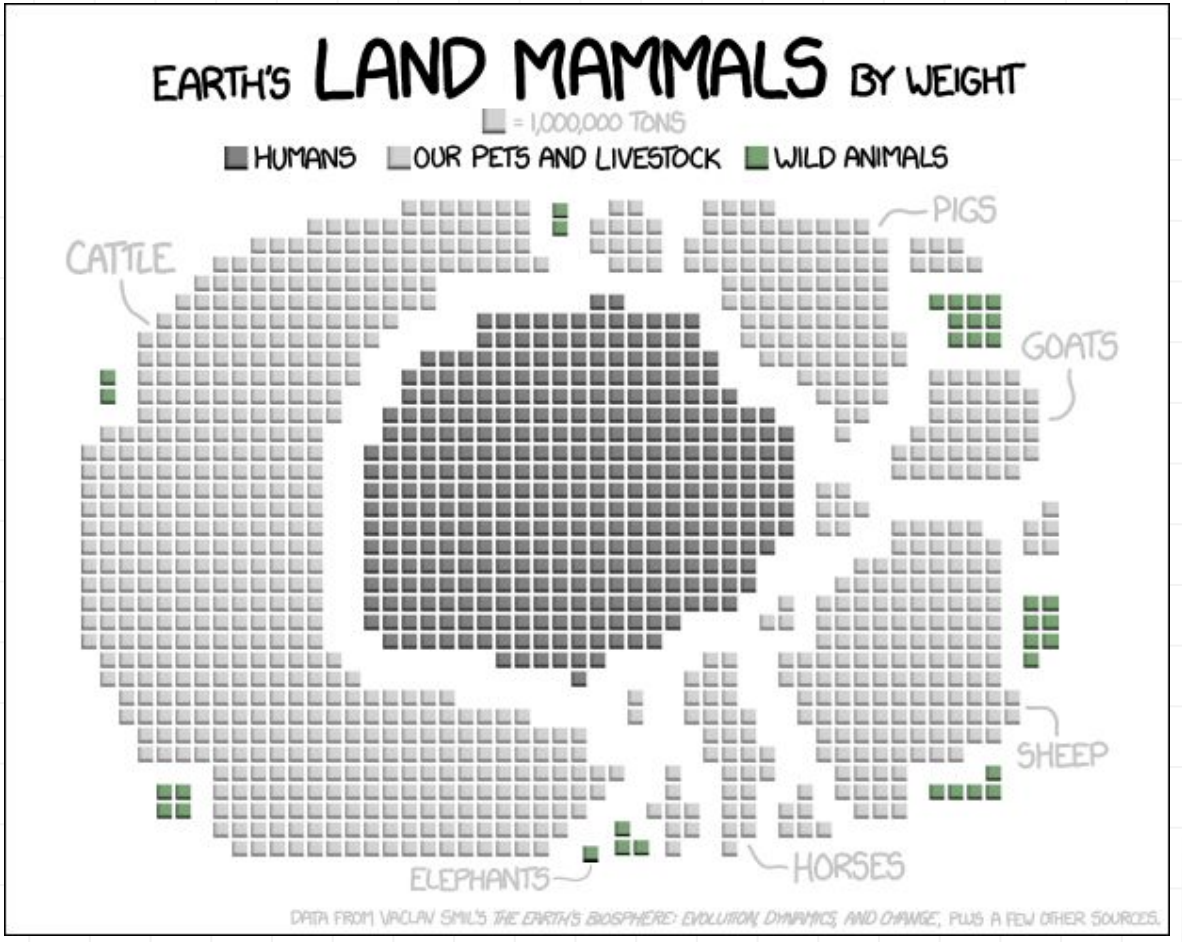
https://xkcd.com/1338/



Mixture Models and Expectation Maximisation

Clustering and density approximation

K-means

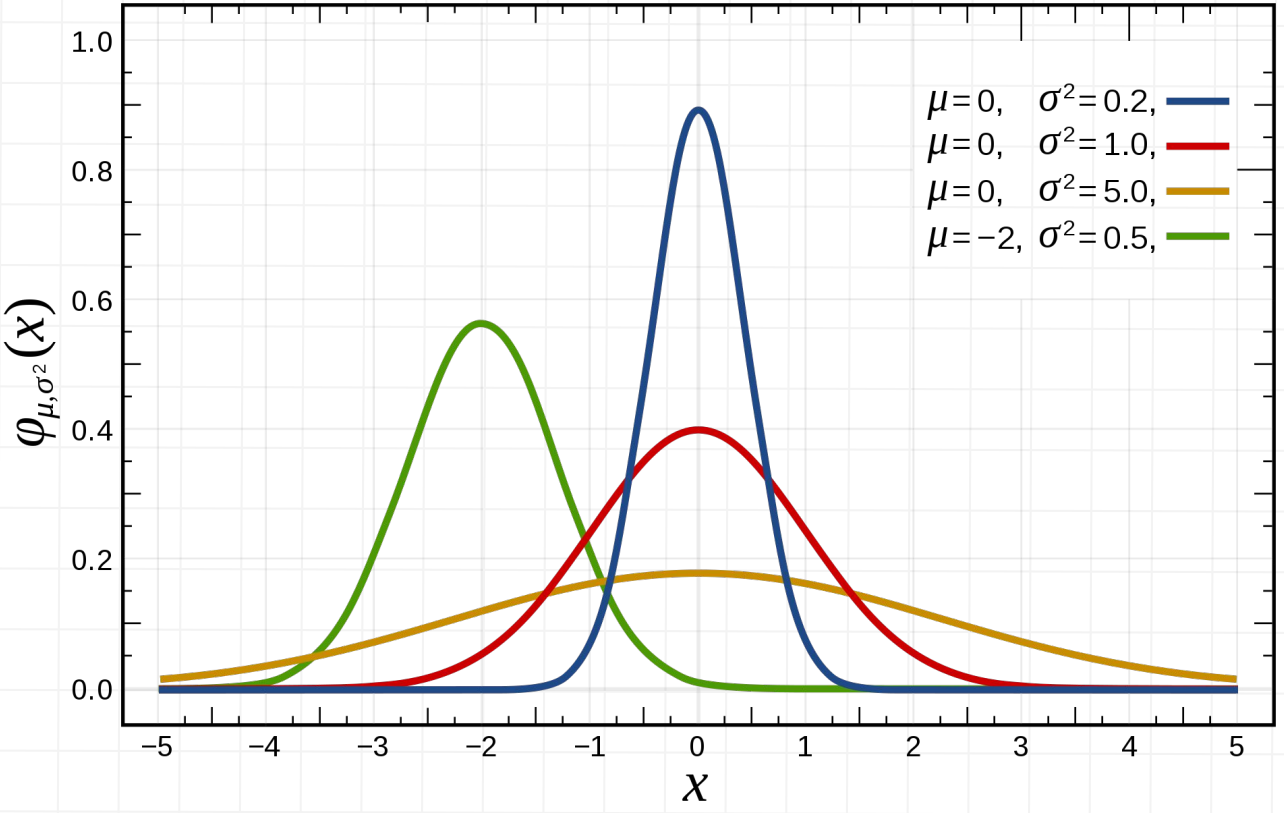
Gaussian mixture models

- no workshop Fri
  clustering / EM #2 after break
- Assignment 1 due today.

Discrete latent variables

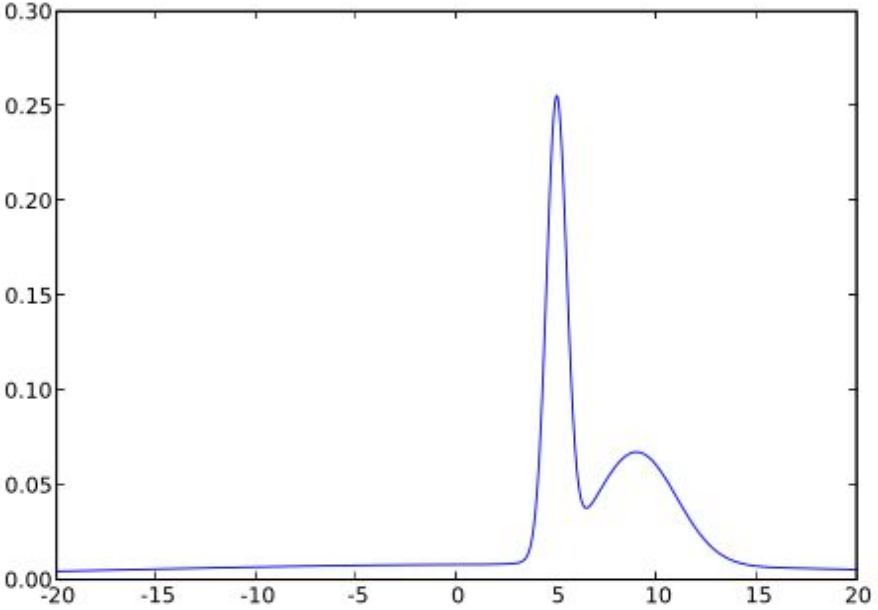
Expectation maximization, a general technique for finding maximum likelihood estimators in latent variable models *-- or, how to solve hard-looking problems by solving several easy-looking pieces.*

Is there something more in the familiar bell curves?



https://en.wikipedia.org/wiki/Normal_distribution

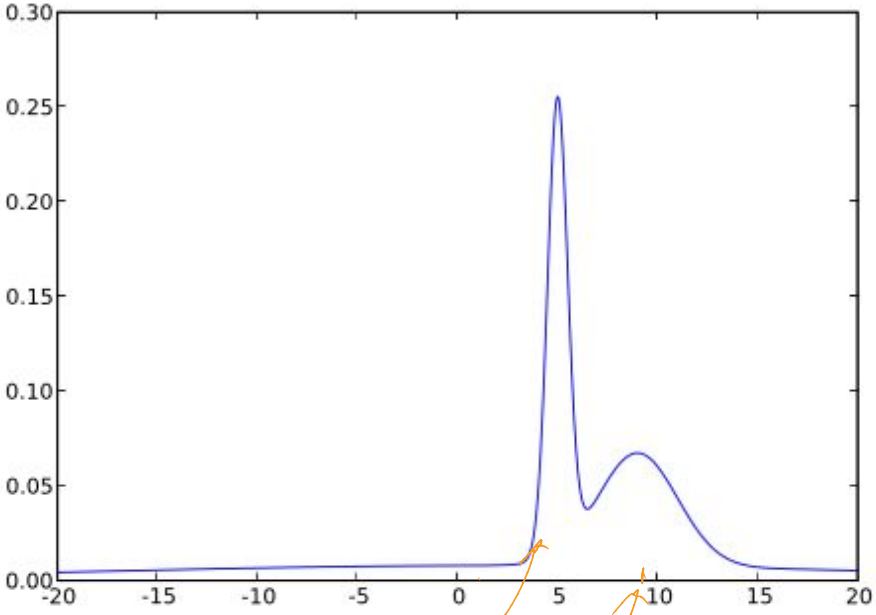
A “Wallaby” distribution

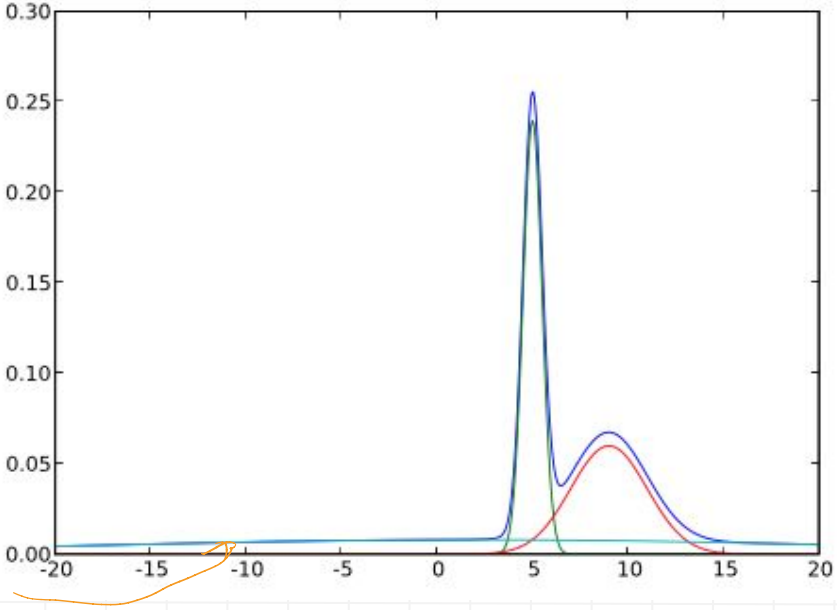


https://animalscomparison.com/wallaby-vs-kangaroo-fight-comparison-who-will-win/

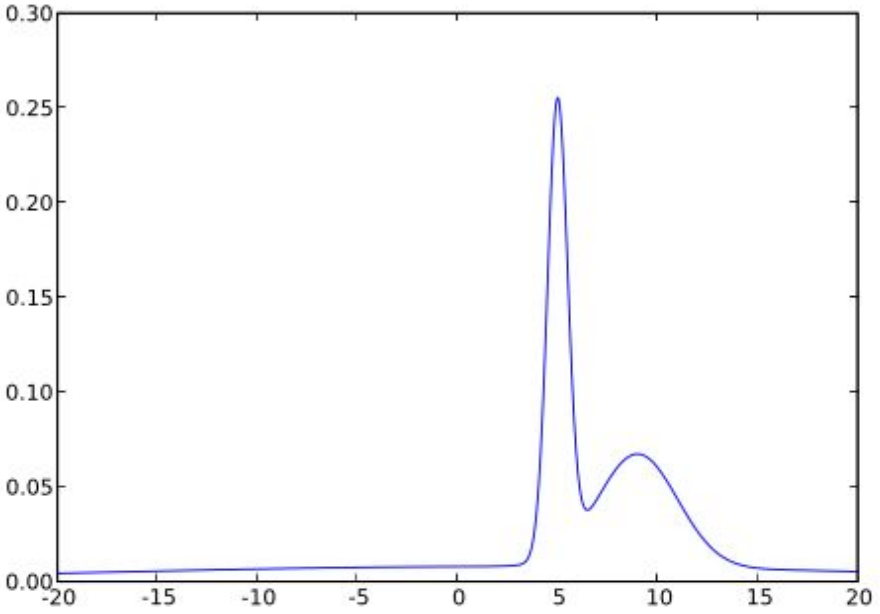
A “Wallaby” distribution

https://animalscomparison.com/wallaby-vs-kangaroo-fight-comparison-who-will-win/





$$p(x) = \frac{3}{10}\mathcal{N}(x \mid 5, 0.5) + \frac{3}{10}\mathcal{N}(x \mid 9, 2) + \frac{4}{10}\mathcal{N}(x \mid 2, 20)$$



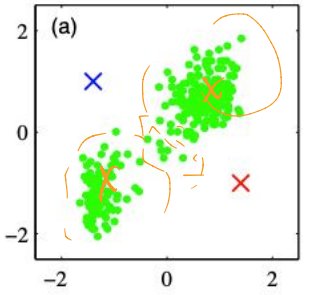
any smooth density can be approximated to arbitrary precision by a Gaussian mixture model with enough components.

Park, J. and Sandberg, I.W., 1991. Universal approximation using radial-basis-function networks. *Neural computation*, *3*(2), pp.246-257.

Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

$$p(x) = \frac{3}{10}\mathcal{N}(x \mid 5, 0.5) + \frac{3}{10}\mathcal{N}(x \mid 9, 2) + \frac{4}{10}\mathcal{N}(x \mid 2, 20)$$

Clustering 101: K-Means



- Given a set of data $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ where $\mathbf{x}_n \in \mathbb{R}^D$, $n = 1, \ldots, N$.
- Goal: Partition the data into K clusters.

given

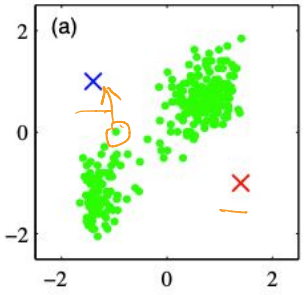
$K=2$

A.K.A.

Vector Quantization

- Each cluster contains points close to each other.
- Introduce a prototype $\boldsymbol{\mu}_k \in \mathbb{R}^D$ for each cluster.
- Goal: Find
  1. a set prototypes $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$, each representing a different cluster.
  2. an assignment of each data point to exactly one cluster.

K-means clustering: representation + loss



- Start with arbitrary chosen prototypes $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$.
  1. Assign each data point to the closest prototype.
  2. Calculate new prototypes as the mean of all data points assigned to each of them.

- Binary indicator variables

$$r_{nk} = \begin{cases} 1, & \text{if data point } \mathbf{x}_n \text{ belongs to cluster } k \\ 0, & \text{otherwise} \end{cases}$$

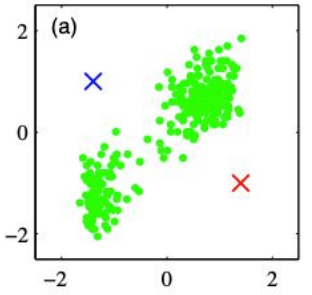
  using the 1-of-K coding scheme.

- Define a distortion measure

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \qquad (9.1)$$

- Find the values for $\{r_{nk}\}$ and $\{\boldsymbol{\mu}_k\}$ so as to minimise J.

K-means clustering: algorithm



$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \qquad (9.1)$$

But $\{r_{nk}\}$ depends on $\{\boldsymbol{\mu}_k\}$, and $\{\boldsymbol{\mu}_k\}$ depends on $\{r_{nk}\}$.

IF $\{\boldsymbol{\mu}_k\}$ are given, we only need to determine r_{nk}

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases} \qquad (9.2)$$

$\{\mu_k\}$ fixed

take the min

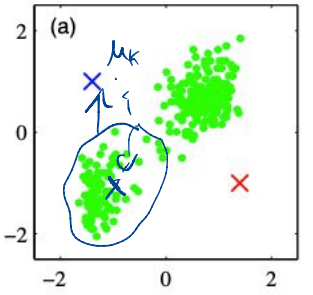
$\|x_n - \mu_1\|^2$

$\|x_n - \mu_2\|^2$

⋮

$\|x_n - \mu_6\|^2$

K-means clustering: algorithm



$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \qquad (9.1)$$

But $\{r_{nk}\}$ depends on $\{\boldsymbol{\mu}_k\}$, and $\{\boldsymbol{\mu}_k\}$ depends on $\{r_{nk}\}$.

Assume r_{nk} is known, what would $\{\boldsymbol{\mu}_k\}$ be?

$$2 \sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0 \qquad (9.3)$$

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}. \qquad (9.4)$$

↖ # $\{x_n\}$ assigned to cluster K

K-means clustering: recap

Where to start?
When to stop?
Why does this work?

random
randomly pick K out of {x_n}
- kmean++

Expectation step

Re-assign data points to clusters, determine r_{nk}

fix μ_k
min. J

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases} \tag{9.2}$$

E-step:
stop if $\{r_{nk}\}$ does not change.

Maximisation step

Re-compute the cluster means - update $\{\boldsymbol{\mu}_k\}$

fix r_{nk}
min J

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}. \tag{9.4}$$

M-step
Stop if $\{\mu_k\}$ do not chang
or $\sum_k \|\mu_k^{(t+1)} - \mu_k^{(t)}\|^2 \leq \varepsilon$

OR maximum iteration reached.

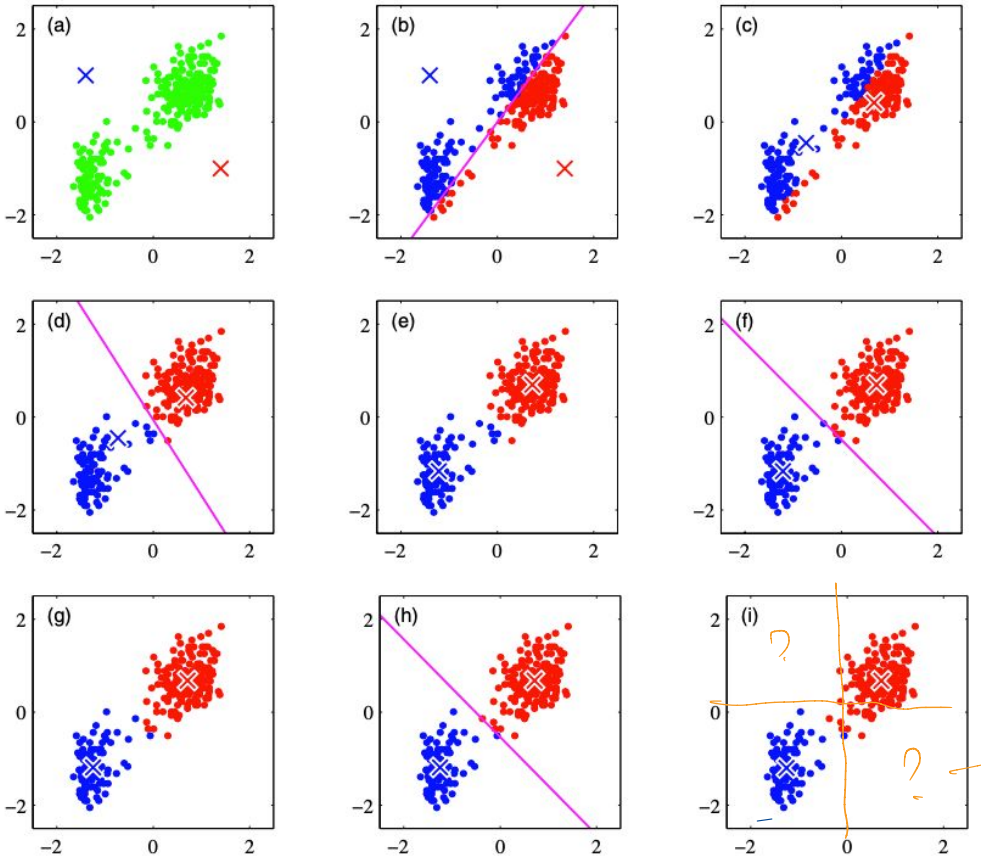
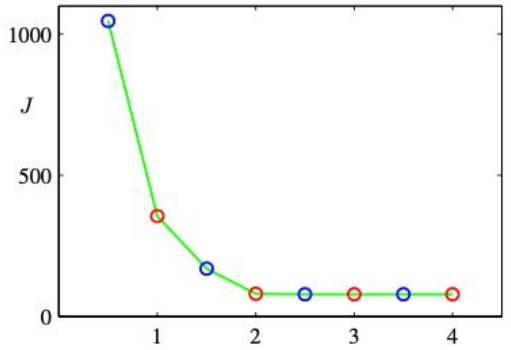


Figure 9.1 Illustration of the K-means algorithm using the re-scaled Old Faithful data set. (a) Green points denote the data set in a two-dimensional Euclidean space. The initial choices for centres μ_1 and μ_2 are shown by the red and blue crosses, respectively. (b) In the initial E step, each data point is assigned either to the red cluster or to the blue cluster, according to which cluster centre is nearer. This is equivalent to classifying the points according to which side of the perpendicular bisector of the two cluster centres, shown by the magenta line, they lie on. (c) In the subsequent M step, each cluster centre is re-computed to be the mean of the points assigned to the corresponding cluster. (d)–(i) show successive E and M steps through to final convergence of the algorithm.

Plot of the cost function J given by (9.1) after each E step (blue points) and M step (red points) of the K-means algorithm for the example shown in Figure 9.1. The algorithm has converged after the third M step, and the final EM cycle produces no changes in either the assignments or the prototype vectors.



- Initial condition crucial for convergence.
- What happens, if at least one cluster centre is too far from all data points?
- Complex step: Finding the nearest neighbour. (Use triangle inequality; build K-D trees, ...)
- Generalise to non-Euclidean dissimilarity measures $\mathcal{V}(\mathbf{x}_n, \boldsymbol{\mu}_k)$ (called K-medoids algorithm),

$$\widetilde{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \mathcal{V}(\mathbf{x}_n, \boldsymbol{\mu}_k).$$

- Online stochastic algorithm
  1. Draw data point $\mathbf{x}_n$ and locate nearest prototype $\boldsymbol{\mu}_k$.
  2. Update only $\boldsymbol{\mu}_k$ using decreasing learning rate η_n

$$\boldsymbol{\mu}_k^{\text{new}} = \boldsymbol{\mu}_k^{\text{old}} + \eta_n(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{old}}).$$

find r_{nk} : nearest μ_k to x_n

- finds a local min. of J
- what to do if $\exists k, \{r_{nk}, \forall n\} = 0$

can't update μ_k,

re-initialise μ_k to another data point

$O(NKD)$ ← data dim.

K-means for illustrating image segmentation and data compression

$K = 2$

4.2%

$K = 3$

8.3%

$K = 10$

16.7%

Original image

100 %

- Segment an image into regions of reasonable homogeneous appearance.
- Each pixel is a point in $\mathbb{R}^3$ (red, blue, green). (Note that the pixel intensities are bounded in the range $[0, 1]$ and therefore this space is strictly speaking not Euclidean).
- Run K-means on all points of the image until convergence. Replace all pixels with the corresponding mean $\boldsymbol{\mu}_k$.
- Results in an image with a palette only K different colours.
- There are much better approaches to image segmentation (but it is an active research topic), this here serves only to illustrate K-means.

- Store the code-book vectors $\boldsymbol{\mu}_k$.
- Store the data in the form of references (labels) to the code-book. Each data point has a label in the range $[1, \ldots, K]$.
- New data points are also compressed by finding the closest code-book vector and then storing only the label.
- This technique is also called vector quantisation.

Mixture Models and Expectation Maximisation

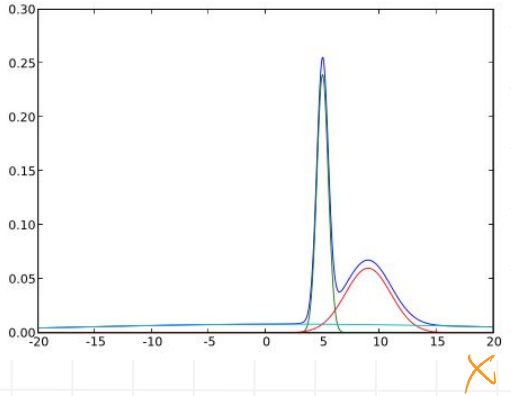
Clustering and density approximation

K-means

Gaussian mixture models

Gaussian Mixture Models (GMM)

Mixture distributions are formed by taking linear combinations of more basic distributions.



$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad (2.188)$$ also (9.7)

$$0 \leqslant \pi_k \leqslant 1. \qquad \sum_{k=1}^{K} \pi_k = 1.$$

complex densities. By using a sufficient number of Gaussians, and by adjusting their means and covariances as well as the coefficients in the linear combination, almost any continuous density can be approximated to arbitrary accuracy.

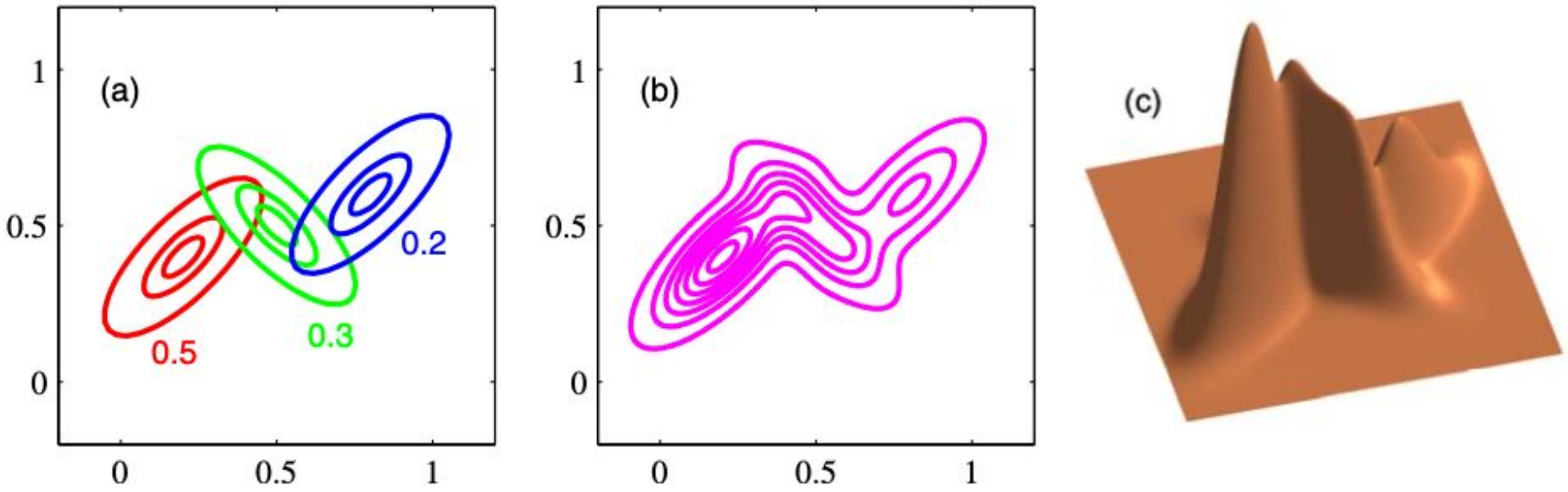


Figure 2.23 Illustration of a mixture of 3 Gaussians in a two-dimensional space. (a) Contours of constant density for each of the mixture components, in which the 3 components are denoted red, blue and green, and the values of the mixing coefficients are shown below each component. (b) Contours of the marginal probability density $p(\mathbf{x})$ of the mixture distribution. (c) A surface plot of the distribution $p(\mathbf{x})$.

GMM as a latent variable model

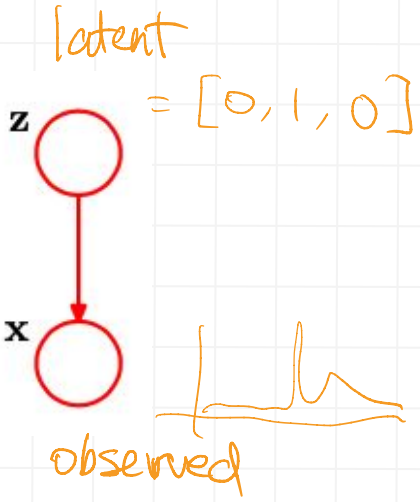


Figure 9.4 Graphical representation of a mixture model, in which the joint distribution is expressed in the form $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$.

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad (2.188)$$

also (9.7)

$$0 \leqslant \pi_k \leqslant 1. \qquad \sum_{k=1}^{K} \pi_k = 1.$$

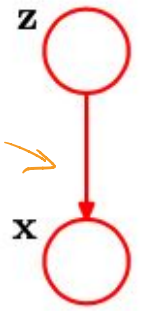
$$p(z_k = 1) = \pi_k$$

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}. \qquad (9.10)$$

{0, 1}

Conditional and marginal distributions

$p(z_k = 1) = \pi_k$



$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

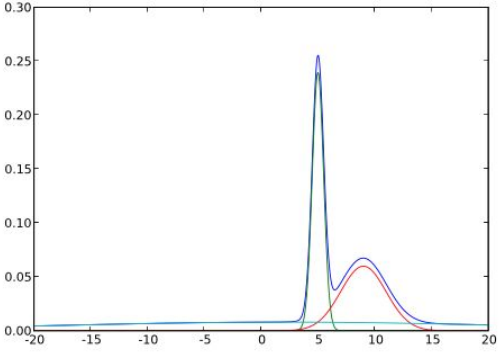
$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}. \qquad (9.11)$$

only one of the z_k is active

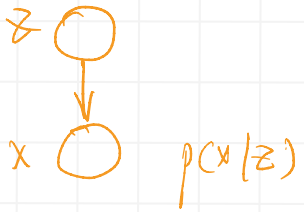
x is gaussian conditioned on z

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad (9.12)$$

$$\sum_{\mathbf{z}} \prod_{k=1}^{K} \pi_k^{z_k} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}$$



Posterior



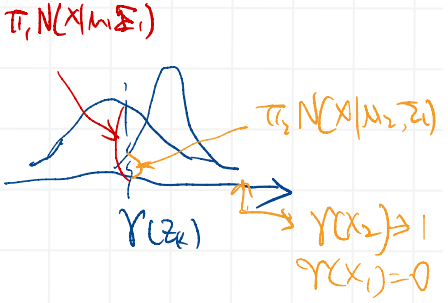
Conditional probability of $\mathbf{z}$ given $\mathbf{x}$ by Bayes' theorem

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)}$$

$$= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \quad (9.13)$$

$\frac{P(x, z_k=1)}{P(x)}$

function of x



$\gamma(z_k)$ is the responsibility of component k to 'explain' the observation $\mathbf{x}$.

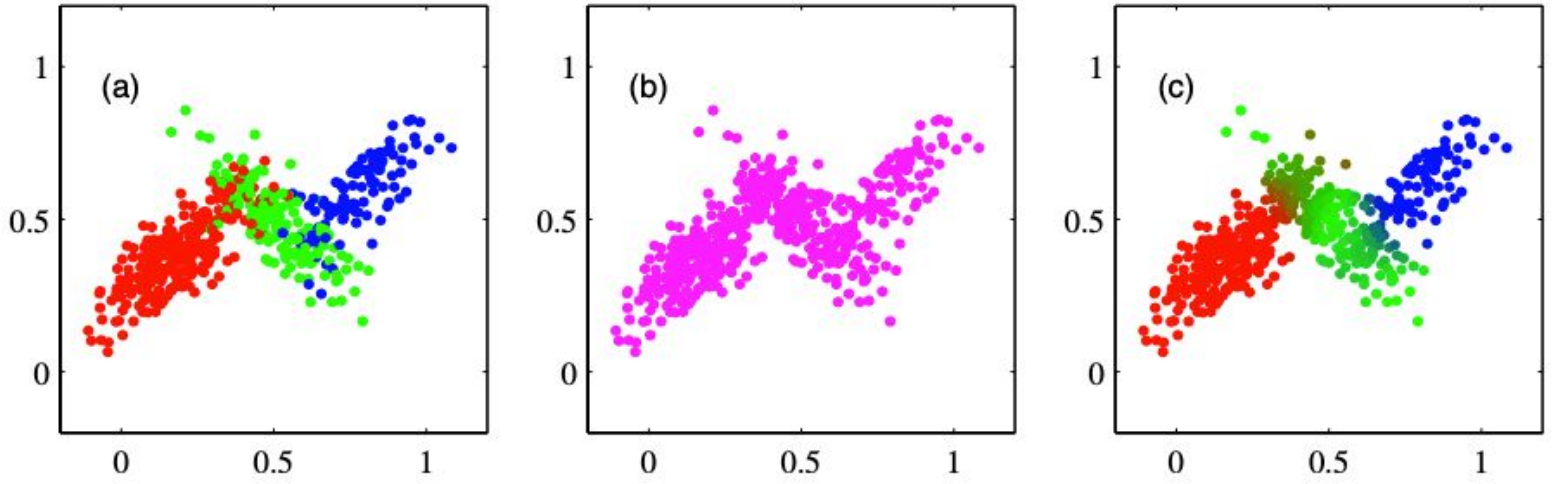


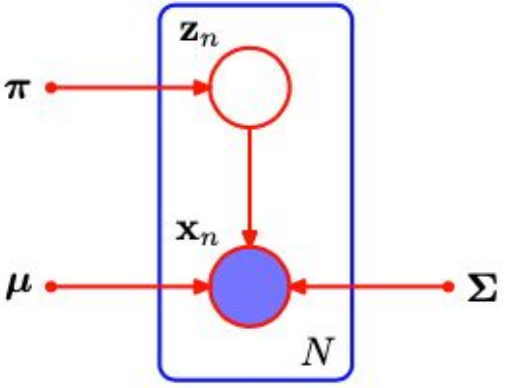
Figure 9.5 Example of 500 points drawn from the mixture of 3 Gaussians shown in Figure 2.23. (a) Samples from the joint distribution $p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$ in which the three states of $\mathbf{z}$, corresponding to the three components of the mixture, are depicted in red, green, and blue, and (b) the corresponding samples from the marginal distribution $p(\mathbf{x})$, which is obtained by simply ignoring the values of $\mathbf{z}$ and just plotting the $\mathbf{x}$ values. The data set in (a) is said to be *complete*, whereas that in (b) is *incomplete*. (c) The same samples in which the colours represent the value of the responsibilities $\gamma(z_{nk})$ associated with data point $\mathbf{x}_n$, obtained by plotting the corresponding point using proportions of red, blue, and green ink given by $\gamma(z_{nk})$ for $k = 1, 2, 3$, respectively

Maximum likelihood estimation

Observed: $\{x_n\}$; Unobserved/latent: $\{z_n\}$

Need to estimate: $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$

Figure 9.6 Graphical representation of a Gaussian mixture model for a set of N i.i.d. data points $\{\mathbf{x}_n\}$, with corresponding latent points $\{\mathbf{z}_n\}$, where $n = 1, \ldots, N$.



$P(x_n)$

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}. \qquad (9.14)$$

Issues in maximum likelihood

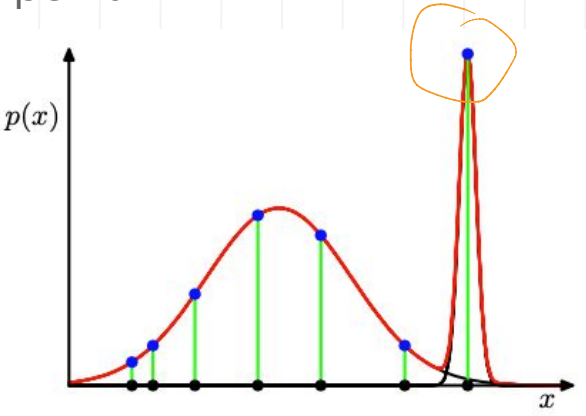
Parameter space symmetries, or identifiability problem

- A K component mixture has a total of $K!$ equivalent solutions corresponding to the $K!$ ways of assigning K sets of parameters to K solutions.
- Also called identifiability problem. Needs to be considered when the parameters discovered by a model are interpreted.

at least K!
global minimums
that are equivalent to
each other

Singularity due to component *collapsing* onto one data point

Figure 9.7 Illustration of how singularities in the likelihood function arise with mixtures of Gaussians. This should be compared with the case of a single Gaussian shown in Figure 1.14 for which no singularities arise.



Challenges in maximising the likelihood function

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}. \qquad (9.14)$$

- Maximising the log likelihood of a Gaussian mixture is more complex then for a single Gaussian. Summation over all K components inside of the logarithm make it harder.
- Setting the derivatives of the log likelihood to zero does not longer result in a closed form.
- May use gradient-based optimisation.
- Or EM algorithm. Stay tuned.

Whole problem: solve for π, μ, Σ

Divide-and-conquer: solve for $\boldsymbol{\mu}_k$

$\propto$ const $\exp(-\mu_k^2 \cdots)$

$$L = \ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}. \quad (9.14)$$

only one term depend on μ_k

Differentiate wrt $\boldsymbol{\mu}_k$

$$\frac{\partial L}{\partial \mu_k} = \sum_{n=1}^{N} \frac{1}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)} \pi_k \left[- \text{const} \cdot \exp[\cdots \mu_k^2 \cdots] \right]$$

$$0 = -\sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{\gamma(z_{nk})} \boldsymbol{\Sigma}_k(\mathbf{x}_n - \boldsymbol{\mu}_k) \quad (9.16)$$

$$0 = -\sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{\gamma(z_{nk})} \boldsymbol{\Sigma}_k(\mathbf{x}_n - \boldsymbol{\mu}_k) \tag{9.16}$$

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \tag{9.17}$$

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}).$$

of "effective" points in cluster k

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}} \tag{9.19}$$

Solve for π_k

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right) \tag{9.20}$$

$$0 = \sum_{n=1}^{N} \frac{\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda \tag{9.21}$$

$\lambda = -N$

$\sum_n \gamma(z_{nk})$

$$\pi_k = \frac{N_k}{N} \tag{9.22}$$

So, what happened?

Given γ_{nk} , solve for $\pi_k \boldsymbol{\mu}_k \boldsymbol{\Sigma}_k$

But $\gamma_{nk,}$ is computed from $\{x_n\}$, $\pi_k \boldsymbol{\mu}_k \boldsymbol{\Sigma}_k$

...

In the *expectation* step, or E step, we use the current values for the parameters to evaluate the posterior probabilities, or responsibilities, given by (9.13).

$$\gamma_k = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{9.13}$$

We then use these probabilities in the *maximization* step, or M step, to re-estimate the means, covariances, and mixing coefficients using the results (9.17), (9.19), and (9.22).

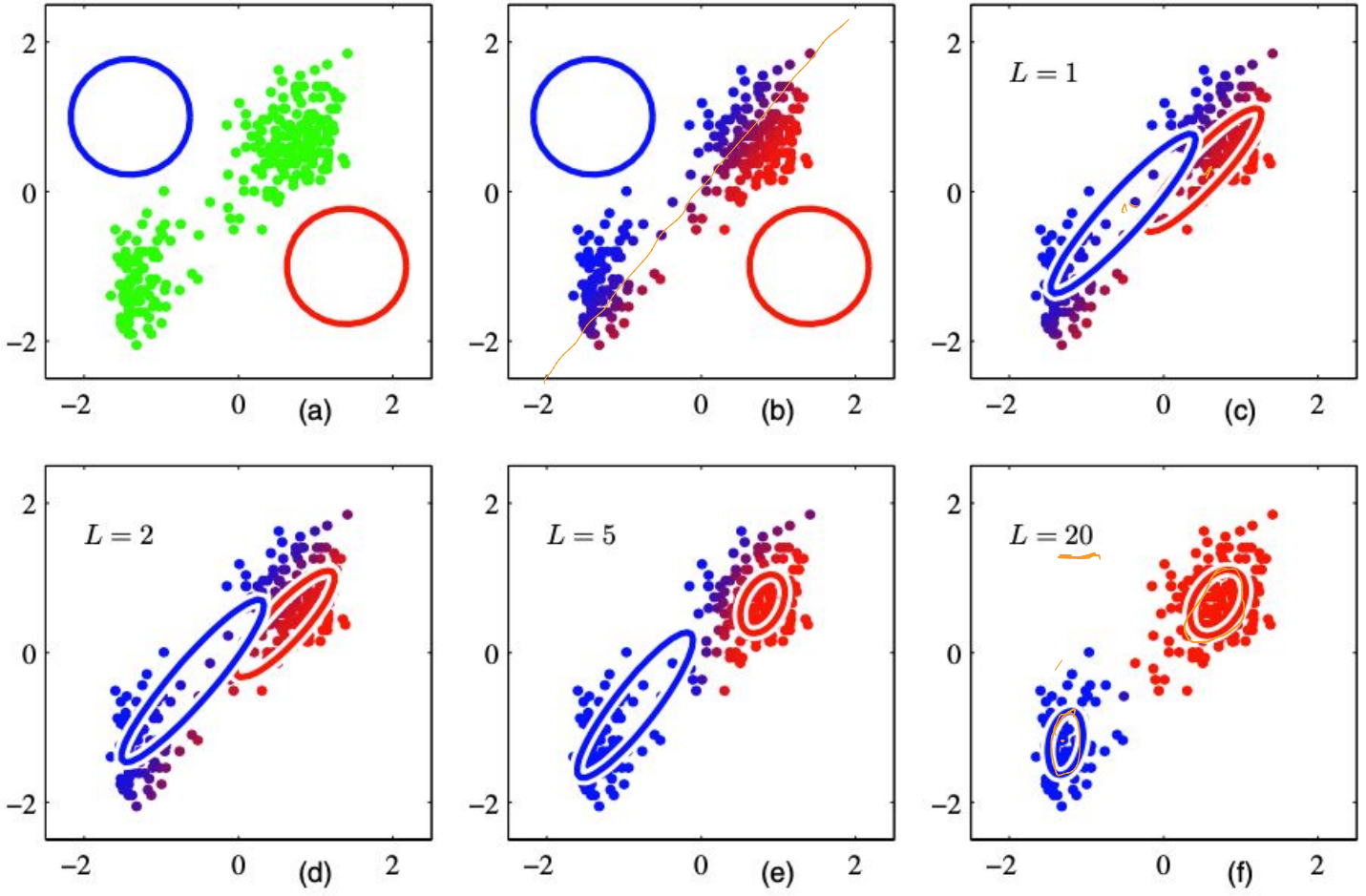
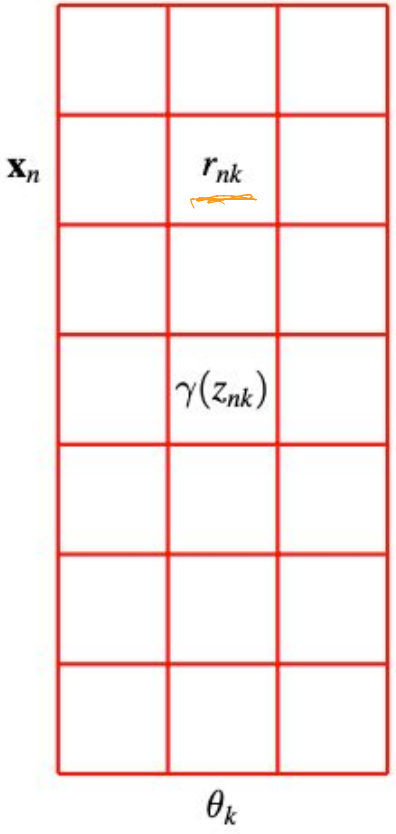


Figure 9.8 Illustration of the EM algorithm using the Old Faithful set as used for the illustration of the K-means algorithm in Figure 9.1. See the text for details.

Responsibilities -- one connection to K-means



For k-means clustering,
we have hard assignments

For GMM,
we have soft assignments

$$r_{nk} = \begin{cases} 1 & \text{if } x_n \text{ belongs to cluster } k \\ 0 & \text{o.w.} \end{cases}$$

$$\gamma(z_{nk}) = P(z_{nk}=1 \mid x_n)$$

EM for Gaussian Mixtures

Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising the means and covariances of the components and the mixing coefficients).

1. Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k, and evaluate the initial value of the log likelihood.

using k-means to init μ_k

2. **E step**. Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

3. **M step**. Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \tag{9.24}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})^{\text{T}} \tag{9.25}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \tag{9.26}$$

where

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}). \tag{9.27}$$

4. Evaluate the log likelihood

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{9.28}$$

and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

Mixture Models and Expectation Maximisation

Clustering and density approximation

K-means

Gaussian mixture models

NEXT TIME:

why "expectation" & "max." ?

why does EM work ?

Discrete latent variables

Expectation maximization, a general technique for finding maximum likelihood estimators in latent variable models *-- or, how to solve hard-looking problems by solving several easy-looking pieces.*