02457 Signal Processing in Non-linear Systems: Lecture 8

Clustering and radial basis function networks

Ole Winther

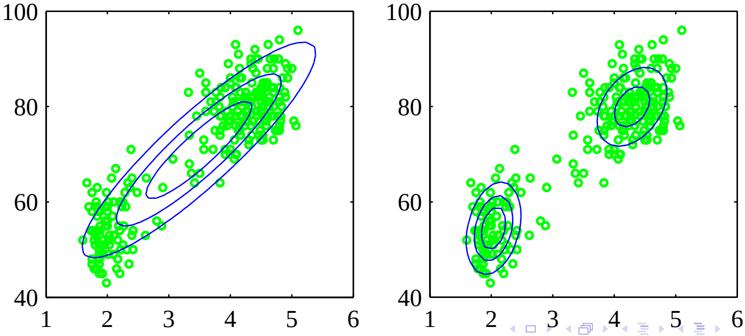
Technical University of Denmark (DTU)

October 22, 2016

- Hour 1
  - Summary lecture 7
    - Unsupervised learning task
    - Mixture modeling
    - Expectation maximization (EM) algorithm
  - Exercise 7 walk through - use at exam (one possible take)
- Hour 2
  - Your turn! Exercise 7 quiz
  - EM - now with proof!
  - Likelihood functions
  - Your turn! Clustering with K-means
- Hour 3
  - Mixture of Gaussians as clustering
  - Hierarchical clustering
  - Radial basis function (RBF) networks

Unsupervised learning

- Learning the distribution of a set of variables $p(\mathrm{input})$.
- Or perhaps just some important characteristics of the distribution

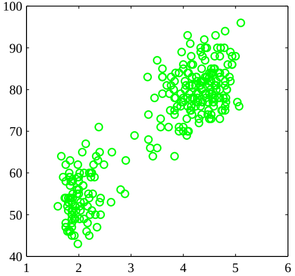


Unsupervised learning task

- Density estimation
  - Compression, creating compact representation of data
  - Generative modeling $P(\mathcal{C}_k|\mathbf{x}) \propto p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)$
  - Outlier detection, identification
  - In this course only continuous densities: Gaussian, mixture of Gaussians and non-parametric (histogram and kernel densities)
- Clustering
  - unsupervised classification
  - prototypical summary
- Feature extraction/visualization –
  - finding sub-space with most variance (PCA)
  - finding regions with high density (K-means).

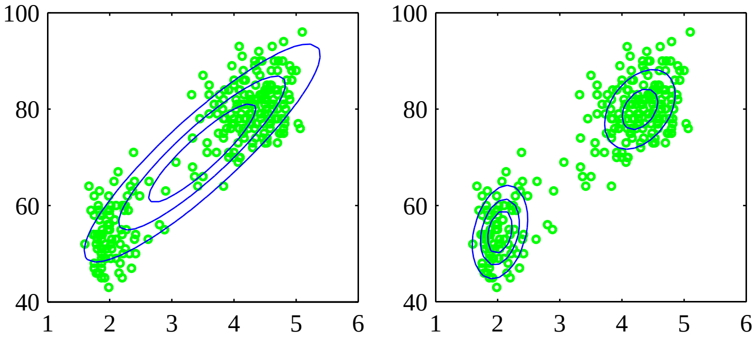
Old Faithful

- Hydrothermal geyser in Yellowstone National Park, Wyoming, USA.



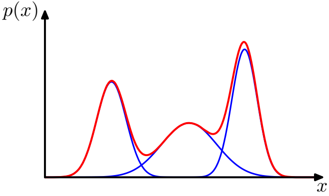
- x-axis duration of eruption in minutes
- y-axis time to next eruption in minutes

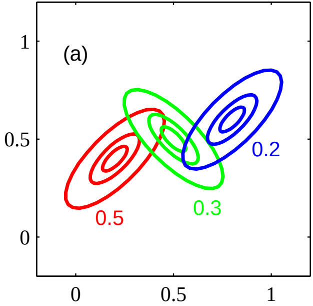
Density estimation



- Mixture modeling – convex combinations of simpler models

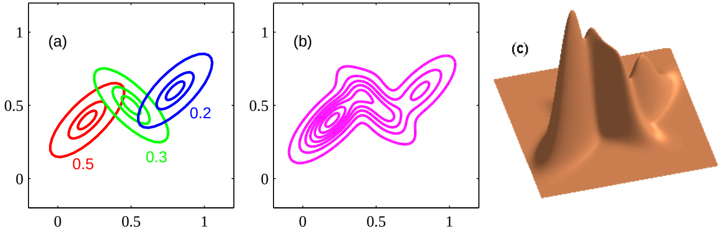
$$p(\mathbf{x}) = \sum_{k=1}^{K} p(k) p(\mathbf{x}|k) , \qquad \sum_k p(k) = 1$$





Mixture of Gaussians (MoG)

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \,, \qquad \sum_k \pi_k = 1$$



- The training set is $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_N\}$
- the likelihood function is given by

$$p(\mathbf{X}|\mathbf{w}) = \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{w})$$

- Parameters $\mathbf{w} = \{\pi_k, \boldsymbol{\mu}_k, \Sigma_k\}$
- The cost function is then (notice sum inside log)

$$\begin{aligned} E(\mathbf{w}) &= -\sum_{n=1}^{N} \log p(\mathbf{x}_n|\mathbf{w}) = -\sum_{n=1}^{N} \log \sum_{k=1}^{K} p(\mathbf{x}_n|\mathbf{w}_k)\pi_k \\ &= -\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \end{aligned}$$

Your turn! – MoG maximum likelihood

- We will try to maximize the log likelihood by setting the gradient to zero wrt to the parameters $\{\pi_k, \boldsymbol{\mu}_k, \Sigma_k\}$

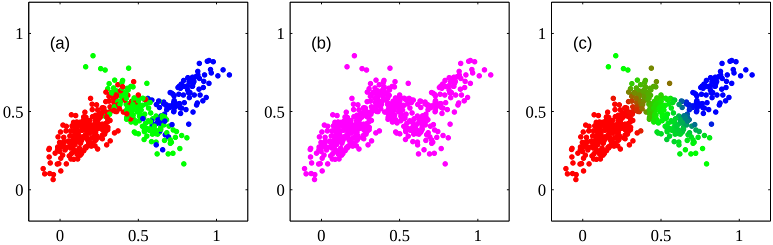
$$E(\mathbf{w}) = -\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) .$$

- Hint: we need to introduce the so-called responsibility

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{k'}, \Sigma_{k'})} \in [0, 1] .$$

Responsibility – soft assignments

$$\begin{aligned}\gamma_{nk} &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{k'}, \Sigma_{k'})} \\ &= \frac{p(k)p(\mathbf{x}_n|k)}{\sum_{k'} p(k')p(\mathbf{x}_n|k')} = p(k|\mathbf{x}_n) \in [0,1] \, .\end{aligned}$$



MoG maximum likelihood - π_k

Derivative wrt π_k of

$$\mathcal{L}(\mathbf{w}, \lambda) = E(\mathbf{w}) + \lambda \left[\sum_{k'=1}^{K} \pi_{k'} - 1 \right] .$$

Cost function and responsibility

$$\begin{aligned} E(\mathbf{w}) &= -\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \\ \gamma_{nk} &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{k'}, \Sigma_{k'})} . \end{aligned}$$

MoG maximum likelihood - $\boldsymbol{\mu}_k$

Use (see appendix C)

$$\frac{\partial}{\partial \boldsymbol{\mu}_k} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k) = \Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k)$$

Cost function and responsibility

$$\begin{aligned} E(\mathbf{w}) &= -\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \Sigma_k) \\ \gamma_{nk} &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_{k'}, \Sigma_{k'})} . \end{aligned}$$

MoG maximum likelihood - Σ_k

Use (see appendix C)

$$\frac{\partial}{\partial \Sigma_k} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k) = -\frac{1}{2}\left[\Sigma_k^{-1} - \Sigma_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}\right] \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}$$

Cost function and responsibility

$$\begin{aligned} E(\mathbf{w}) &= -\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \\ \gamma_{nk} &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{k'}, \Sigma_{k'})} . \end{aligned}$$

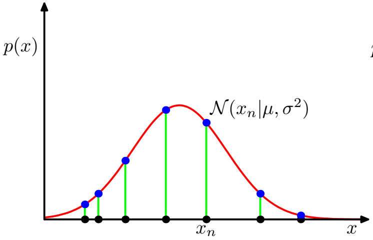
E-step - for $n = 1, \ldots, N$ and $k = 1, \ldots, K$:

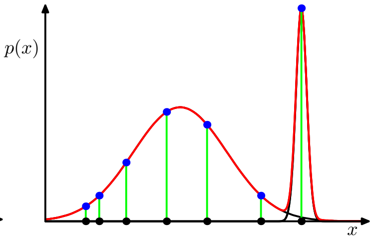
$$\gamma_{nk} \leftarrow \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_{k'}, \Sigma_{k'})} .$$

M-step - for $k = 1, \ldots, K$:

$$\begin{aligned}
N_k &\leftarrow \sum_{n=1}^{N} \gamma_{nk} \\
\pi_k &\leftarrow \frac{N_k}{N} \\
\boldsymbol{\mu}_k &\leftarrow \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \mathbf{x}_n \\
\Sigma_k &\leftarrow \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T
\end{aligned}$$

Nature of the maximum likelihood solution

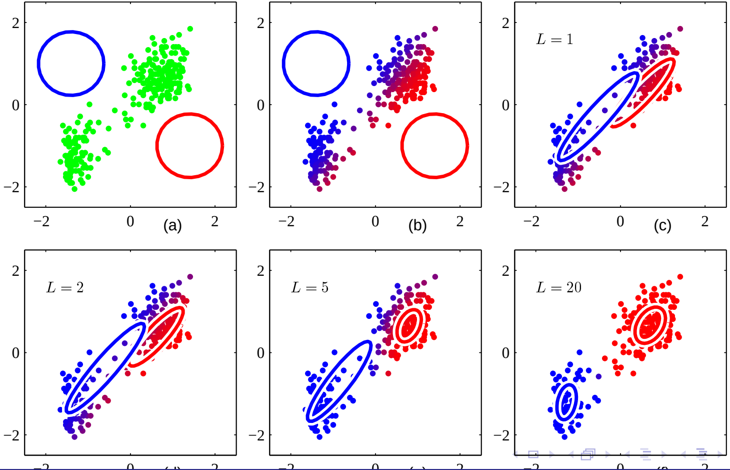




$$E(\mathbf{w}) = -\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)$$

Consider cost when $\mu_k = \mathbf{x}_n$, $\pi_k > 0$ and $\Sigma_k \to 0$.

MoG for Old Faithful



- Exercise 7 walk through - use at exam (one possible take)
- Your turn! Exercise 7 quiz

Expectation-Maximization algorithm

- A general scheme for maximum likelihood estimation.
- We are about to prove that in every step the likelihood will increase or stay constant.
- We bound the change in cost function:

$$
\begin{aligned}
E^{\text{new}}(\mathbf{w}) &= -\sum_{n=1}^{N} \log p^{\text{new}}(\mathbf{x}_n|\mathbf{w}) \\
&= -\sum_{n=1}^{N} \log \sum_{k=1}^{K} p^{\text{new}}(\mathbf{x}_n|k)\pi_k^{\text{new}} \frac{\gamma_{nk}^{\text{old}}}{\gamma_{nk}^{\text{old}}} \\
&\leq -\sum_{n=1}^{N}\sum_{k=1}^{K} \gamma_{nk}^{\text{old}} \log \frac{p^{\text{new}}(\mathbf{x}_n|k)\pi_k^{\text{new}}}{\gamma_{nk}^{\text{old}}}
\end{aligned}
$$

M-step - minimize upper bound

- Jensen's inequality

$$\log\left(\sum_j \lambda_j x_j\right) \geq \sum_j \lambda_j \log(x_j) \qquad \sum_j \lambda_j = 1$$

Minimize upper bound:

$$E^{\text{new}}(\mathbf{w}) \leq -\sum_{n=1}^{N}\sum_{k=1}^{K} \gamma_{nk}^{\text{old}} \log \frac{p^{\text{new}}(\mathbf{x}_n|k)\pi_k^{\text{new}}}{\gamma_{nk}^{\text{old}}}$$

- No log inside product anymore so easy to take derivatives and set $= 0$.

Minimizing the bound

- M-step – minimizing the bound gives, $N_k^{\text{new}} = \sum_{n=1}^N \gamma_{nk}^{\text{old}}$:

$$\pi_k^{\text{new}} = \frac{N_k^{\text{new}}}{N}$$

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k^{\text{new}}} \sum_{n=1}^{N} \gamma_{nk}^{\text{old}} \mathbf{x}_n$$

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk}^{\text{old}} (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})^T$$

- Extra material on even more general view!

- E-step - make bound tight
- The upper bound can be decomposed

$$-\sum_{n=1}^{N}\sum_{k}\gamma_{nk}^{\mathrm{old}}\log\frac{p^{\mathrm{new}}(\mathbf{x}_n|k)\pi_k^{\mathrm{new}}}{\gamma_{nk}^{\mathrm{old}}} = -\sum_{n}\log p^{\mathrm{new}}(\mathbf{x}_n) - \sum_{n}\sum_{k}\gamma_{nk}^{\mathrm{old}}\log\frac{\gamma_{nk}^{\mathrm{new}}}{\gamma_{nk}^{\mathrm{old}}}$$

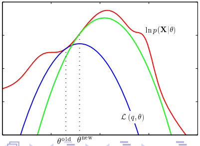
- Use definition of responsibility: $\gamma_{nk}^{\mathrm{new}} = \frac{p^{\mathrm{new}}(\mathbf{x}_n|k)\pi_k^{\mathrm{new}}}{p^{\mathrm{new}}(\mathbf{x}_n)}$

- E-step - make bound tight
- The upper bound can be decomposed

$$-\sum_{n=1}^{N}\sum_{k}\gamma_{nk}^{\mathrm{old}}\log\frac{p^{\mathrm{new}}(\mathbf{x}_n|k)\pi_k^{\mathrm{new}}}{\gamma_{nk}^{\mathrm{old}}} = -\sum_{n}\log p^{\mathrm{new}}(\mathbf{x}_n) - \sum_{n}\sum_{k}\gamma_{nk}^{\mathrm{old}}\log\frac{\gamma_{nk}^{\mathrm{new}}}{\gamma_{nk}^{\mathrm{old}}}$$

- Use definition of responsibility: $\gamma_{nk}^{\mathrm{new}} = \frac{p^{\mathrm{new}}(\mathbf{x}_n|k)\pi_k^{\mathrm{new}}}{p^{\mathrm{new}}(\mathbf{x}_n)}$

- Last term: Kullback-Leibler divergence
- $KL \geq 0$ and
- $KL = 0$ when $\gamma_{nk}^{\mathrm{old}} = \gamma_{nk}^{\mathrm{new}}$



The likelihood function

- See blackboard!

Your turn! – invent a clustering algorithm

Ingredients:

- The training set is $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_N\}$
- Choose K prototypes in D-dimensional space:

$$\boldsymbol{\mu}_k \ , \qquad k = 1, \ldots, K \ .$$

- Choose a distance measure $d(\mathbf{x}, \mathbf{x}')$, e.g. Euclidian

$$d(\mathbf{x}, \mathbf{x}') = ||\mathbf{x} - \mathbf{x}'||^2 = \sum_{i=1}^{D} (x_i - x_i')^2 \ .$$

- Hint: It is not possible to make a one-shot algorithm. It is necessary to make it iterative.

K-means clustering

- The training set is $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_N\}$
- A set of K mean vectors: $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$.
- A set of assignments (binary indicators): $r_{nk} \in \{0, 1\}$.
- $r_{nk} = 1$ if data point $\mathbf{x}_n$ is assigned to cluster k and zero otherwise.
- Objective function (distortion measure):

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \boldsymbol{\mu}_k||^2$$

Your turn! – K-means

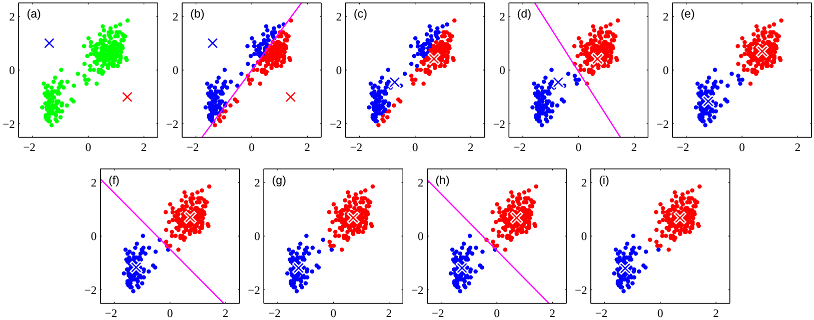
Objective function (distortion measure):

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \boldsymbol{\mu}_k||^2$$

Your turn! - make an algorithm that alternates between minimizing J with respect to

1. (E-step) r_{nk}, $n = 1, \ldots, N$ for *fixed* $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$ and
2. (M-step) $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$ for *fixed* r_{nk}, $n = 1, \ldots, N$.

K-means for Old Faithful



K-means limitations and extension

- Complexity? $\mathcal{O}(KNN_{\text{ite}})$ distance evaluations.
- Noticed anything funny with the axes? – try it yourself code included in exercise.
- What is the distance measure used?
- Should it be adaptive?
- What if the mean is meaningless? Think of discrete data.
- Solution – use data points as centers.

K-mediods

- K-medoids – code included in exercise

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} D(\mathbf{x}_n, \boldsymbol{\mu}_k) \ , \qquad \boldsymbol{\mu}_k \in \mathbf{X}$$

- $D(\mathbf{x}, \mathbf{x}')$ general dissimilarity measure.
- Additional complexity $\mathcal{O}(\sum_k N_k^2 N_{\mathrm{ite}})$
- Other possibilities like K-medians.

K-means for image segmentation

- Images are quite redundant
- Many small patches are very similar.
- In the example we treat each RGB pixel as a 3d vector.
- Cluster with k-means and transmit cluster centers (code vectors) and assignments.

Original image

Image segmentation

Original image

$K = 2$

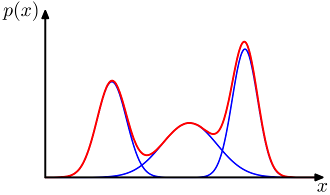
$K = 3$

$K = 10$

Lossy compression

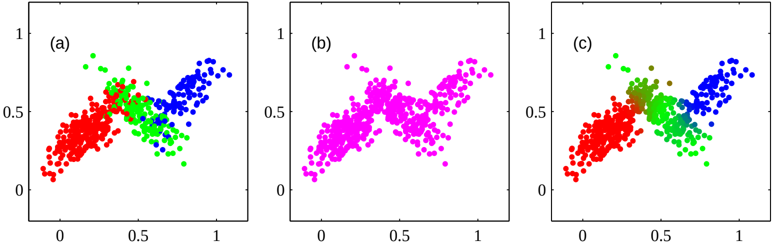
- Compression for 8 bit accuracy and N pixel image.
- Original image: $3 * 8 * N$ bits.
- Cluster means (code vectors): $3 * 8 * K$ bits.
- Assignments: $N * \log_2 K$ bits.
- Ratio, $K = 2, 3$ and 10: $4.2\%, 8.3\%$ and 16.3%.

- K-means is non-probabilistic - no likelihood.
- For example the assignments are hard!
- Propose a probabilistic model for clustering.
- Mixture modeling is the solution.

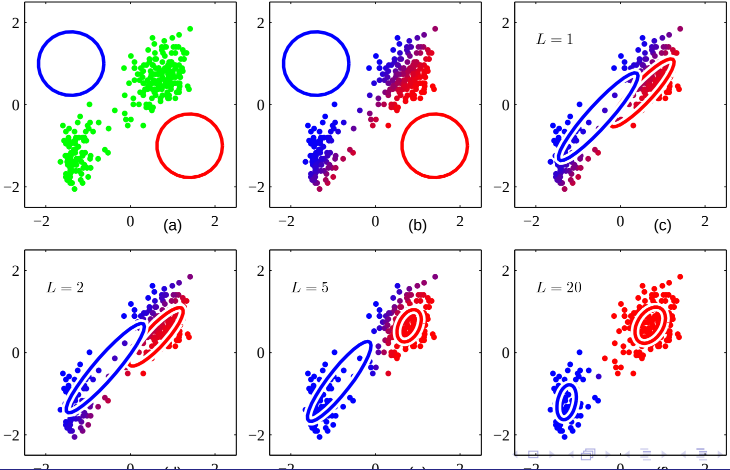


Responsibility – soft assignments

$$
\begin{aligned}
\gamma_{nk} &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{k'}, \Sigma_{k'})} \\
&= \frac{p(k)p(\mathbf{x}_n|k)}{\sum_{k'} p(k')p(\mathbf{x}_n|k')} = p(k|\mathbf{x}_n) \in [0, 1] \, .
\end{aligned}
$$

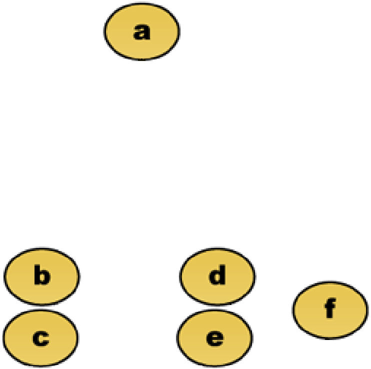


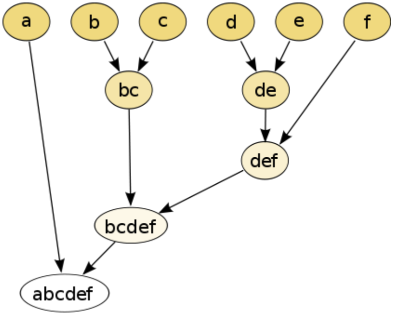
MoG for Old Faithful



Hierarchical clustering

Hierarchical clustering





Hierarchical Clustering

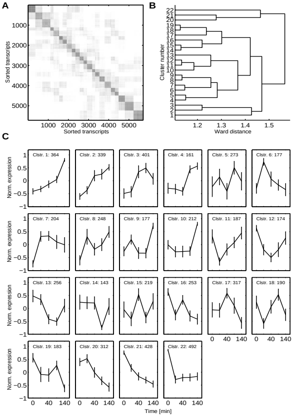
- Define a hierarchy (binary tree, dendrogram)
- Groups/samples are split on dissimilarity.
- **Between-sample dissimilarity** (or distance) $d(X, X')$. For example
$$d(X_n, X_{n'}) = ||X_n - X_{n'}||^2 \ .$$
- **Between-group dissimilarity** $d(G, H)$. For example so-called single linkage
$$d(G, H) = d_{\mathrm{SL}} = \min_{n \in G, n' \in H} d(X_n, X_{n'})$$

Hierarchical Clustering Cont.

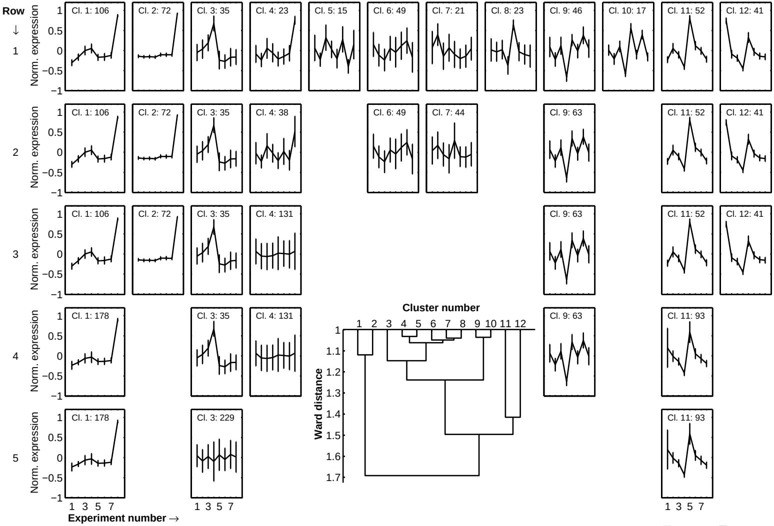
- Agglomerative (bottom-up) and divisive (top-down) approaches.
- Result unique for specific choice of $d(G, H)$.
- But not very robust to perturbation of data.
- That is if you subsample data you might get a very different dendrogram.

Hierarchical Clustering Algoritmics

- Bottom up - N clusters initially
- Merge the two clusters with smallest between-group dissimilarity
- Plot the dendrogram such that height indicates the between-group dissimilarity.



Hierarchical Clustering Algoritmics



Between Group Dissimilarity Measures

- Single linkage

$$d(G,H) = d_{\mathrm{SL}} = \min_{n \in G, n' \in H} d(X_n, X_{n'})$$

- Complete linkage

$$d(G,H) = d_{\mathrm{CL}} = \max_{n \in G, n' \in H} d(X_n, X_{n'})$$

- Group average

$$d(G,H) = d_{\mathrm{GA}} = \frac{1}{N_G N_H} \sum_{n \in G} \sum_{n' \in H} d(X_n, X_{n'})$$

- Others exist, for example Ward-distance.
- They give different results – try it out!

- Your turn! homework – cluster by hand!
- One dimensional data set of size $N = 5$

$$x_1 = 1, x_2 = 3, x_3 = 6, x_4 = 7, x_5 = 11$$

- Use Euclidian dissimilarity, dendrogram:

$$(((12)(34))5).$$

- The notation (ab) means that a and b are joined in the binary tree. a and b can themselves be binary trees.
- What group dissimilarity measure(s) $d(G, H)$ could have been used?
- Single linkage (SL) and/or complete linkage (CL)?

$$d_{\mathrm{SL}} = \min_{n \in G, n' \in H} d(X_n, X_{n'}) \qquad d_{\mathrm{CL}} = \max_{n \in G, n' \in H} d(X_n, X_{n'})$$

Radial basis function (RBF) networks

- The remainder of the lecture is on a different class of models historically called radial basis function networks.
- First we will consider the Bayes classifier again.
- But this time we will let the density for each class be a MoG.
- Next we will consider regression.
- Model joint probability $p(t, \mathbf{x})$ with MoG and get regression by

$$p(t|\mathbf{x}) = \frac{p(t, \mathbf{x})}{p(\mathbf{x})} = \frac{p(t, \mathbf{x})}{\int p(\mathbf{x}, t')dt'}$$

Signal Detection: Bayes decision theory

- We (and every child) know this already!
- Compute

$$p(C_k|\mathbf{x})$$

- and to maximize probability of correct predict the class with highest probability.
- We will use the generative approach

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})}$$

- $p(\mathbf{x}|C_k)$ = Gaussian $\rightarrow$ LDA or QDA
- $p(\mathbf{x}|C_k)$ = Mixture of Gaussians $\rightarrow$ RBF network

Signal detection with mixtures

- A change of notation k index for classes C_k and j for mixture components $P(j) = \pi_j$ and $P(j|\mathbf{x}_n) = \gamma_{nj}$.
- Let's recollect Bayes formula

$$P(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)P(C_k)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|C_k)P(C_k)}{\sum_{k'} p(\mathbf{x}|C_{k'})P(C_{k'})}$$

- Network interpretation basis functions and weights

$$\phi_k(\mathbf{x}) = \frac{p(\mathbf{x}|C_k)}{\sum_{k'} p(\mathbf{x}|C_{k'})P(C_{k'})} \qquad w_k = P(C_k)$$

- A single Gaussian per class is LDA or QDA.
- Might be over-simplified!
- use a Gaussian *mixture* for each class

$$p(\mathbf{x}|C_k) = \sum_j p(\mathbf{x}|j)P(j|C_k)$$

- The marginal density

$$p(\mathbf{x}) = \sum_k \sum_j p(\mathbf{x}|j)P(j|C_k)P(C_k) = \sum_j p(\mathbf{x}|j)P(j)$$

- with priors defined by $P(j) = \sum_k P(j|C_k)P(C_k)$.

- We are interested in a network that gives us the posterior probabilities

$$P(C_k|\mathbf{x}) = \frac{\sum_j p(\mathbf{x}|j)P(j|C_k)P(C_k)}{\sum_{j'} p(\mathbf{x}|j')P(j')} \frac{P(j)}{P(j)} = \sum_j w_{k,j}\phi_j(\mathbf{x})$$

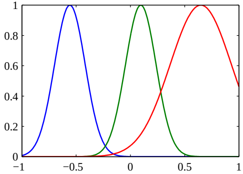
- with the definitions

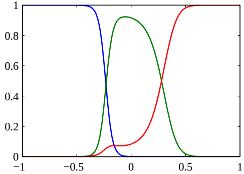
$$\phi_j(\mathbf{x}) = \frac{p(\mathbf{x}|j)P(j)}{\sum_{j'} p(\mathbf{x}|j')P(j')} = P(j|\mathbf{x})$$

$$w_{k,j} = \frac{P(j|C_k)P(C_k)}{P(j)} = P(C_k|j)$$

$$
\begin{aligned}
P(C_k|\mathbf{x}) &= \sum_j w_{k,j}\phi_j(\mathbf{x}) \\
\phi_j(\mathbf{x}) &= \frac{p(\mathbf{x}|j)P(j)}{\sum_{j'} p(\mathbf{x}|j')P(j')} = P(j|\mathbf{x}) \\
w_{k,j} &= \frac{P(j|C_k)P(C_k)}{P(j)} = P(C_k|j)
\end{aligned}
$$

So the basis functions are "normalized" by spatially variant functions, hence no longer Gaussians.





Generalization error - regression

- The mean square error of the model $y(\mathbf{x}; \mathbf{w})$ is given by

$$E = \frac{1}{2} \sum_{n=1}^{N} (y(\mathbf{x}_n; \mathbf{w}) - t_n)^2$$

- Now consider the limit of large sets, the error per example

$$E = \lim_{N \to \infty} \frac{1}{2N} \sum_{n=1}^{N} (y(\mathbf{x}_n; \mathbf{w}) - t_n)^2 = \frac{1}{2} \int (y(\mathbf{x}; \mathbf{w}) - t)^2 p(t, \mathbf{x}) \, dt d\mathbf{x}$$

- This is the average (or expected) error on a test datum $(\mathbf{x}, t)$, which we call the generalization error.

- The generalization error

$$E = \frac{1}{2} \int \int (y(\mathbf{x};\mathbf{w}) - t)^2 p(t|\mathbf{x}) p(\mathbf{x}) dt d\mathbf{x}$$

- can be rewritten using the definitions

$$\begin{aligned}
< t|\mathbf{x} > &= \int t\, p(t|\mathbf{x})\, dt \\
< t^2|\mathbf{x} > &= \int t^2\, p(t|\mathbf{x})\, dt \\
\{y - t\}^2 &= \{y - < t|\mathbf{x} > + < t|\mathbf{x} > - t\}^2 \\
&= \{y - < t|\mathbf{x} >\}^2 + 2\{y - < t|\mathbf{x} >\}\{< t|\mathbf{x} > - t\} + \\
&\quad \{< t|\mathbf{x} > - t\}^2
\end{aligned}$$

- Then the generalization error becomes

$$
\begin{aligned}
E &= \frac{1}{2}\int\int (y(\mathbf{x};\mathbf{w}) - t)^2 p(t|\mathbf{x})p(\mathbf{x})dtd\mathbf{x} \\
&= \frac{1}{2}\int\int \Big[\{y- < t|\mathbf{x} >\}^2 + 2\{y- < t|\mathbf{x} >\}\{< t|\mathbf{x} > -t\} + \ldots \\
&\quad \ldots\{< t|\mathbf{x} > -t\}^2\Big] \; p(t|\mathbf{x})p(\mathbf{x}) \; dtd\mathbf{x}
\end{aligned}
$$

- leading to the simplification

$$
\begin{aligned}
E &= \frac{1}{2}\int (y(\mathbf{x};\mathbf{w}) - < t|\mathbf{x} >)^2 p(\mathbf{x})d\mathbf{x} \\
&+ \frac{1}{2}\int \{< t^2|\mathbf{x} > - < t|\mathbf{x} >^2\} p(\mathbf{x})d\mathbf{x}
\end{aligned}
$$

- The generalization error - final expression

$$
\begin{aligned}
E &= \frac{1}{2}\int (y(\mathbf{x};\mathbf{w}) - < t|\mathbf{x} >)^2 p(\mathbf{x}) d\mathbf{x} + \\
&\quad \frac{1}{2}\int \{< t^2|\mathbf{x} > - < t|\mathbf{x} >^2\} p(\mathbf{x}) d\mathbf{x}
\end{aligned}
$$

- we see that the generalization error is minimal (as function of $y(\mathbf{x};\mathbf{w})$) if

$$
y(\mathbf{x};\mathbf{w}) = < t|\mathbf{x} >
$$

- The model should output the conditional mean, hence be a “regression”

- Start from $p(t, \mathbf{x})$ and use $p(t|\mathbf{x}) = p(t, \mathbf{x})/p(\mathbf{x})$:

$$y(\mathbf{x}) \quad = \quad < t|x >= \int t\, p(t|\mathbf{x})\, dt = \frac{\int t\, p(t, \mathbf{x})\, dt}{\int p(t', \mathbf{x})\, dt'}$$

- If our joint density is of the form with centers (ν, μ)

$$p(t, \mathbf{x}) = \sum_{j=1}^{M} P(j) \frac{1}{(2\pi\sigma_j^2)^{\frac{d+c}{2}}} \exp\left(-\frac{(\mathbf{x} - \boldsymbol{\mu}_j)^2}{2\sigma_j^2} - \frac{(t - \nu_j)^2}{2\sigma_j^2} \right)$$

- then the conditional mean is given by

$$y(\mathbf{x}) = \frac{\sum_{j=1}^{M} \frac{P(j)\nu_j}{(2\pi\sigma_j^2)^{\frac{d}{2}}} \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_j)^2}{2\sigma_j^2} \right)}{\sum_{j'=1}^{M} \frac{P(j')}{(2\pi\sigma_{j'}^2)^{\frac{d}{2}}} \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_{j'})^2}{2\sigma_{j'}^2} \right)}$$

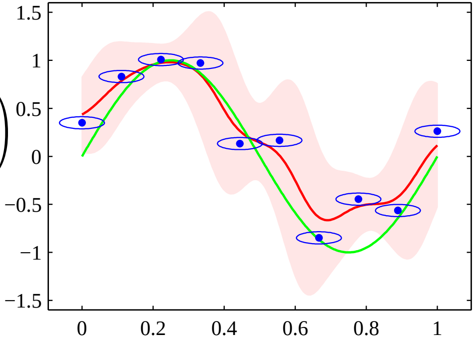
Regression with RBFs

- Prediction with the joint density
- Basis functions Gaussian

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_j)^2}{2\sigma_j^2}\right)$$

- then the RBF network regression function is

$$y_j(\mathbf{x}) = \sum_{j=1}^{M} w_{k,j}\phi(\mathbf{x}) + w_{k,0}$$



Training RBF networks

- In exercise 8 we will simply use EM.
- More generally we can do the following:
- For fixed basis functions the weights can be trained using least squares for the linear model
- For fixed weights we can using gradients (or conjugate gradients) for the the basis function parameters.

- Mixture modelling – building complex models from simpler components.
- EM Guaranteed to reach local maxima of likelihood.
- Clustering - K-means family and hierarchical
- Radial basis function (RBF) networks - train with EM and beyond.
- Thank you and hope to see again! 02460 and projects.

- Mixture of Gaussians – Bishop 2.3.9
- Mixture models – Bishop 9, 9.2-9.3.1
- K-means – Bishop 9.1, 9.3.2
- Hierarchical clustering – Hastie, Tibshirani and Friedman 14.3.12
- Radial basis function (RBF) networks Bishop 6.3.
- Alternative free pdf books:
- Hastie, Tibshirani and Friedman, The Elements of Statistical Learning, Springer and
- MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge

EM algorithm the general view v2

- Optimization of the likelihood simpler by introduction of extra set of latent (=unobserved) variables $\mathbf{Z}$.
- We introduce the simpler (factorizing) complete likelihood

$$p(\mathbf{X}, \mathbf{Z}|\mathbf{w})$$

- Incomplete (=normal) likelihood is obtained by marginalization:

$$p(\mathbf{X}|\mathbf{w}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\mathbf{w})$$

EM algorithm the general view v2 cont.

- E-step: Evaluate $P(\mathbf{Z}|\mathbf{X}, \mathbf{w}^{\mathrm{old}})$
- M-step: Find $\mathbf{w}^{\mathrm{new}}$ by

$$\begin{aligned} \mathbf{w}^{\mathrm{new}} &= \operatorname*{argmax}_{\mathbf{w}} \mathcal{Q}(\mathbf{w}, \mathbf{w}^{\mathrm{old}}) \\ \mathcal{Q}(\mathbf{w}, \mathbf{w}^{\mathrm{old}}) &= \sum_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{X}, \mathbf{w}^{\mathrm{old}}) \log p(\mathbf{X}, \mathbf{Z}|\mathbf{w}) \end{aligned}$$

- For mixture model $P(z_{nk} = 1|\mathbf{X}, \mathbf{w}^{\mathrm{old}}) = \gamma_{nk} = p(k|\mathbf{x}_n)$.