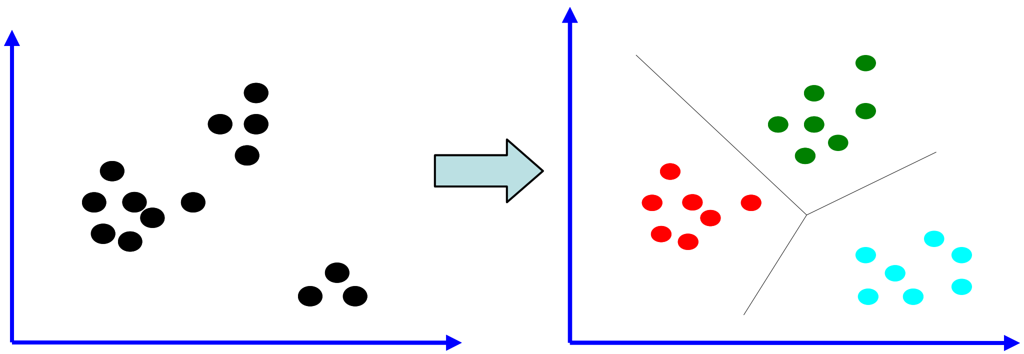


Clustering example: visual words

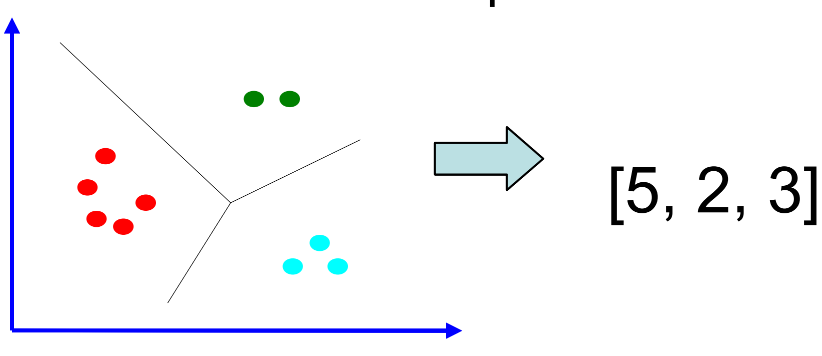


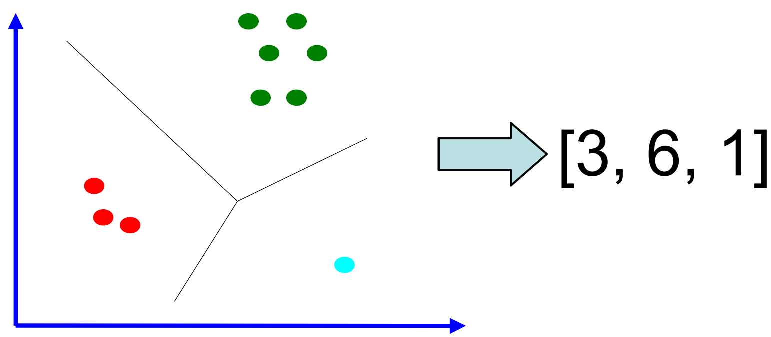
Clustering descriptors into visual words

- Offline training: Find groups of similar local descriptors
  - Using many descriptors from training images



- New image:
  - Detect local regions
  - Compute local descriptors
  - Count descriptors in each cluster





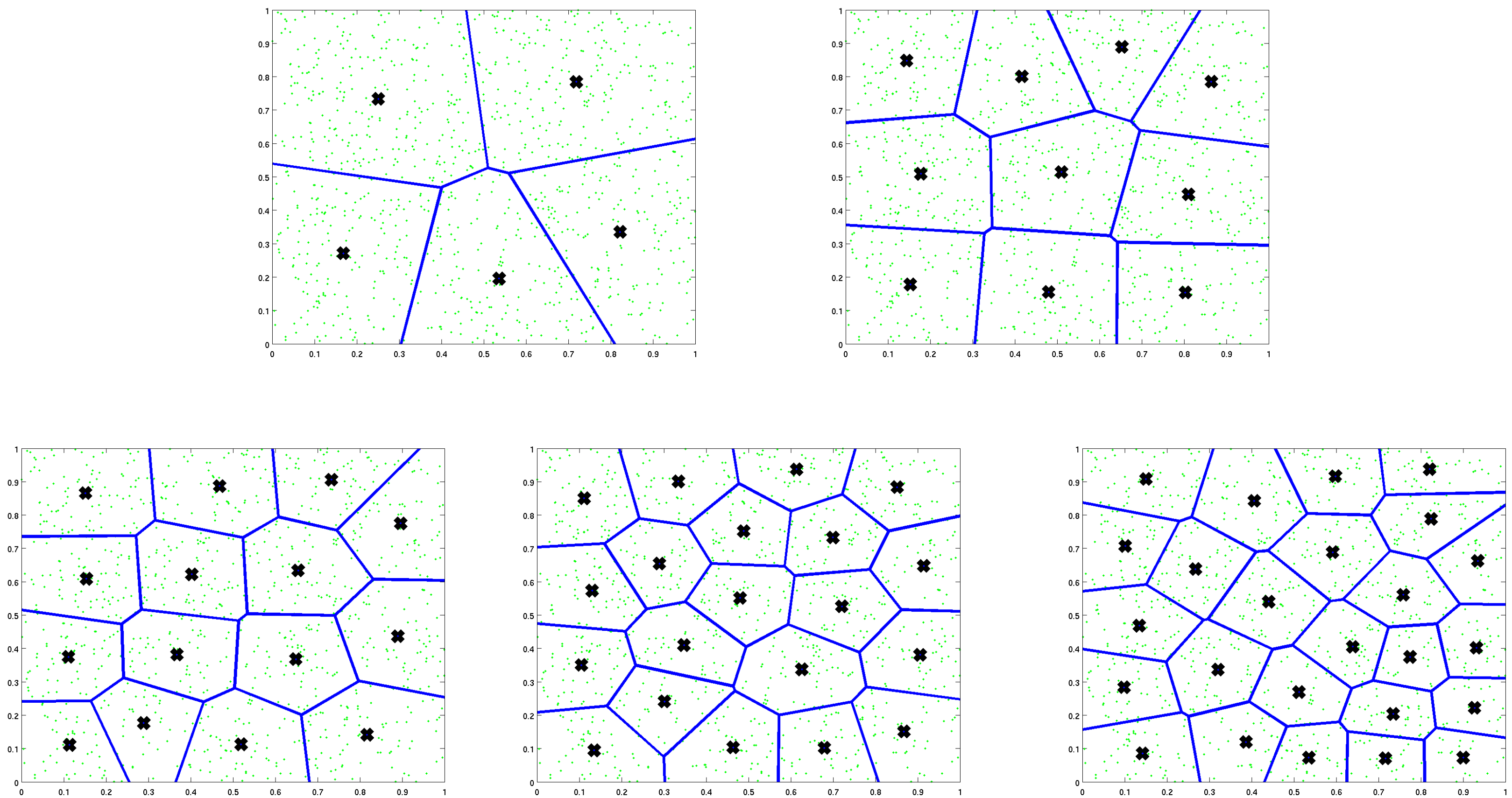
Definition of k-means clustering

- Given: data set of N points x_n, n=1,…,N
- Goal: find K cluster centers m_k, k=1,…,K
- Clustering: assignment of data points to cluster centers
  - Indicator variables r_{nk}=1 if x_n assgined to x_n, r_{nk}=0 otherwise
- Error criterion: sum of squared distances between each data point and assigned cluster center

$$E(\{m_k\}_{k=1}^{K}) = \sum_n \sum_k r_{nk} \|x_n - m_k\|^2$$

Examples of k-means clustering

- Data uniformly sampled in unit square, running k-means with 5, 10, 15, 20 and 25 centers



Minimizing the error function

$$E\left(\{m_k\}_{k=1}^{K}\right)=\sum_n \sum_k r_{nk}\|x_n-m_k\|^2$$

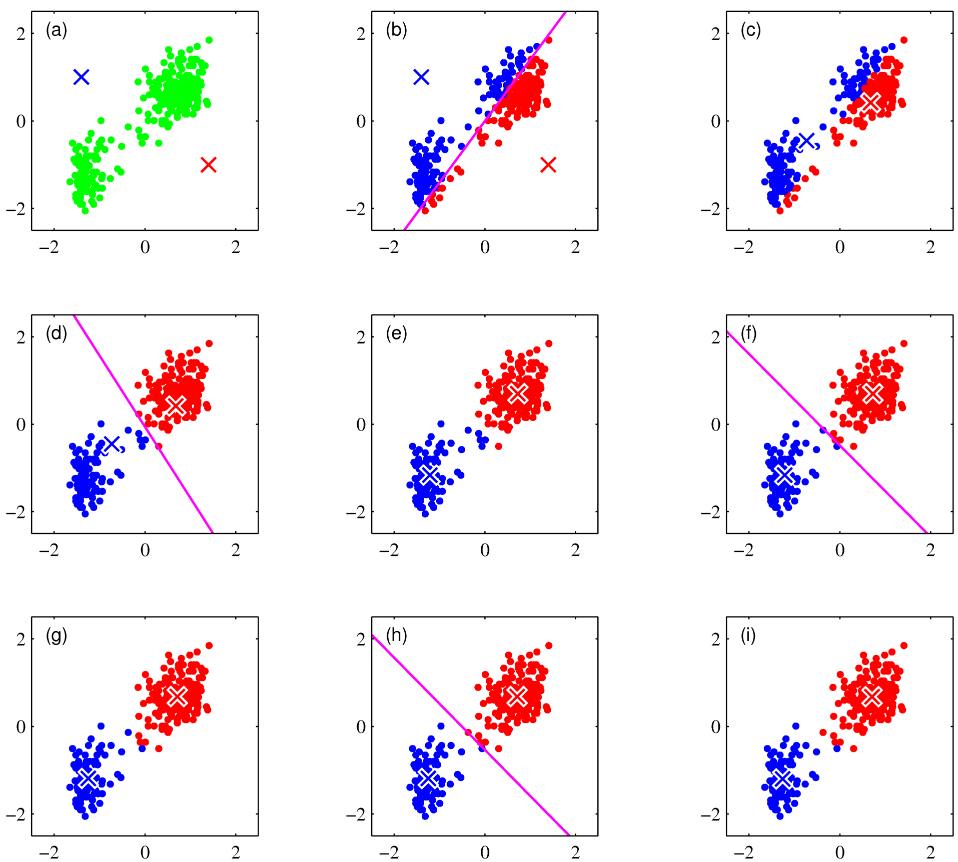
- Goal find centers m_k and assignments r_{nk} to minimize the error function

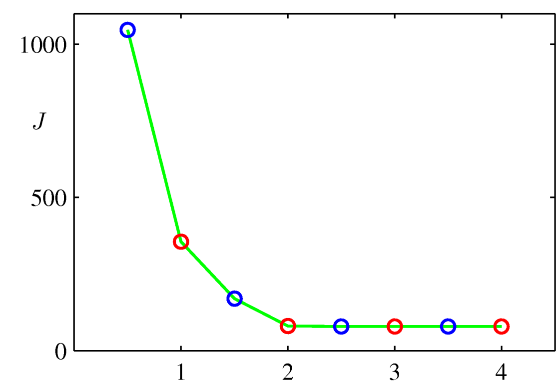
- An iterative algorithm
  1) Initialize cluster centers, eg. on randomly selected data points
  2) Update assignments r_{nk} for fixed m_k
  3) Update centers m_k for fixed data assignments r_{nk}
  4) If cluster centers changed: return to step 2)
  5) Return cluster centers

- Iterations monotonically decrease error function

Examples of k-means clustering

- Several iterations with two centers





Error function

Minimizing the error function

$$E\left(\{m_k\}_{k=1}^{K}\right) = \sum_n \sum_k r_{nk} \|x_n - m_k\|^2$$

- Update assignments r_{nk} for fixed m_k $\quad \sum_k r_{nk} \|x_n - m_k\|^2$
  - Decouples over the data points
  - Only one r_{nk} =1, rest zero
  - Assign to closest center

- Update centers m_k for fixed assignments r_{nk}
  - Decouples over the centers $\quad \sum_n r_{nk} \|x_n - m_k\|^2$
  - Set derivative to zero
  - Put center at mean of assigned data points

$$\frac{\partial E}{\partial m_k} = 2 \sum_n r_{nk} (x_n - m_k) = 0$$

$$m_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

Minimizing the error function

$$E\left(\{m_k\}_{k=1}^{K}\right)=\sum_n \sum_k r_{nk}\|x_n-m_k\|^2$$

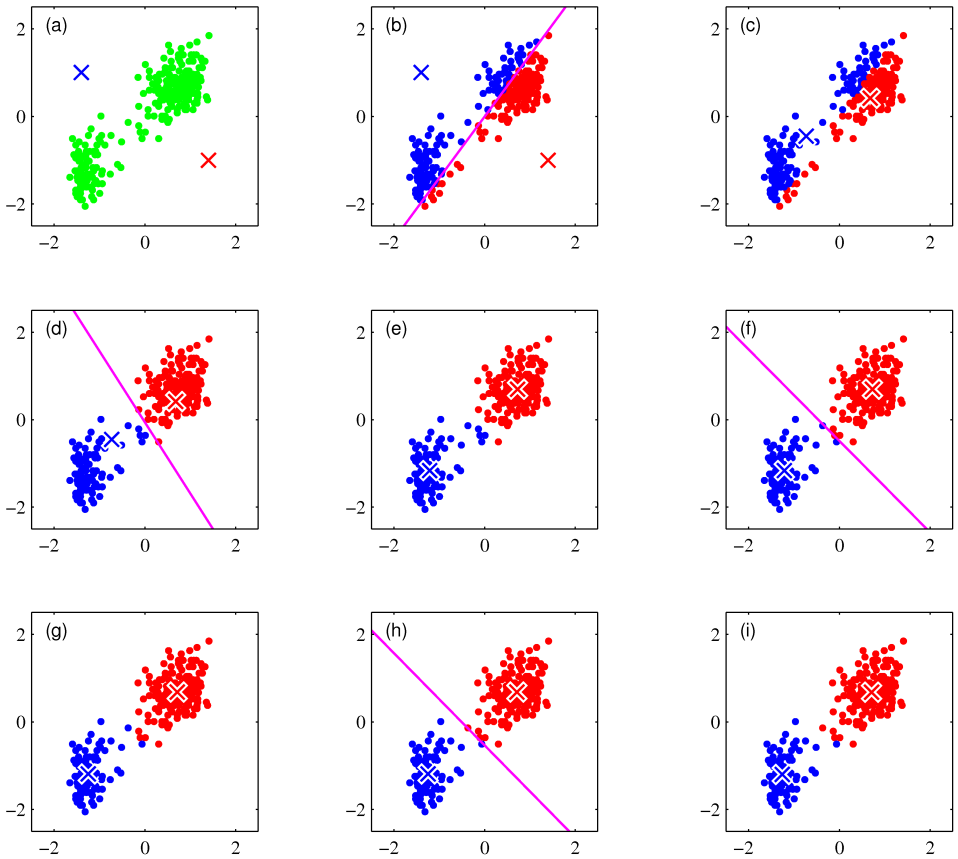
- Goal find centers m_k and assignments r_{nk} to minimize the error function

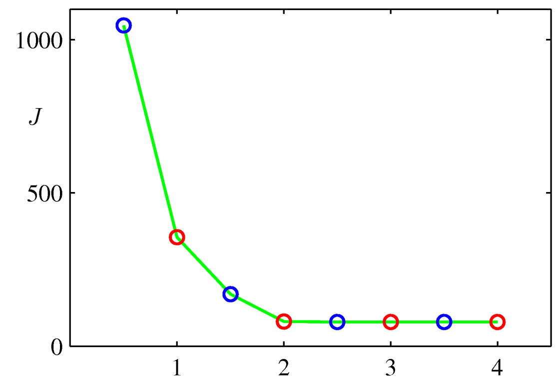
- An iterative algorithm
  1) Initialize cluster centers, somehow
  2) **Assign x_n to closest m_k**
  3) **Update centers m_k as center of assigned data points**
  4) If cluster centers changed: return to step 2)
  5) Return cluster centers

- Iterations monotonically decrease error function
  - **Both steps reduce the error function**
  - **Only a finite number of possible assignments**

Examples of k-means clustering

- Several iterations with two centers

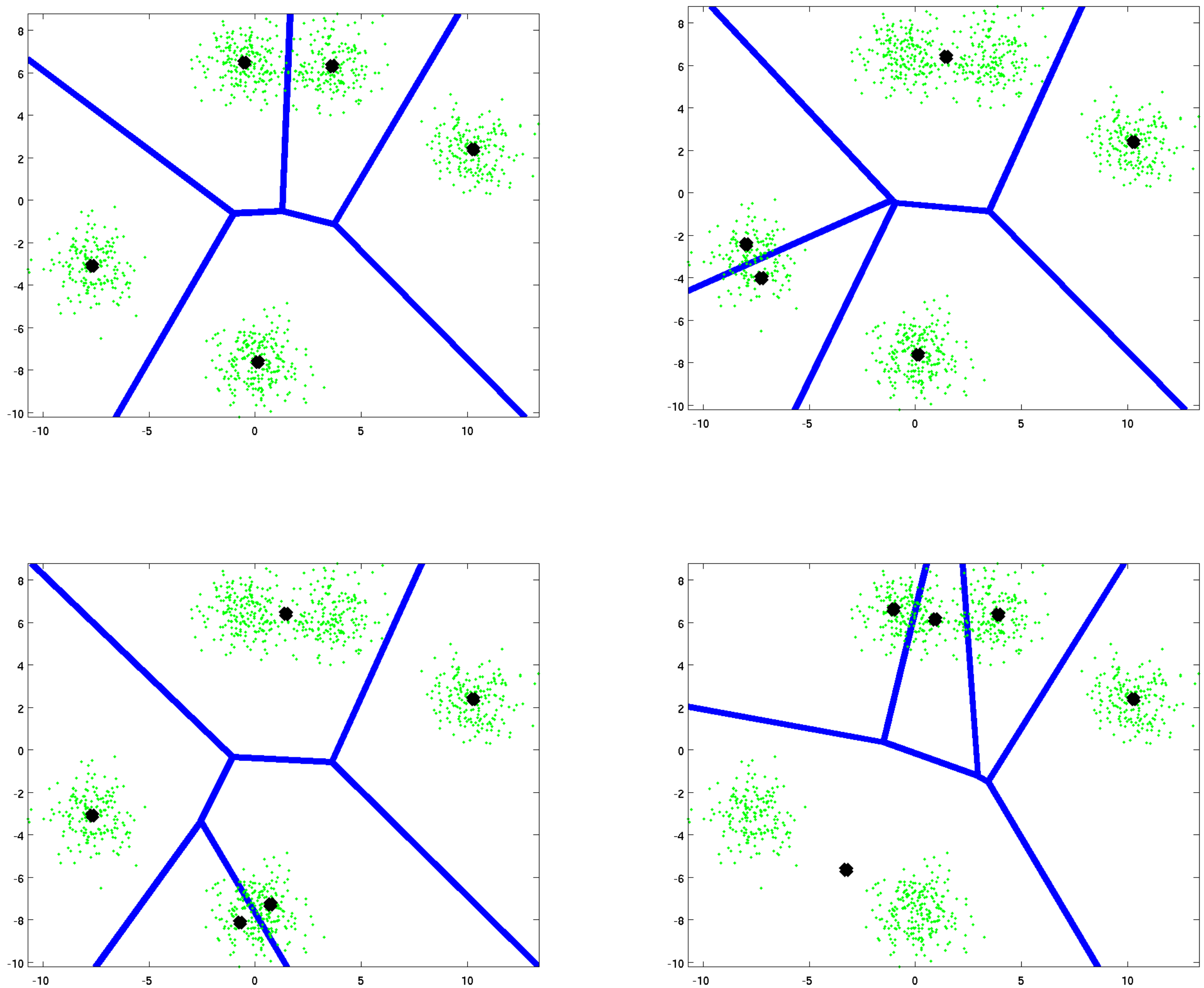




Error function

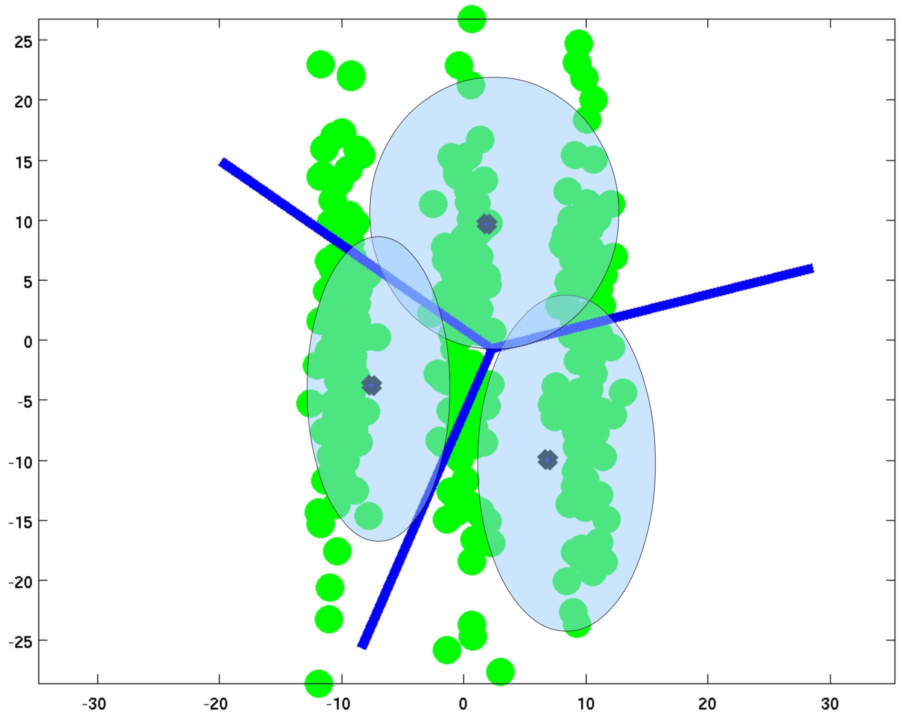
What goes wrong with k-means clustering?

- Solution depends heavily on initialization



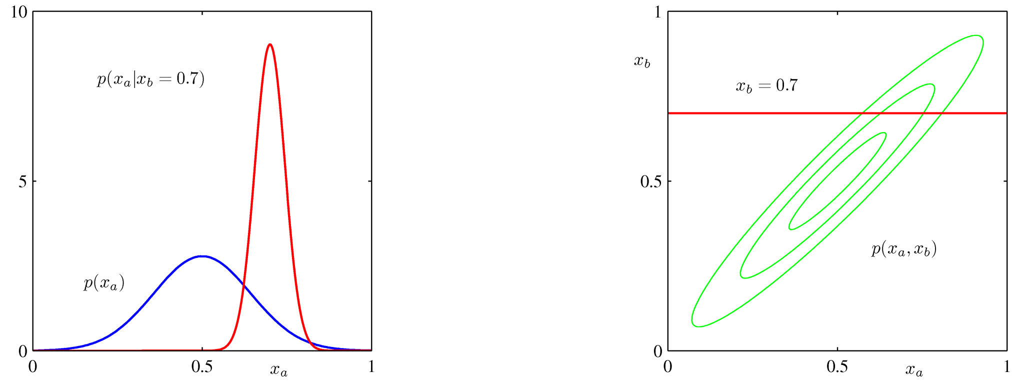
What goes wrong with k-means clustering?

- Assignment of data points to clusters is only based on the distance to the cluster center
  - No representation of the shape of the cluster
  - Let's fix this by using simple elliptical shapes



Clustering with Gaussian mixture density

- Each cluster represented by Gaussian density
  - Center, as in k-means
  - Covariance matrix: cluster spread around center



$$p(x) = N(x|m, C) = (2\pi)^{-d/2} |C|^{-1/2} \exp\left(-\frac{1}{2}(x-m)^T C^{-1}(x-m)\right)$$

Data dimension d

Determinant of covariance matrix C

Quadratic function of point x and mean m

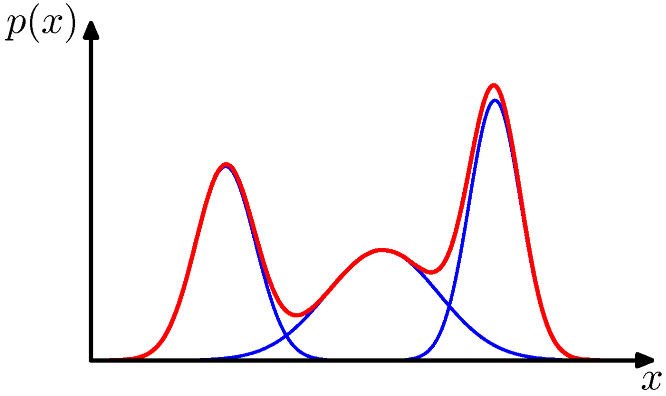
Mixture of Gaussian (MoG) density

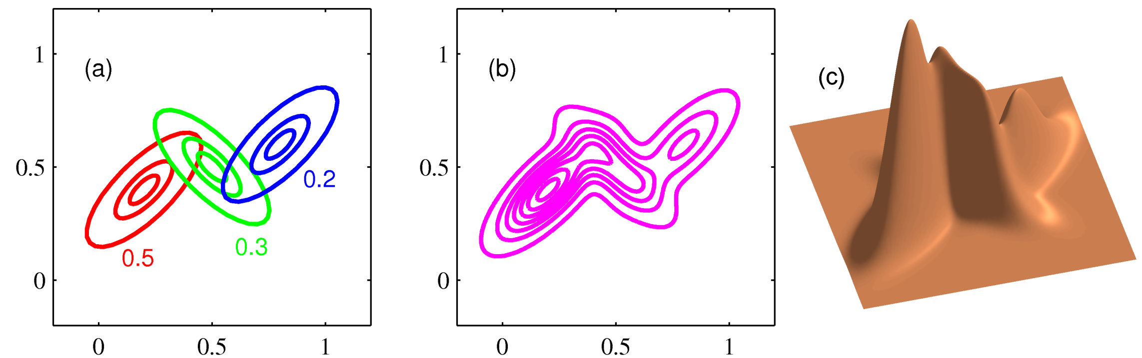
- Mixture density is weighted sum of Gaussians
  - Mixing weight: importance of each cluster

$$p(x) = \sum_{k=1}^{K} \pi_k N(x|m_k, C_k)$$

- Density has to integrate to 1, so we require

$$\pi_k \geq 0$$
$$\sum_k \pi_k = 1$$





Clustering with Gaussian mixture density

- Given: data set of N points x_n, n=1,…,N

- Find mixture of Gaussians (MoG) that best explains data
  - Maximize log-likelihood of fixed data set X w.r.t. parameters of MoG
  - Assume data points are drawn independently from MoG

$$L(\theta)=\sum_{n=1}^{N} \log p(x_n)=\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k N(x_n|m_k,C_k)$$

$$\theta=\{\pi_k, m_k, C_k\}_{k=1}^{K}$$

- MoG clustering very similar to k-means clustering
  - In addition to centers also represents cluster shape: cov. matrix
  - Also an iterative algorithm to find parameters
  - Also sensitive to initialization of paramters

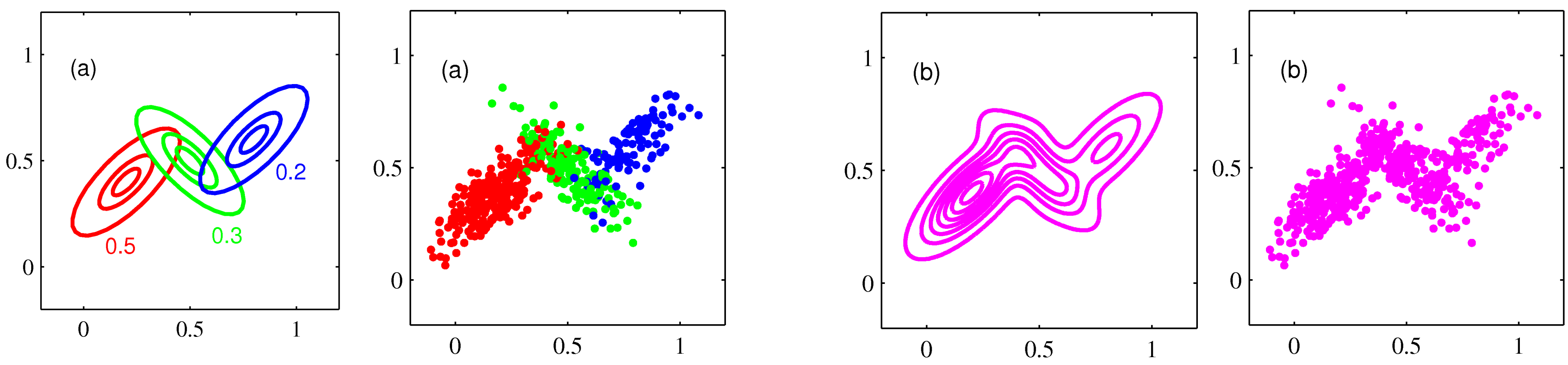
Assignment of data points to clusters

- As with k-means z_n indicates cluster index for x_n
- To sample point from MoG
  - Select cluster index k with probability given by mixing weight
  - Sample point from the k-th Gaussian
  - MoG recovered if we marginalize over the unknown cluster index

$$p(z=k) = \pi_k$$

$$p(x|z=k) = N(x|m_k, C_k)$$

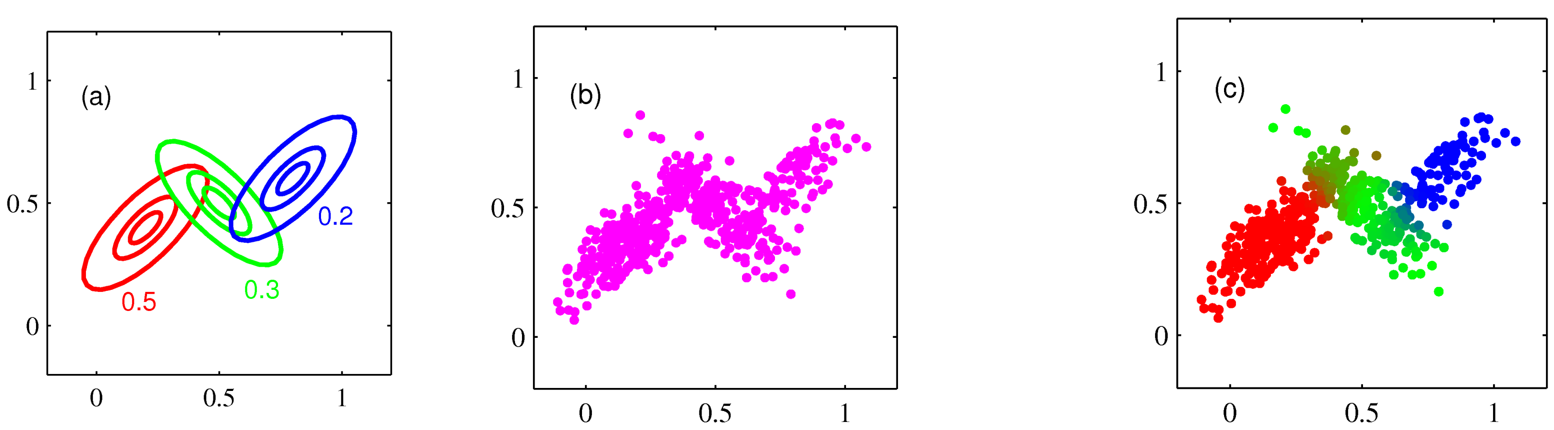
$$p(x) = \sum_k p(z=k)\, p(x|z=k) = \sum_k \pi_k N(x|m_k, C_k)$$



Soft assignment of data points to clusters

- Given data point x, infer value of z

$$p(z=k|x) = \frac{p(x,z=k)}{p(x)} = \frac{p(z=k)\,p(x|z=k)}{\sum_k p(z=k)\,p(x|z=k)} = \frac{\pi_k N(x|m_k, C_k)}{\sum_k \pi_k N(x|m_k, C_k)}$$



Maximum likelihood estimation of Gaussian

- Given data points x_n, n=1,…,N
- Find Gaussian that maximizes data log-likelihood

$$L(\theta)=\sum_{n=1}^{N}\log p(x_n)=\sum_{n=1}^{N}\log N(x_n|m,C)=\sum_{n=1}^{N}\left(-\frac{d}{2}\log\pi-\frac{1}{2}\log|C|-\frac{1}{2}(x_n-m)^T C^{-1}(x_n-m)\right)$$

- Set derivative of data log-likelihood w.r.t. parameters to zero

$$\frac{\partial L(\theta)}{\partial m}=C^{-1}\sum_{n=1}^{N}(x_n-m)=0$$

$$m=\frac{1}{N}\sum_{n=1}^{N}x_n$$

$$\frac{\partial L(\theta)}{\partial C^{-1}}=\sum_{n=1}^{N}\left(\frac{1}{2}C-\frac{1}{2}(x_n-m)(x_n-m)^T\right)=0$$

$$C=\frac{1}{N}\sum_{n=1}^{N}(x_n-m)(x_n-m)^T$$

- Parameters set as data covariance and mean

Maximum likelihood estimation of MoG

- No simple equation as in the case of a single Gaussian
- Use EM algorithm
  - Initialize MoG: parameters or soft-assign
  - E-step: soft assign of data points to clusters
  - M-step: update the cluster parameters
  - Repeat EM steps, terminate if converged
    - Convergence of parameters or assignments

- E-step: compute posterior on z given x: $q_{nk} = p(z = k | x_n)$
- M-step: update Gaussians from data points weighted by posterior

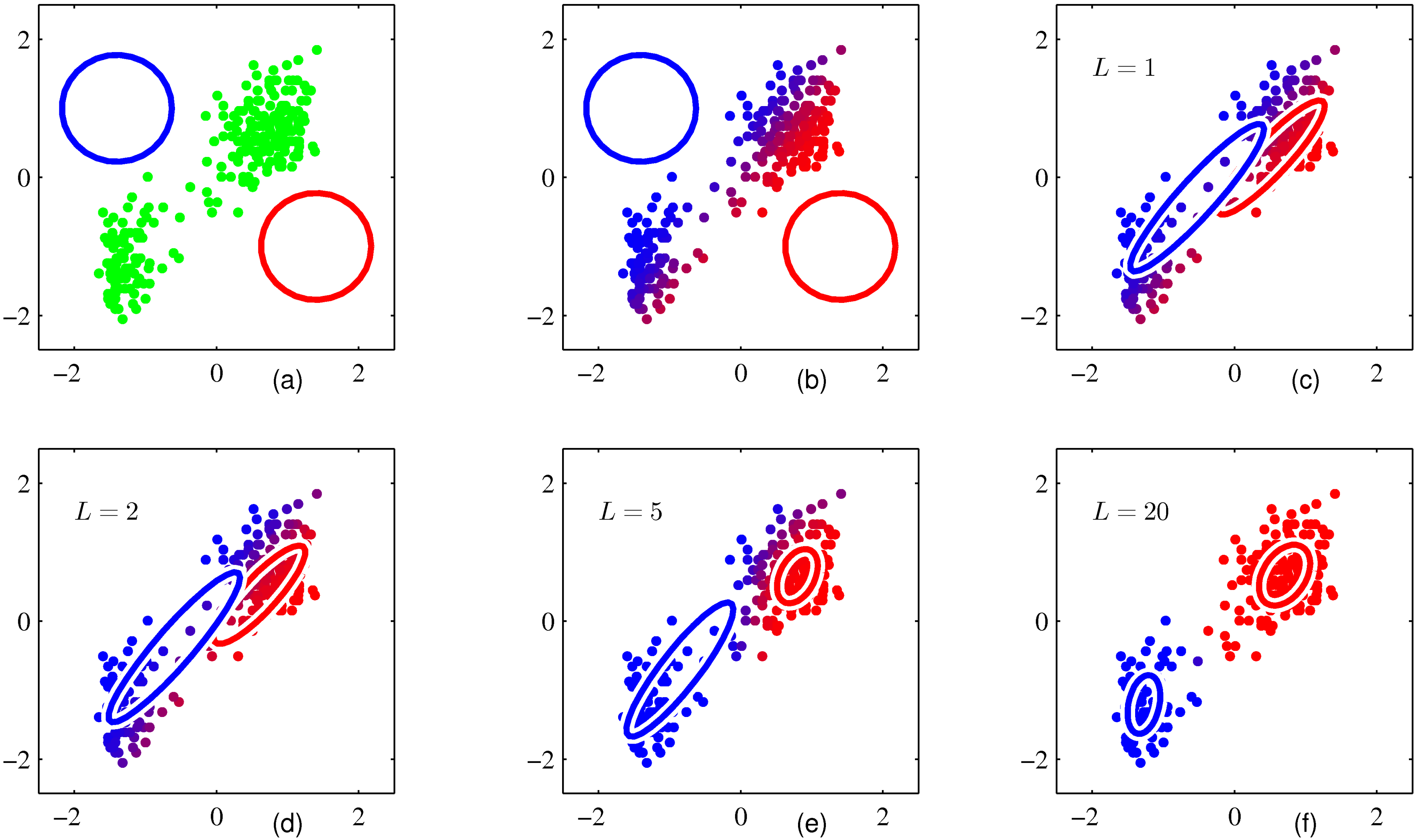
$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} q_{nk}$$

$$m_k = \frac{1}{N \pi_k} \sum_{n=1}^{N} q_{nk} x_n$$

$$C_k = \frac{1}{N \pi_k} \sum_{n=1}^{N} q_{nk} (x_n - m_k)(x_n - m_k)^T$$

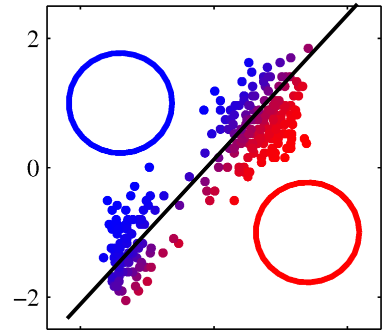
Maximum likelihood estimation of MoG

- Example of several EM iterations



Clustering with k-means and MoG

- Assignment:
  - K-means: hard assignment, discontinuity at cluster border
  - MoG: soft assignment, 50/50 assignment at midpoint



(b)

- Cluster representation
  - K-means: center only
  - MoG: center, covariance matrix, mixing weight

- If all covariance matrices are constrained to be $C_k = \epsilon I$ and $\epsilon \rightarrow 0$ then EM algorithm = k-means algorithm

- For both k-means and MoG clustering
  - Number of clusters needs to be fixed in advance
  - Results depend on initialization, no optimal learning algorithms
  - Can be generalized to other types of distances or densities

Further reading

- For more details on k-means and mixture of Gaussian learning with EM see the following book chapter (recommended !)

- Pattern Recognition and Machine Learning, chapter 9

  Chris Bishop, 2006, Springer