Clustering with k-means and Gaussian mixture distributions

Machine Learning and Object Recognition 2017-2018

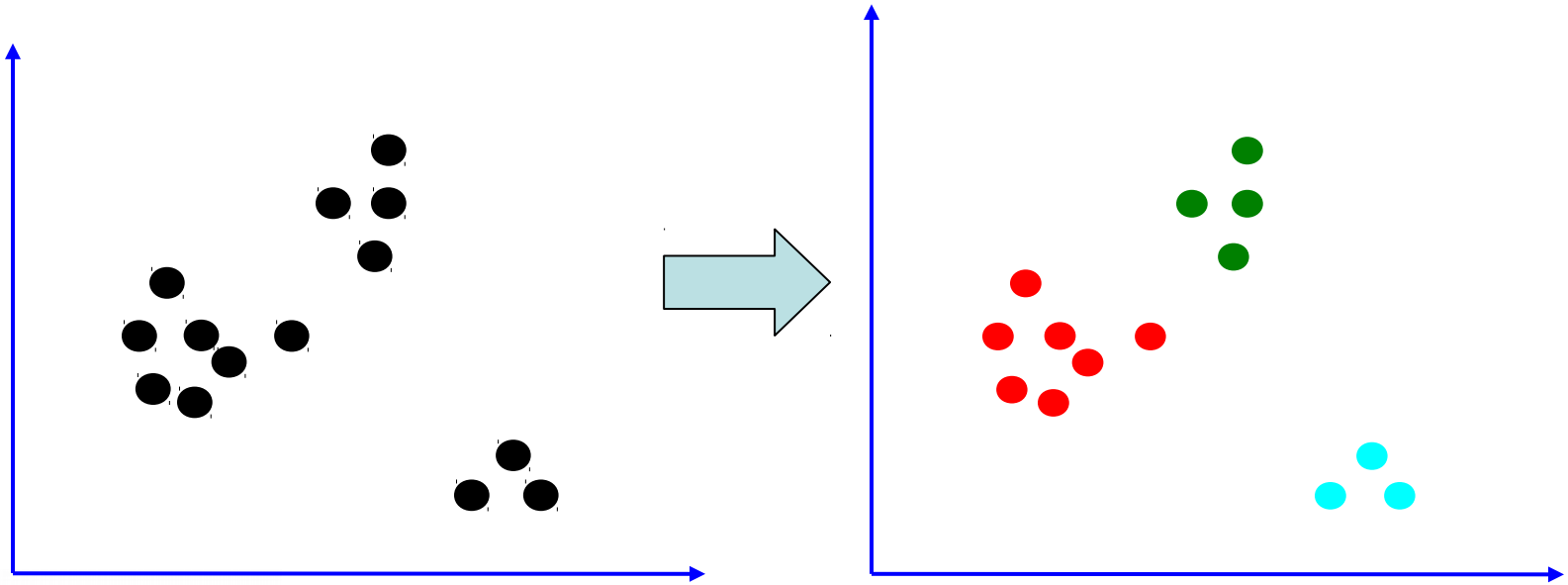
Jakob Verbeek

informatics mathematics

Clustering

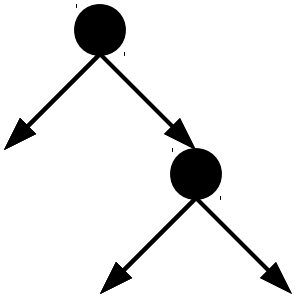
- Finding a group structure in the data
  - Data in one cluster similar to each other
  - Data in different clusters dissimilar

- Maps each data point to a discrete cluster index in {1, ... , K}
  - "Flat" methods: do not suppose any structure among the clusters
  - "Hierarchical" methods: tree structure of groupings



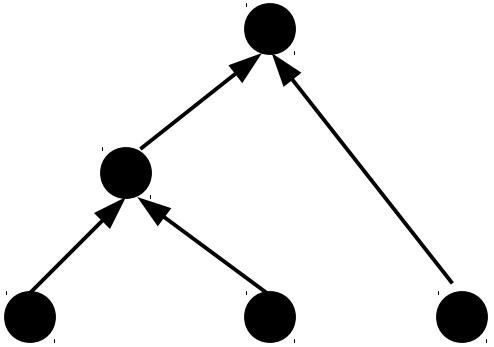
Hierarchical Clustering

- Data set is organized into a tree structure
  - Various level of granularity can be obtained by cutting-off the tree



- 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 nearest clusters
  - Distance between clusters A and B
    - E.g. min, max, or mean distance between elements in A and B

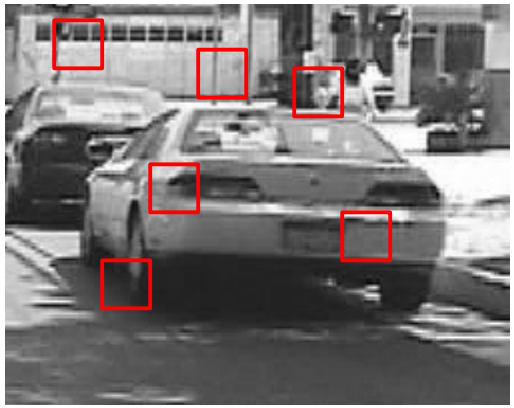


Feature clustering in Bag-Of-Words image representation

- Inspired from bag-of-word document representation
  - Vector of word counts in document

- Treat local image descriptors as the “words”

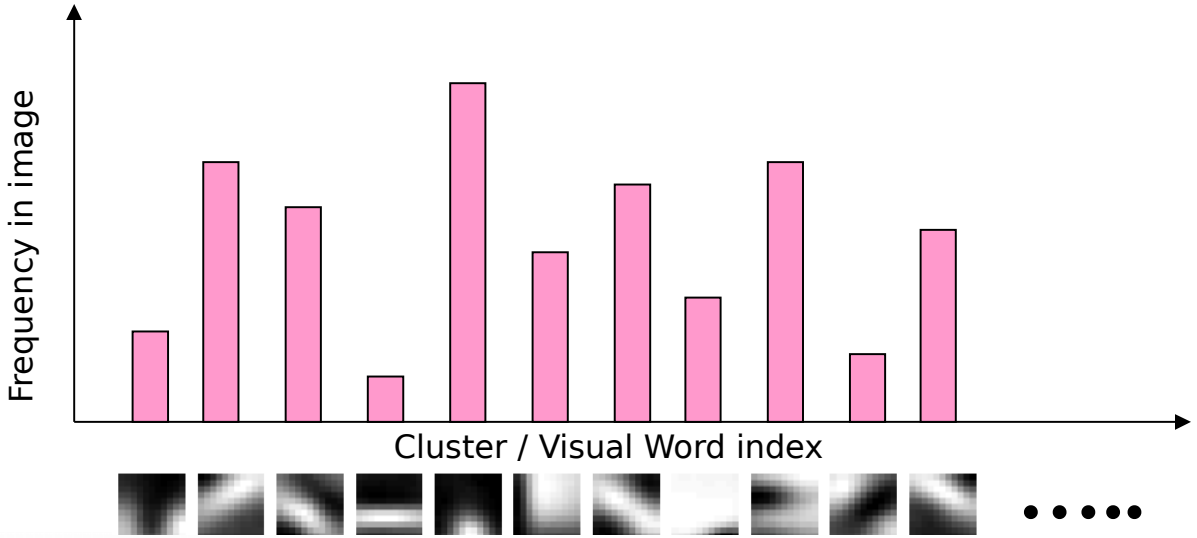
1) Sample local image patches, either using
   - Interest point detectors (most useful for retrieval)
   - Dense regular sampling grid (most useful for classification)
2) Compute descriptors of these regions
   - For example SIFT descriptors

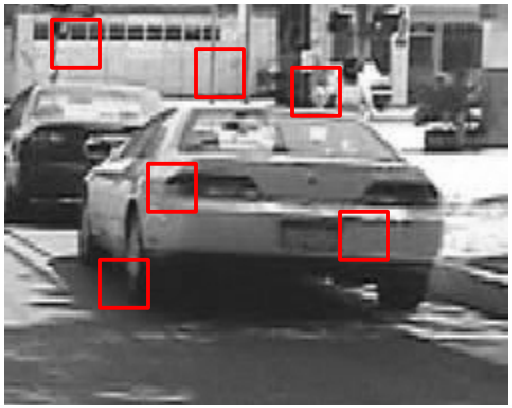


Bag-of-words image representation in a nutshell

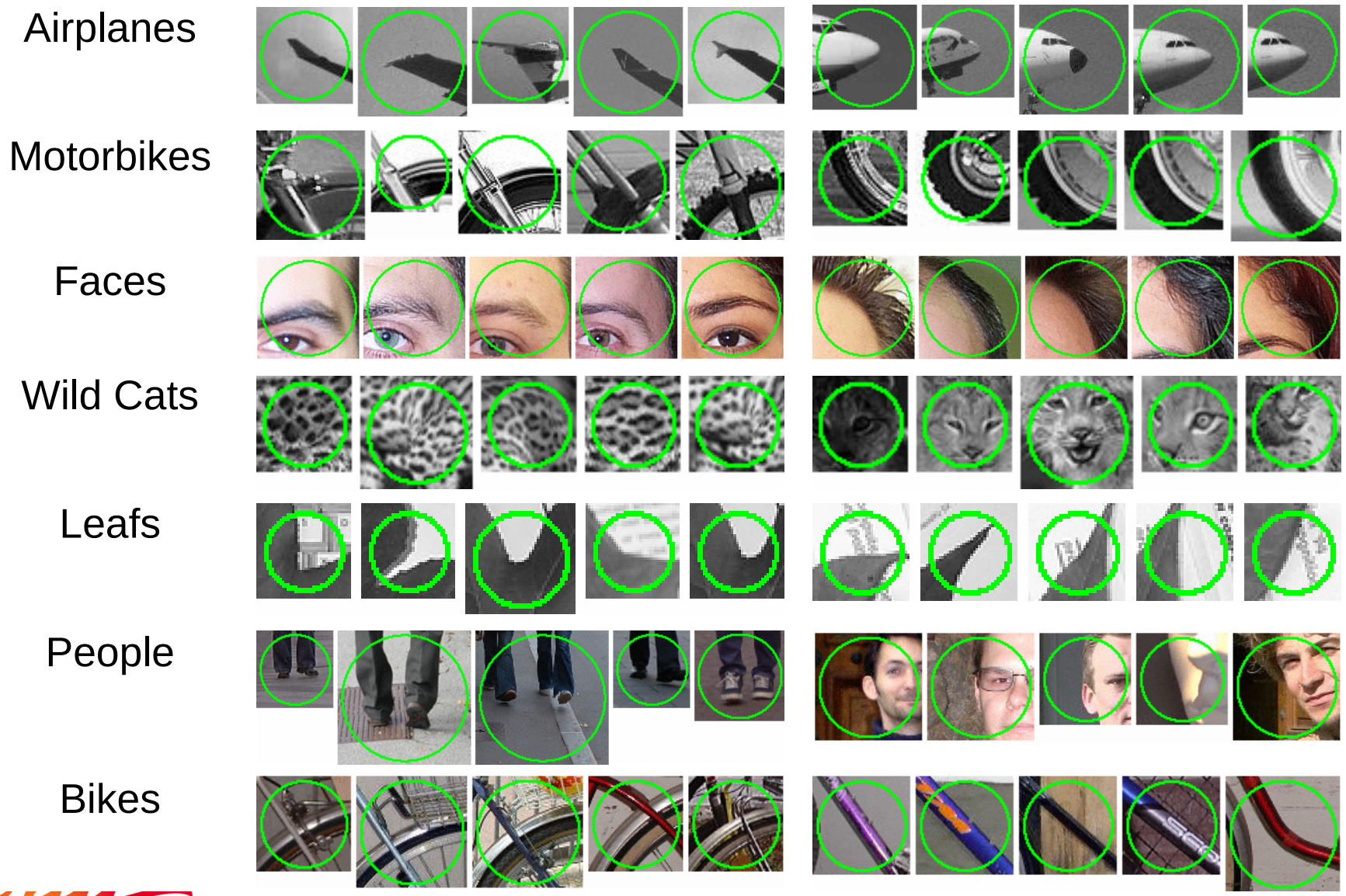
3) Aggregate the local descriptor statistics into bag-of-word histogram
   - Local descriptors are continuous, need to be discretized
   - Map each local descriptor to one of K clusters, a.k.a. “visual words”
   - Use K-dimensional histogram of visual word counts to represent image

4) Process images based on this representation
   - Classification, Retrieval, etc.



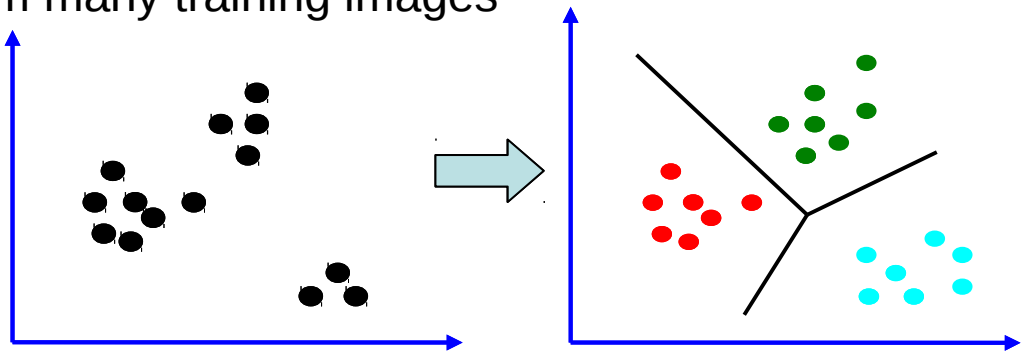


Example visual words found by k-means clustering

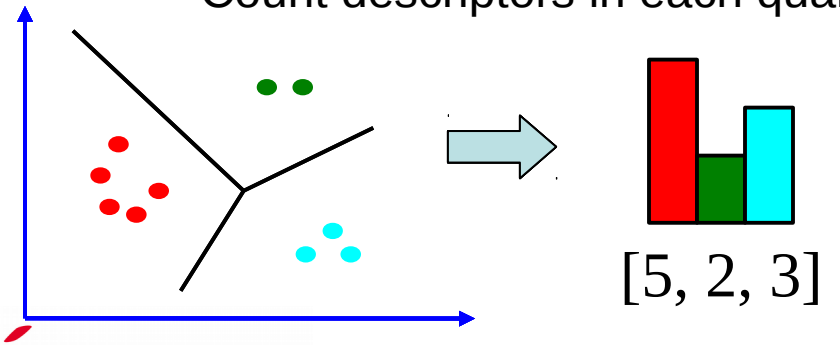


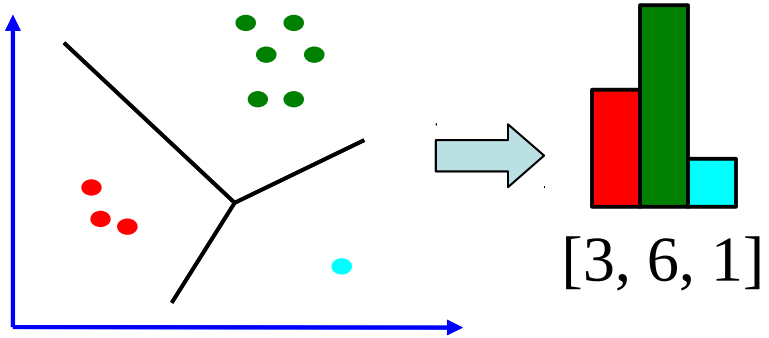
Clustering descriptors into visual words

- **Off-line clustering**: Find groups of similar local descriptors
  - Quantization of descriptor space
  - Using many descriptors from many training images



- **Encoding new images**
  - Detect local regions
  - Compute local descriptors
  - Count descriptors in each quantization cell





k-means clustering

- Given: data set of N points x_n, n=1,…,N
- Goal: find K **cluster centers** m_k, k=1,…,K

  that **minimize the squared distance to nearest cluster centers**

$$E\left(\{m_k\}_{k=1}^{K}\right)=\sum_{n=1}^{N} min_{k\in\{1,\dots,K\}}\|x_n-m_k\|^2$$

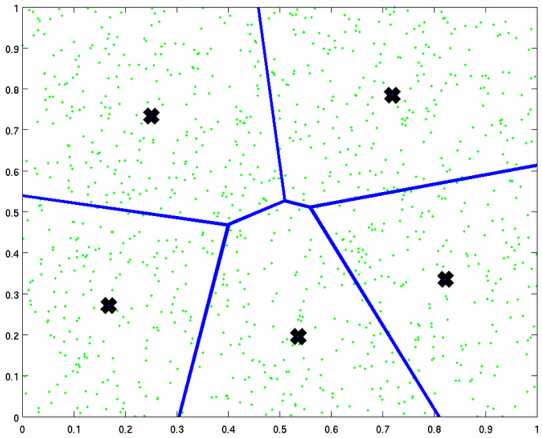
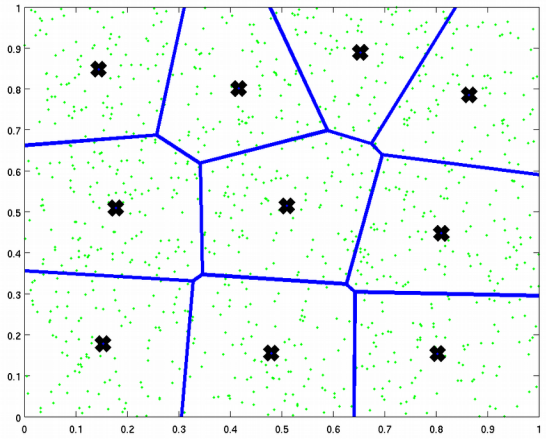
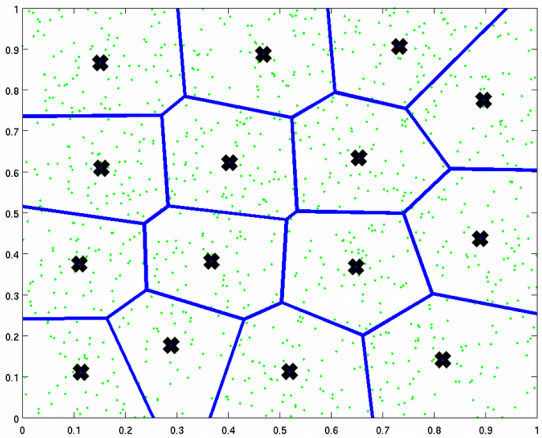
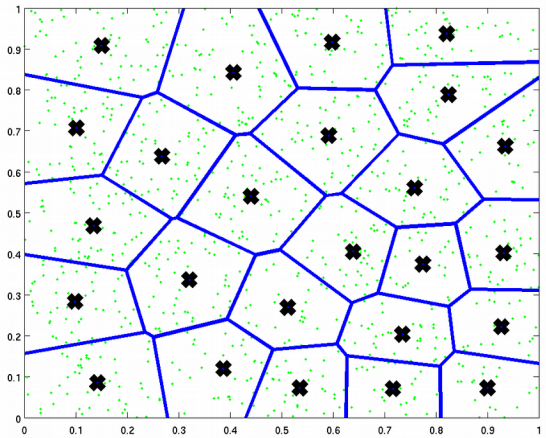
- **Clustering = assignment** of data points cluster centers
  - Indicator variables r_{nk}=1 if x_n assgined to m_k, r_{nk}=0 otherwise

- Error criterion equals sum of squared distances between each data point and assigned cluster center, **if assigned to the nearest cluster**

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

Examples of k-means clustering

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

Minimizing the error function

- Goal find centers m_k to minimize the error function

$$E\left(\{m_k\}_{k=1}^{K}\right)=\sum_{n=1}^{N} min_{k\in\{1,\ldots,K\}}\|x_n-m_k\|^2$$

- **Any set of assignments**, not just assignment to closest centers, **gives an upper-bound on the error:**

$$E\left(\{m_k\}_{k=1}^{K}\right)\leq F\left(\{m_k\},\{r_{nk}\}\right)=\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|x_n-m_k\|^2$$

- The **k-means algorithm** iteratively minimizes this bound
  1) Initialize cluster centers, eg. on randomly selected data points
  2) **Update assignments** r_{nk} for fixed centers 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

Minimizing the error bound

$$F(\{m_k\},\{r_{nk}\}) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - m_k\|^2$$

- **Update assignments** r_{nk} for fixed centers m_k

  $$\sum_k r_{nk} \|x_n - m_k\|^2$$

  - Constraint: exactly one r_{nk}=1, rest zero
  - Decouples over the data points
  - Solution: assign to closest center

Minimizing the error bound

$$F(\{m_k\},\{r_{nk}\}) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - m_k\|^2$$

- **Update centers** m_k for fixed assignments r_{nk}

$$\sum_n r_{nk} \|x_n - m_k\|^2$$

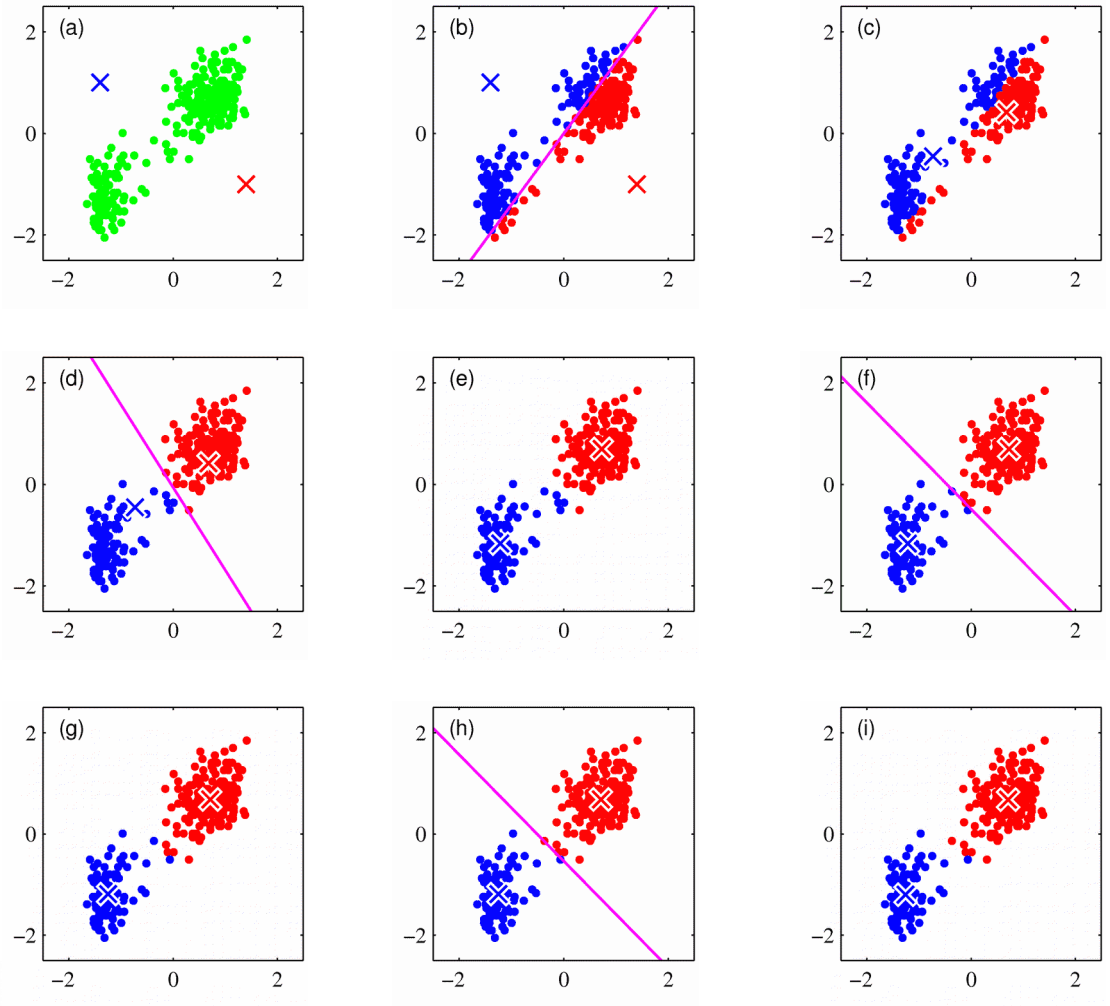
  - Decouples over the centers
  - Set derivative to zero
  - Put center at mean of assigned data points

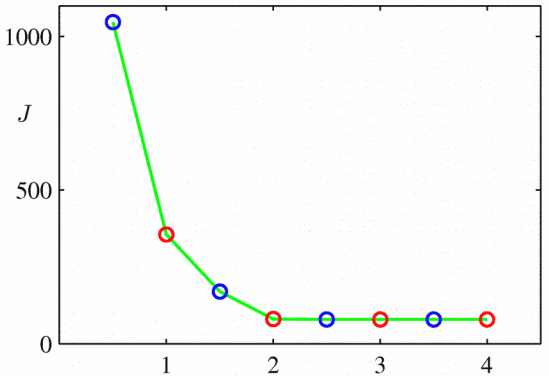
$$\frac{\partial F}{\partial m_k} = 2 \sum_n r_{nk} (x_n - m_k) = 0$$

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

Examples of k-means clustering

- Several k-means iterations with two centers





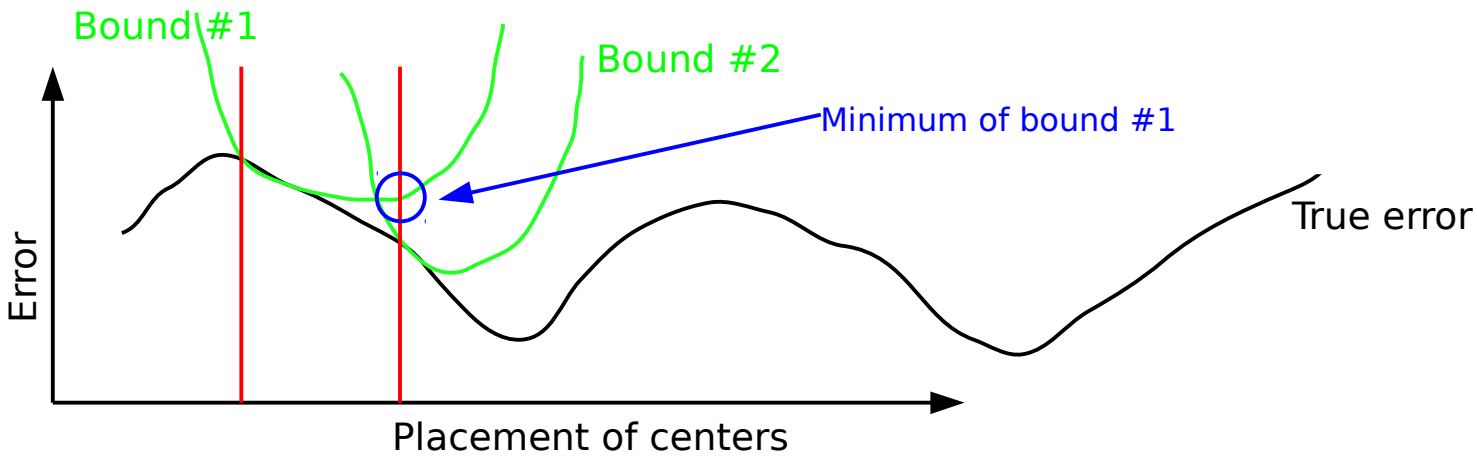
Error function

Minimizing the error function

- K-means iteratively minimizes error bound F in centers and assignments

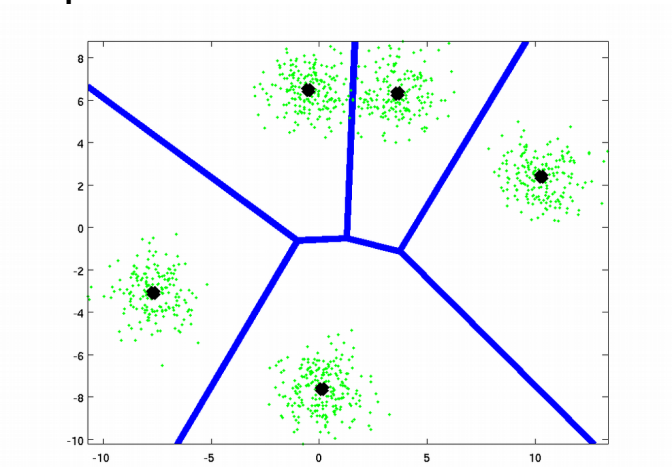
$$E\left(\{m_k\}_{k=1}^{K}\right) \leq F\left(\{m_k\},\{r_{nk}\}\right) = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\left\|x_n - m_k\right\|^2$$

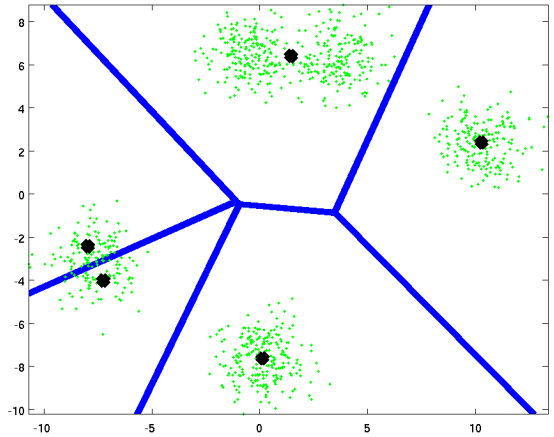
- K-means iterations monotonically decrease error function E since
  - Both steps reduce the error bound
  - Error bound matches true error after update of the assignments
  - Since finite nr. of assignments, algorithm converges to local minimum

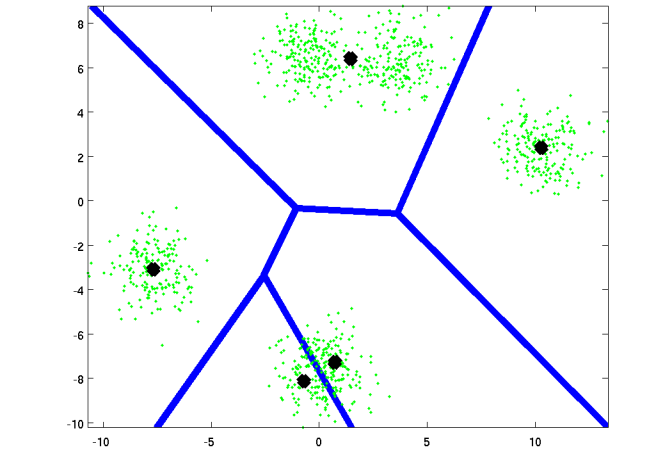


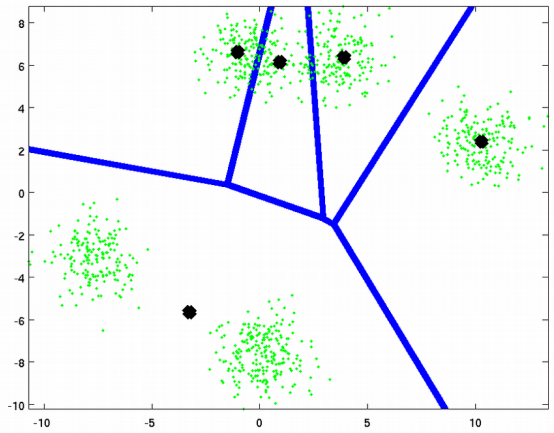
Limitations k-means clustering

- Results depend on initialization
  - Run with different initializations
  - Keep result with lowest error



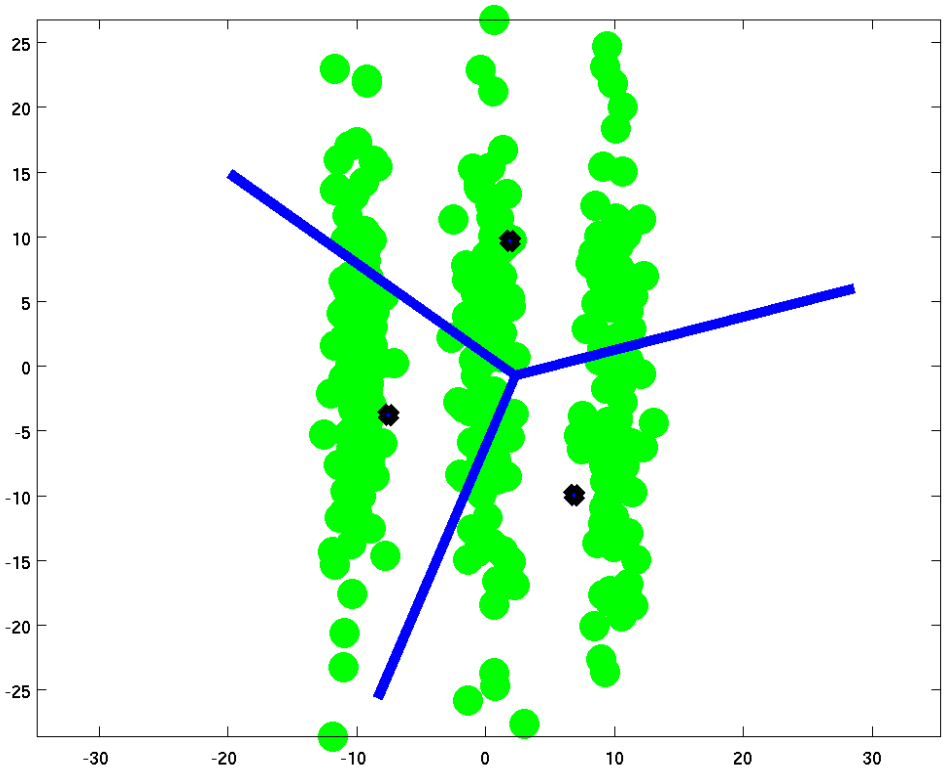






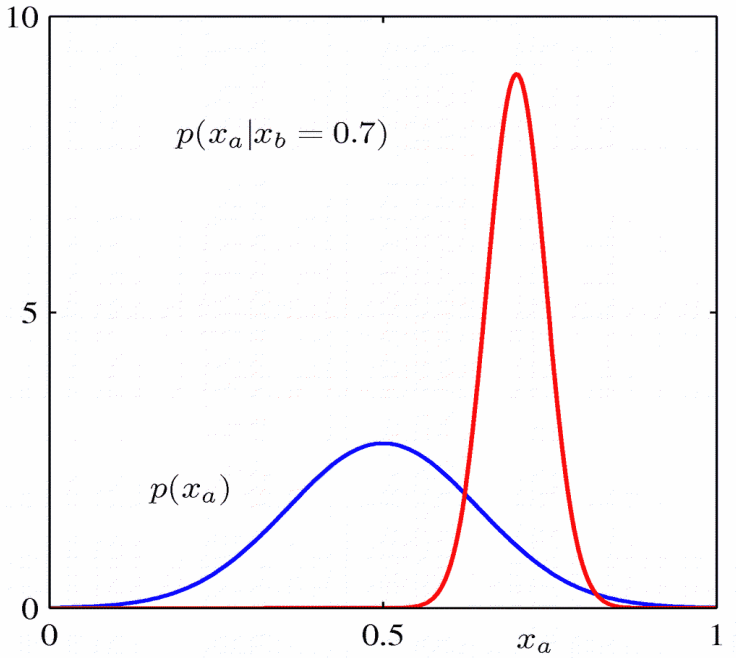
Limitations k-means clustering

- Assignment of data to clusters is only based on the distance to center
  - **No representation of the shape** of the cluster
  - Euclidean distance implicitly assumes spherical cluster shape

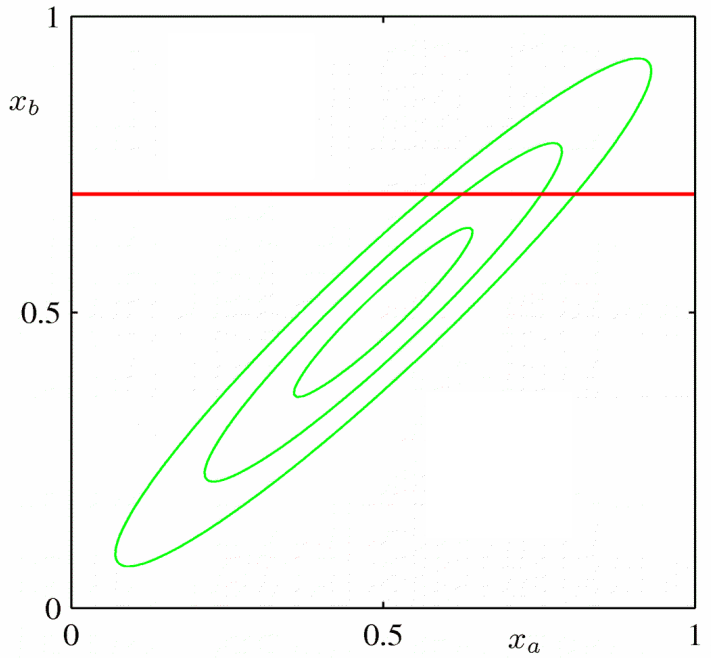


Clustering with Gaussian mixture density

- Each cluster represented by Gaussian density
  - Parameters: center m, covariance matrix C
  - Covariance matrix encodes spread around center,
    can be interpreted as defining a non-isotropic distance around center



Two Gaussians in 1 dimension



A Gaussian in 2 dimensions

Clustering with Gaussian mixture density

- Each cluster represented by Gaussian density
  - Parameters: center m, covariance matrix C
  - Covariance matrix encodes spread around center,
    can be interpreted as defining a non-isotropic distance around center

- Definition of Gaussian density in d dimensions

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

Determinant of covariance matrix C

Quadratic function of point x and mean m Mahanalobis distance

Mixture of Gaussian (MoG) density

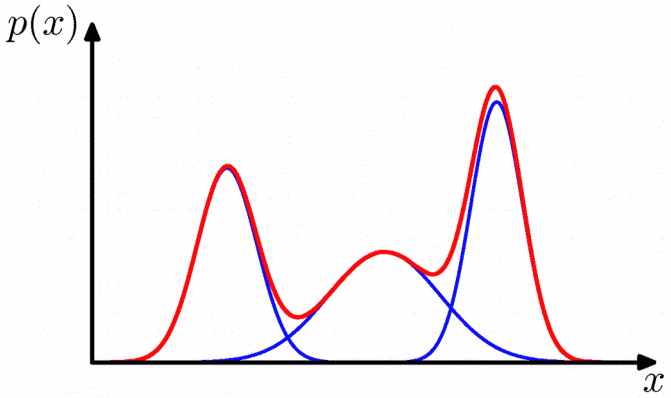
- Mixture density is weighted sum of Gaussian densities
  - Mixing weight: importance of each cluster

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

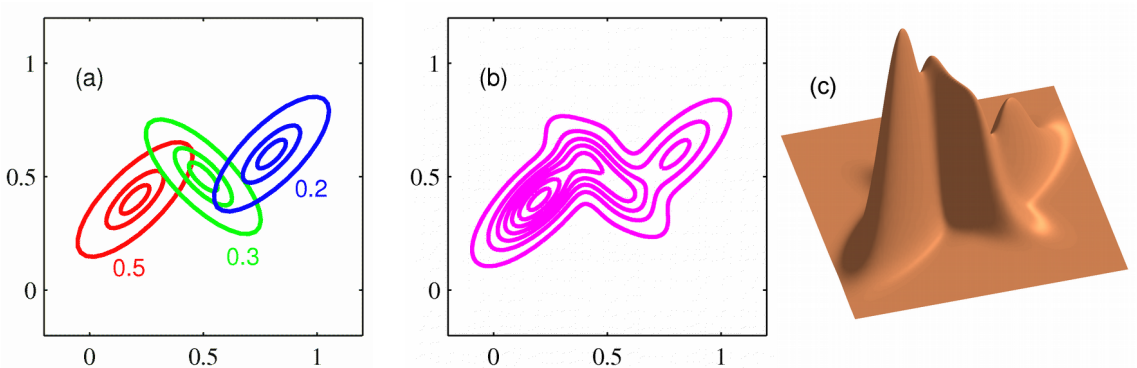
- Density has to integrate to 1, so we require

$$\pi_k \geq 0$$

$$\sum_{k=1}^{K} \pi_k = 1$$



Mixture in 1 dimension



Mixture in 2 dimensions

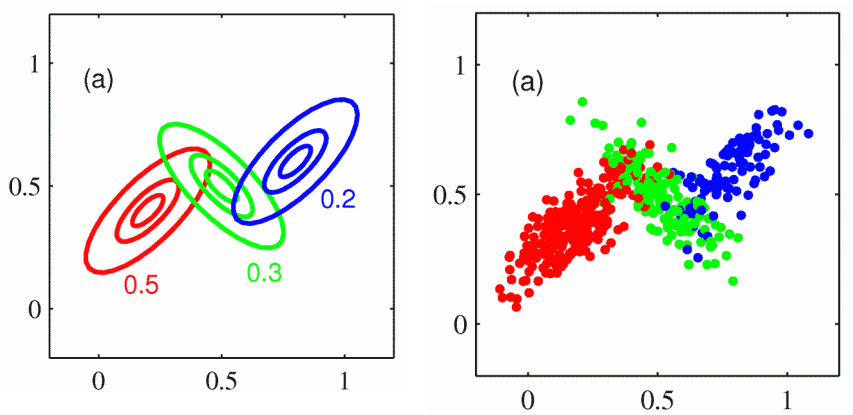
What is wrong with this picture ?!

Sampling data from a MoG distribution

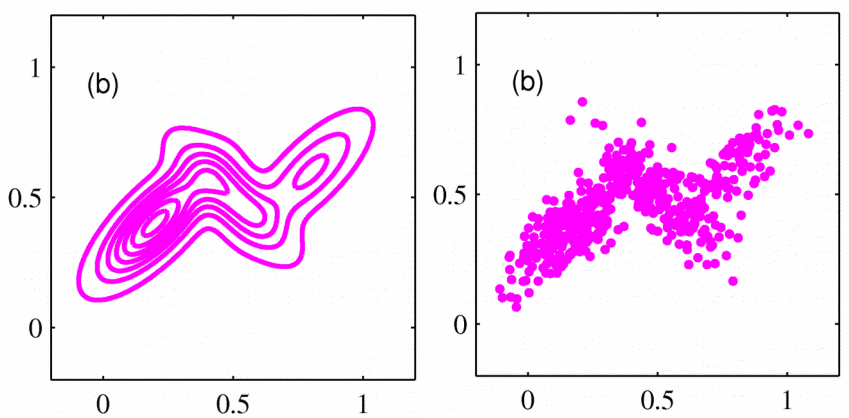
- Let z indicate cluster index
- Sample both z and x from joint distribution
  - Select z=k with probability given by mixing weight $p(z=k)=\pi_k$
  - Sample x from the k-th Gaussian $p(x|z=k)=N(x|m_k,C_k)$
- MoG recovered if we marginalize over the unknown cluster index

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

Color coded model and data of each cluster



Mixture model and data from it

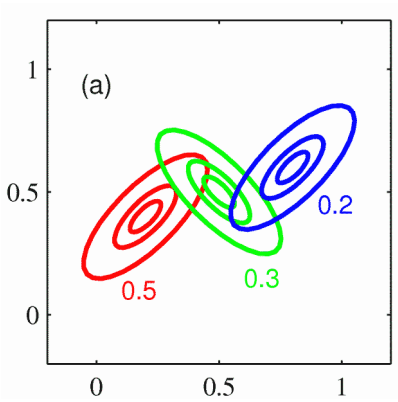


Soft assignment of data points to clusters

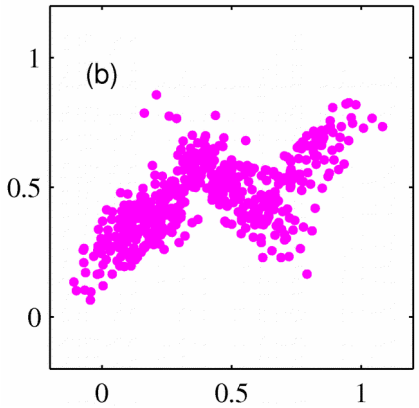
- Given data point x, infer latent underlying cluster index z

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

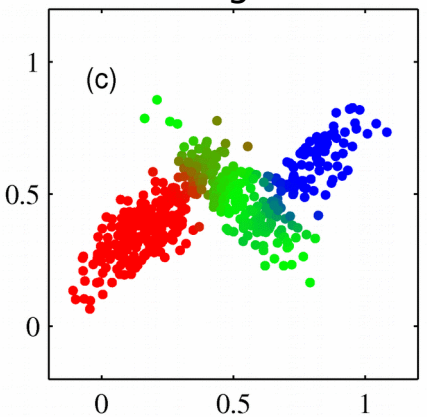
MoG model



Data



Color-coded soft-assignments



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 data log-likelihood w.r.t. parameters of MoG
  - Assuming data drawn independently from MoG

$$L(\theta)=\sum_{n=1}^{N}\log p(x_n;\theta)$$

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

- EM algorithm to learn MoG similar to k-means
  - Also an iterative algorithm to find parameters
  - Also sensitive to initialization of parameters

Maximum likelihood estimation of single Gaussian

- Given data points x_n, n=1,…,N
- Maximize 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 closed form equations for MoG
- Use **EM algorithm**
  - Initialize MoG parameters
  - E-step: soft assign of data points to clusters
  - M-step: update the mixture parameters
  - Repeat EM steps, terminate if converged
    - Convergence: of parameters, assignments, log-likelihood

- E-step: compute **soft-assignments**: $q_{nk} = p(z = k | x_n)$
- M-step: **update Gaussians** from weighted data points

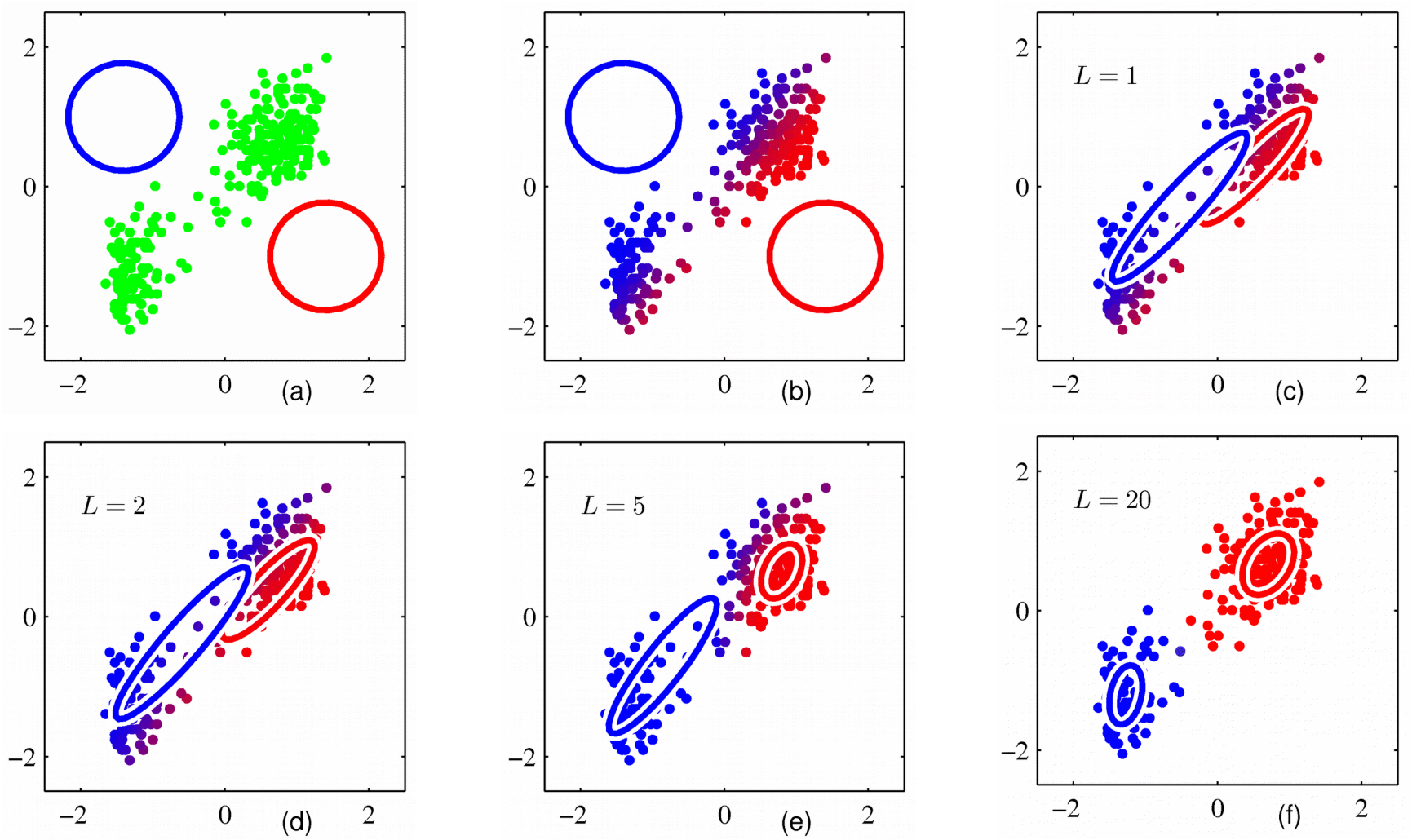
$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} q_{nk}$$

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

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

Maximum likelihood estimation of MoG

- Example of several EM iterations



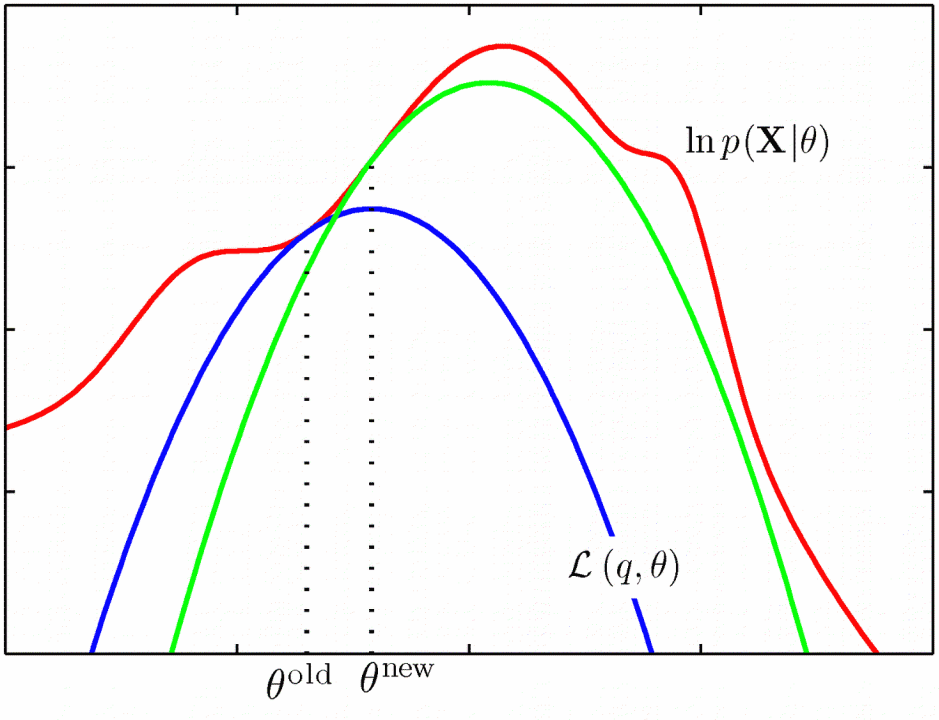
EM algorithm as iterative bound optimization

- EM algorithm is an iterative bound optimization algorithm, like k-means
  - Goal: Maximize data log-likelihood, can not be done in closed form

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

  - Solution: iteratively maximize (easier) bound on the log-likelihood

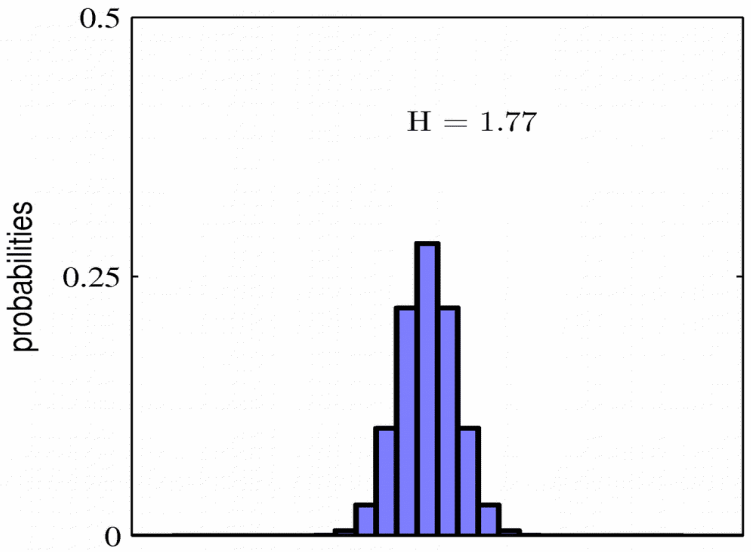
- Bound uses two information theoretic quantities
  - Entropy
  - Kullback-Leibler divergence



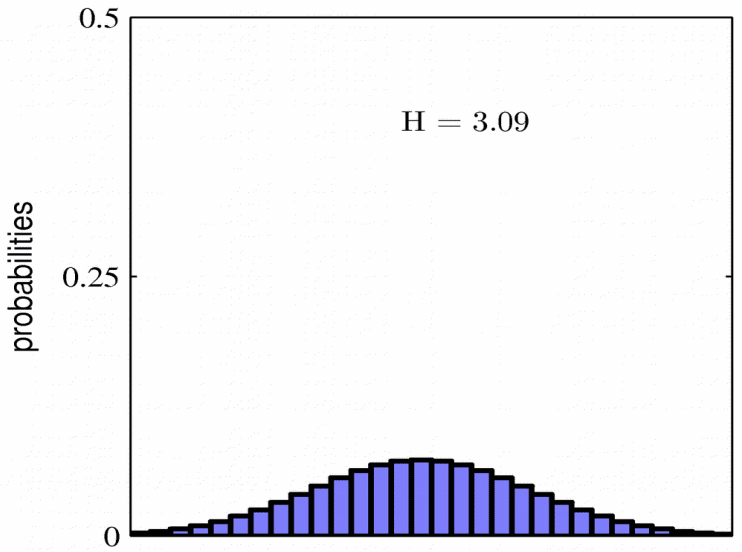
Entropy of a distribution

- Entropy captures uncertainty in a distribution
  - Maximum for uniform distribution
  - Minimum, zero, for delta peak on single value

$$H(q) = -\sum_{k=1}^{K} q(z=k) \log q(z=k)$$



Low entropy distribution



High entropy distribution

Entropy of a distribution

$$H(q) = -\sum_{k=1}^{K} q(z=k) \log q(z=k)$$

- Connection to information coding (Noiseless coding theorem, Shannon 1948)
  - Frequent messages short code, rare messages long code
  - optimal code length is (at least) -log q bits
  - Entropy: expected (optimal) code length per message

- Suppose uniform distribution over 8 outcomes: 3 bit code words
- Suppose distribution: 1/2,1/4, 1/8, 1/16, 1/64, 1/64, 1/64, 1/64, entropy 2 bits!
  - Code words: 0, 10, 110, 1110, 111100, 111101,111110,111111
- Codewords are “self-delimiting”:
  - Do not need a “space” symbol to separate codewords in a string
  - If first zero is encountered after 4 symbols or less, then stop. Otherwise, code is of length 6.

Kullback-Leibler divergence

- Asymmetric dissimilarity between distributions
  - Minimum: zero, if distributions are equal
  - Maximum: infinity, if p has a zero in support of q

$$D(q\|p)=\sum_{k=1}^{K} q(z=k)\log\frac{q(z=k)}{p(z=k)}$$

- Interpretation in coding theory
  - Sub-optimality when messages distributed according to q, but using codeword lengths derived from p
  - Difference of expected code lengths

Cross-entropy

$$D(q\|p)=-\sum_{k=1}^{K} q(z=k)\log p(z=k)-H(q)\geq 0$$

  - Suppose distribution q: 1/2,1/4, 1/8, 1/16, 1/64, 1/64, 1/64, 1/64
  - Coding with p: uniform over the 8 outcomes
  - Cross-entropy: 3 bits
  - Optimal expected code length, entropy H(q) = 2 bits
  - KL divergence D(q|p) = 1 bit

EM bound on MoG log-likelihood

- We want to bound the log-likelihood of a Gaussian mixture

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

- Bound log-likelihood by subtracting KL divergence D(q(z) || p(z|x))
  - Inequality follows immediately from non-negativity of KL

$$F(\theta, q) = \log p(x;\theta) - D\big(q(z) \| p(z|x,\theta)\big) \leq \log p(x;\theta)$$

  - p(z|x) true posterior distribution on cluster assignment
  - q(z) an **arbitrary** distribution over cluster assignment (similar to assignments used in k-means algorithm)

- Sum per-datapoint bounds to bound the log-likelihood of a data set:

$$F(\theta, \{q_n\}) = \sum_{n=1}^{N} \log p(x_n;\theta) - D\big(q_n(z) \| p(z|x_n,\theta)\big) \leq \sum_{n=1}^{N} \log p(x_n;\theta)$$

Maximizing the EM bound on log-likelihood

- **E-step:**
  - fix model parameters,
  - update distributions q_n to maximize the bound

$$F(\theta, \{q_n\}) = \sum_{n=1}^{N} \left[\log p(x_n) - D(q_n(z_n) \| p(z_n|x_n)) \right]$$

  - KL divergence zero if distributions are equal
  - Thus set $q_n(z_n) = p(z_n|x_n)$
  - **After updating the q_n the bound equals the true log-likelihood**

Maximizing the EM bound on log-likelihood

- M-step:
  - fix the soft-assignments q_n,
  - update model parameters

$$F(\theta, \{q_n\}) = \sum_{n=1}^{N} \left[\log p(x_n) - D(q_n(z_n) \| p(z_n|x_n)) \right]$$

$$= \sum_{n=1}^{N} \left[\log p(x_n) - \sum_k q_{nk} \left(\log q_{nk} - \log p(z_n = k|x_n) \right) \right]$$

$$= \sum_{n=1}^{N} \left[H(q_n) + \sum_k q_{nk} \log p(z_n = k, x_n) \right]$$

$$= \sum_{n=1}^{N} \left[H(q_n) + \sum_k q_{nk} \left(\log \pi_k + \log N(x_n; m_k, C_k) \right) \right]$$

$$= \sum_{k=1}^{K} \sum_{n=1}^{N} q_{nk} \left(\log \pi_k + \log N(x_n; m_k, C_k) \right) + \sum_{n=1}^{N} H(q_n)$$

- Terms for each Gaussian decoupled from rest

Maximizing the EM bound on log-likelihood

- Derive the optimal values for the mixing weights
  - Maximize $\sum_{n=1}^{N}\sum_{k=1}^{K} q_{nk}\log\pi_k$
  - Take into account that weights sum to one, define $\pi_1 = 1-\sum_{k=2}^{K}\pi_k$
  - Set derivative for mixing weight j >1 to zero

$$\frac{\partial}{\partial\pi_j}\sum_{n=1}^{N}\sum_{k=1}^{K} q_{nk}\log\pi_k = \frac{\sum_{n=1}^{N} q_{nj}}{\pi_j} - \frac{\sum_{n=1}^{N} q_{n1}}{\pi_1} = 0$$

$$\frac{\sum_{n=1}^{N} q_{nj}}{\pi_j} = \frac{\sum_{n=1}^{N} q_{n1}}{\pi_1}$$

$$\pi_1\sum_{n=1}^{N} q_{nj} = \pi_j\sum_{n=1}^{N} q_{n1}$$

$$\pi_1\sum_{n=1}^{N}\sum_{j=2}^{K} q_{nj} = \sum_{j=2}^{K}\pi_j\sum_{n} q_{n1}$$

$$\pi_1 N = \sum_{n=1}^{N} q_{n1}$$

$$\pi_j = \frac{1}{N}\sum_{n=1}^{N} q_{nj}$$

Maximizing the EM bound on log-likelihood

- Derive the optimal values for the MoG parameters
  - For each Gaussian maximize $\sum_n q_{nk} \log N(x_n; m_k, C_k)$
  - Compute gradients and set to zero to find optimal parameters

$$\log N(x; m, C) = \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|C| - \frac{1}{2}(x_n - m)^T C^{-1}(x_n - m)$$

$$\frac{\partial}{\partial m}\log N(x; m, C) = C^{-1}(x - m)$$

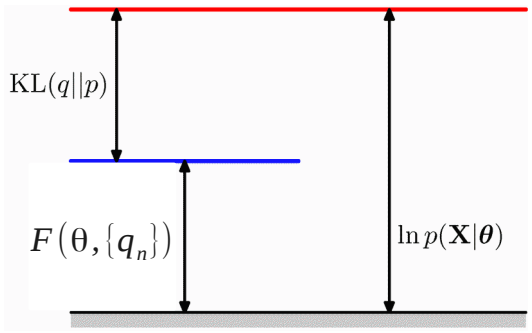
$$m_k = \frac{\sum_n q_{nk} x_n}{\sum_n q_{nk}}$$

$$\frac{\partial}{\partial C^{-1}}\log N(x; m, C) = \frac{1}{2}C - \frac{1}{2}(x - m)(x - m)^T$$

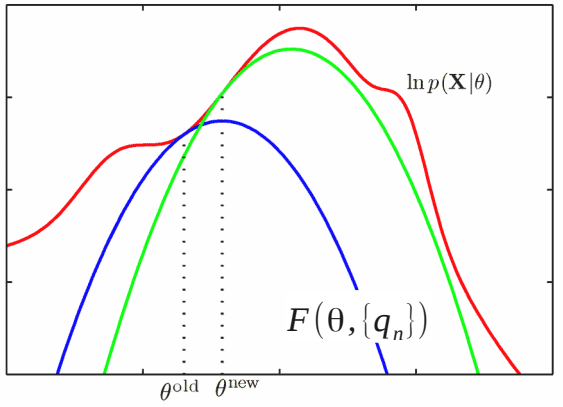
$$C_k = \frac{\sum_n q_{nk}(x_n - m)(x_n - m)^T}{\sum_n q_{nk}}$$

EM bound on log-likelihood

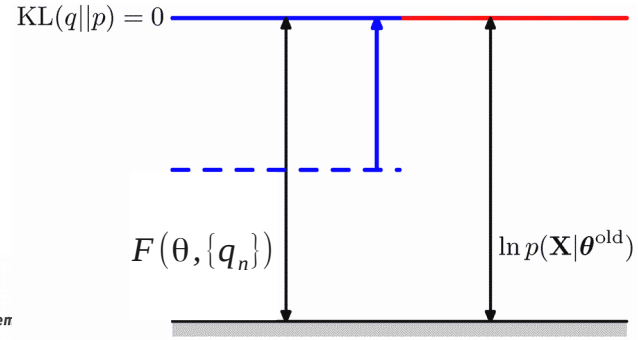
- L is bound on data log-likelihood for auxiliary distribution q

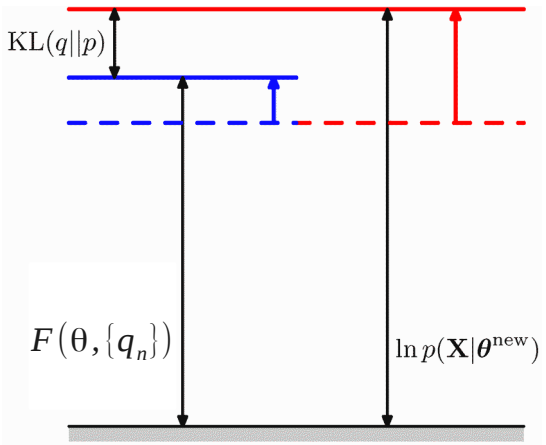


$$F(\theta,\{q_n\}) = \sum_{n=1}^{N}\left[\log p(x_n) - D(q_n(z_n)\|p(z_n|x_n))\right]$$



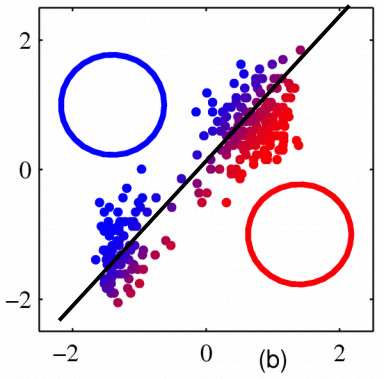
- Iterative coordinate ascent on F
  - E-step optimize q, makes bound tight
  - M-step optimize parameters





Clustering with k-means and MoG

- Assignment:
  - K-means: hard assignment, discontinuity at cluster border
  - MoG: soft assignment according to posterior p(z|x)



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

- If mixing weights are equal and
  all covariance matrices are constrained to be $C_k = \epsilon I$ and $\epsilon \to 0$
  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

Reading material

- Questions to expect on exam:
  - Describe objective function for one of these methods
  - Derive some of the update equations for the model parameters
  - Derive k-means as special case of MoG clustering

- More details on k-means and mixture of Gaussian learning with EM
  - Pattern Recognition and Machine Learning,

    Chapter 9

    Chris Bishop, 2006, Springer

  - R. Neal and G. Hinton

    A view of the EM algorithm that justifies incremental, sparse, and other variants

    In “Learning in Graphical Models”, Kluwer, 1998, 355-368