Mixture of Gaussians

Lecture 07

A reading list for this week:

Bishop 9.1,9.2,

Hastie et al 13.1, 13.2, 14.3 - 14.3.11

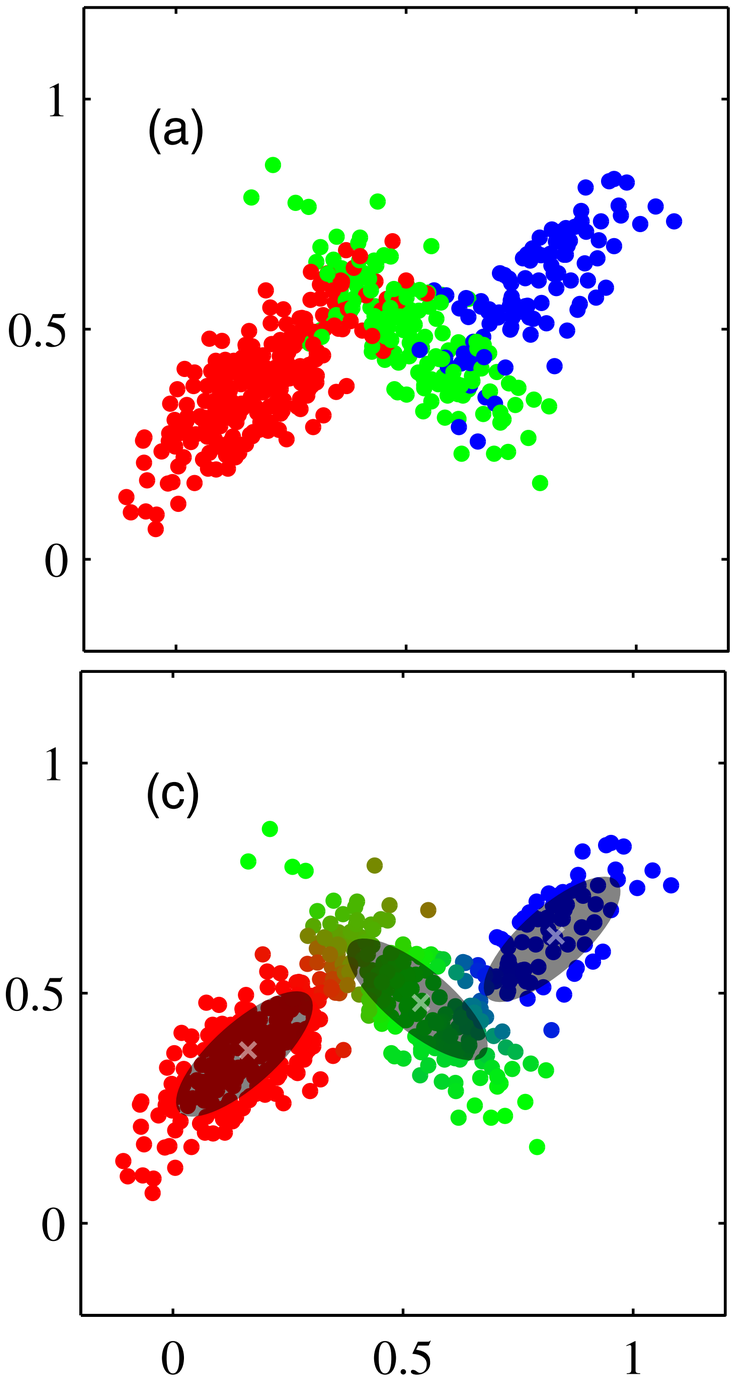
Bonaccorso “Clustering fundamentals”

Logistics

- No pre lecture, life happened. Sorry

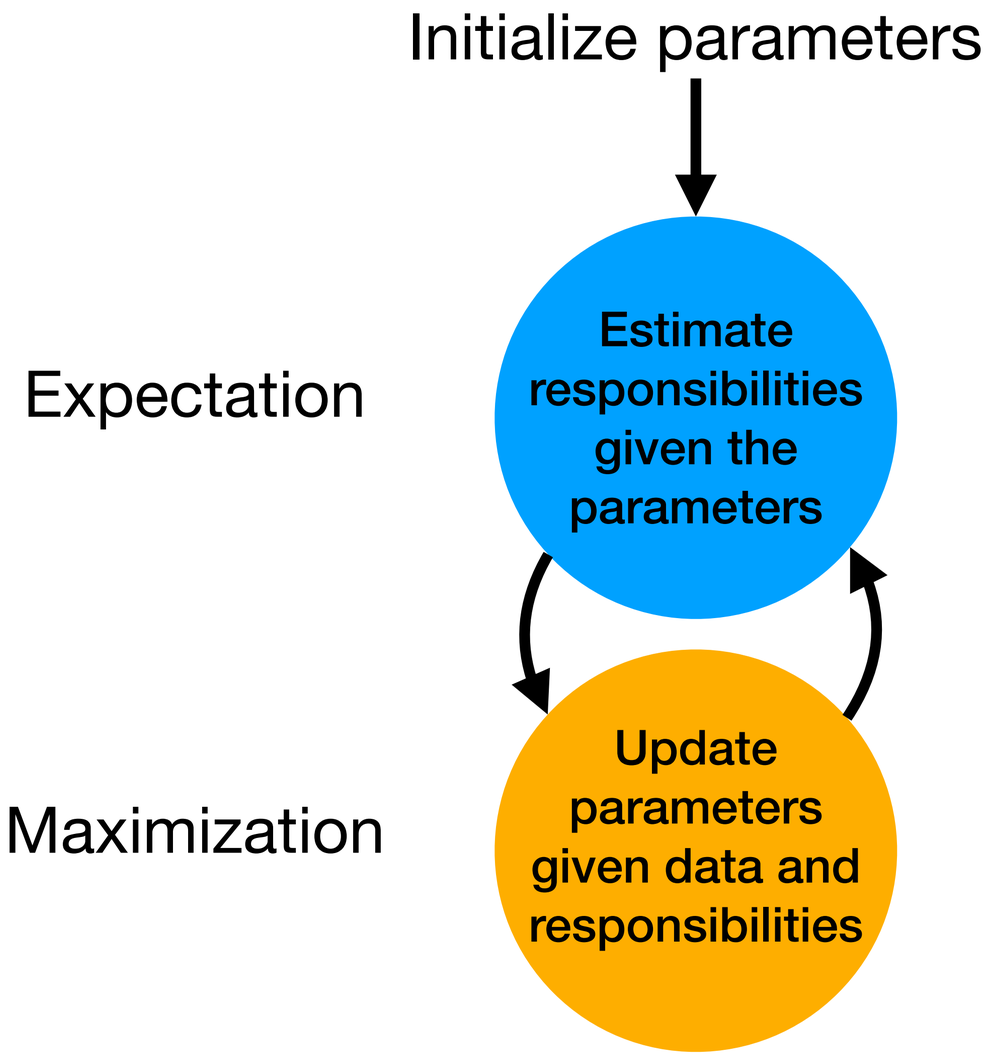
Mixture models

- Assume your data is generated by several independent processes
- Fit a model where you try to figure out
  - What are the parameters of each process
  - What is the responsibility (ie. likelihood) of each process for generating a given datapoint



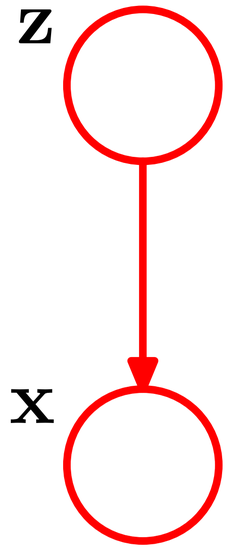
Expectation-Maximization for mixture models

- Powerful way to do MLE
- Can be generalized to do Bayesian inference
- Iterative; potentially high computational cost
- Dempster-Laird-Rubin, 1977
- E-step: what is the responsibility of each element of the mixture for generating each datapoint
  - In K-means this is 0/1; here we estimate a probability
- M-step: re-estimate the parameters using the current responsibilities



Latent variables

- Latent variables represents things we cannot directly observe but will instead infer
  - Which process made this datapoint
  - Other things we will see later too!



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

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \qquad \text{A mixture of Gaussians}$$

Let us introduce a K-dimensional binary random variable $\mathbf{z}$ having a 1-of-K representation in which a particular element z_k is equal to 1 and all other elements are equal to 0. The values of z_k therefore satisfy $z_k \in \{0, 1\}$ and $\sum_k z_k = 1$, and we see that there are K possible states for the vector $\mathbf{z}$ according to which element is nonzero. We shall define the joint distribution $p(\mathbf{x}, \mathbf{z})$ in terms of a marginal distribution $p(\mathbf{z})$ and a conditional distribution $p(\mathbf{x}|\mathbf{z})$, corresponding to the graphical model in Figure 9.4. The marginal distribution over $\mathbf{z}$ is specified in terms of the mixing coefficients π_k, such that

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

$0 \leqslant \pi_k \leqslant 1$

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

+z_k is one hot =

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

Similarly, the conditional distribution of $\mathbf{x}$ given a particular value for $\mathbf{z}$ is a Gaussian

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

which can also be written in the form

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

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

Marginal and joint distributions

Another quantity that will play an important role is the conditional probability of $\mathbf{z}$ given $\mathbf{x}$. We shall use $\gamma(z_k)$ to denote $p(z_k = 1|\mathbf{x})$, whose value can be found using Bayes' theorem

$$\begin{aligned}\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) &= \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \end{aligned} \tag{9.13}$$

We shall view π_k as the prior probability of $z_k = 1$, and the quantity $\gamma(z_k)$ as the corresponding posterior probability once we have observed $\mathbf{x}$. As we shall see later, $\gamma(z_k)$ can also be viewed as the *responsibility* that component k takes for 'explaining' the observation $\mathbf{x}$.

- Data $\mathbf{X}$ is an NxM design matrix, latent variable $\mathbf{Z}$ is an NxK matrix where rows are one-hot
- Assume datapoints are i.i.d. then

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

Set $\frac{\partial \mathcal{L}}{\partial \mu} = 0$ to find the solutions to describing the means

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

where we have made use of the form (2.43) for the Gaussian distribution. Note that the posterior probabilities, or responsibilities, given by (9.13) appear naturally on the right-hand side. Multiplying by $\boldsymbol{\Sigma}_k^{-1}$ (which we assume to be nonsingular) and rearranging we obtain

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

Weighted

empirical mean

where we have defined

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}). \tag{9.18}$$

If we set the derivative of $\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to $\boldsymbol{\Sigma}_k$ to zero, and follow a similar line of reasoning, making use of the result for the maximum likelihood solution for the covariance matrix of a single Gaussian, we obtain

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

Weighted

empirical covariance

Finally, we maximize $\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to the mixing coefficients π_k. Here we must take account of the constraint (9.9), which requires the mixing coefficients to sum to one. This can be achieved using a Lagrange multiplier and maximizing the following quantity

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

which gives

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

where again we see the appearance of the responsibilities. If we now multiply both sides by π_k and sum over k making use of the constraint (9.9), we find $\lambda = -N$. Using this to eliminate λ and rearranging we obtain

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

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

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{9.23}$$

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

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

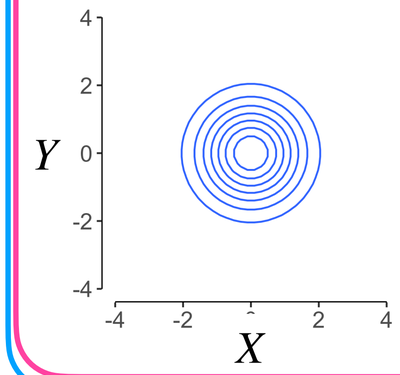
$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\text{T}} \tag{9.25}$$

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

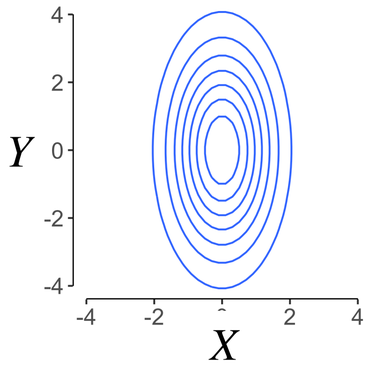
where

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

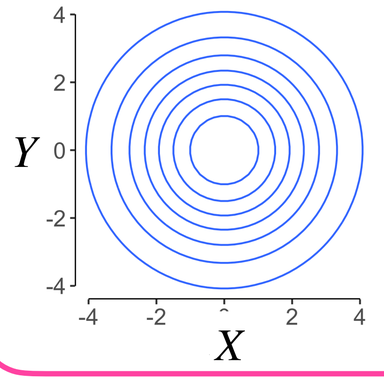
$\rho = 0, \sigma_X = \sigma_Y = 1$



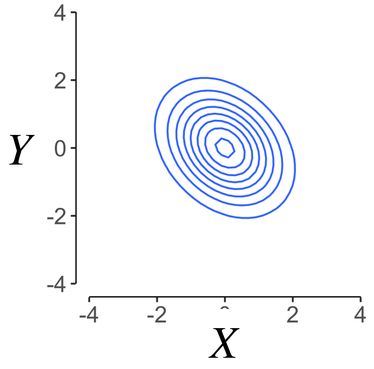
$\rho = 0, \sigma_X = 1, \sigma_Y = 2$



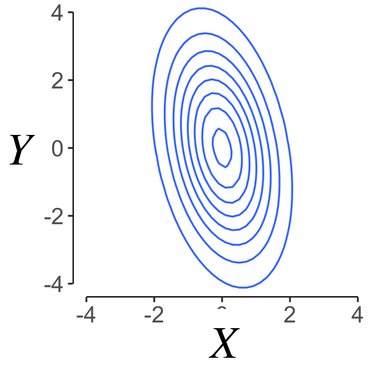
$\rho = 0, \sigma_X = \sigma_Y = 2$



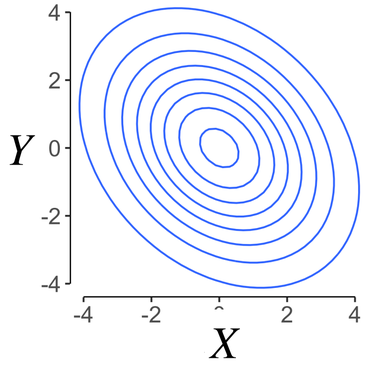
$\rho = -0.3, \sigma_X = \sigma_Y = 1$



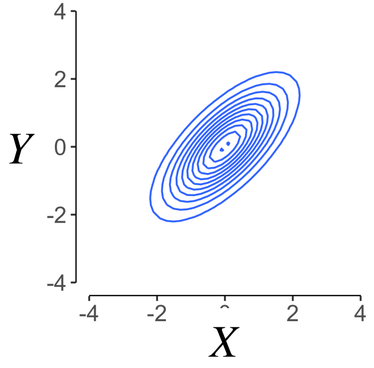
$\rho = -0.3, \sigma_X = 1, \sigma_Y = 2$



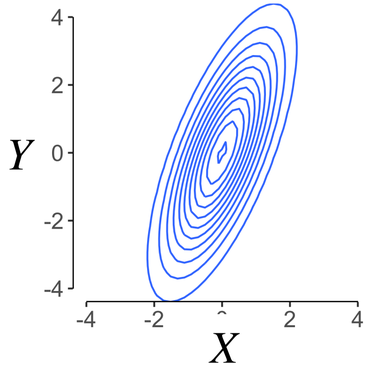
$\rho = -0.3, \sigma_X = \sigma_Y = 2$



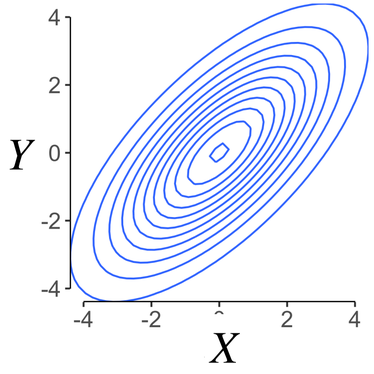
$\rho = 0.7, \sigma_X = \sigma_Y = 1$



$\rho = 0.7, \sigma_X = 1, \sigma_Y = 2$



$\rho = 0.7, \sigma_X = \sigma_Y = 2$



We can choose to constrain our GMM

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{pmatrix}$$

Axis-aligned: $\rho = 0$

Spherical: all σs are the same

$$\Sigma = \sigma\mathbf{I}$$

Lets do a GMM notebook

Self-organizing maps

A NN that is kinda sorta like K-means but with topology

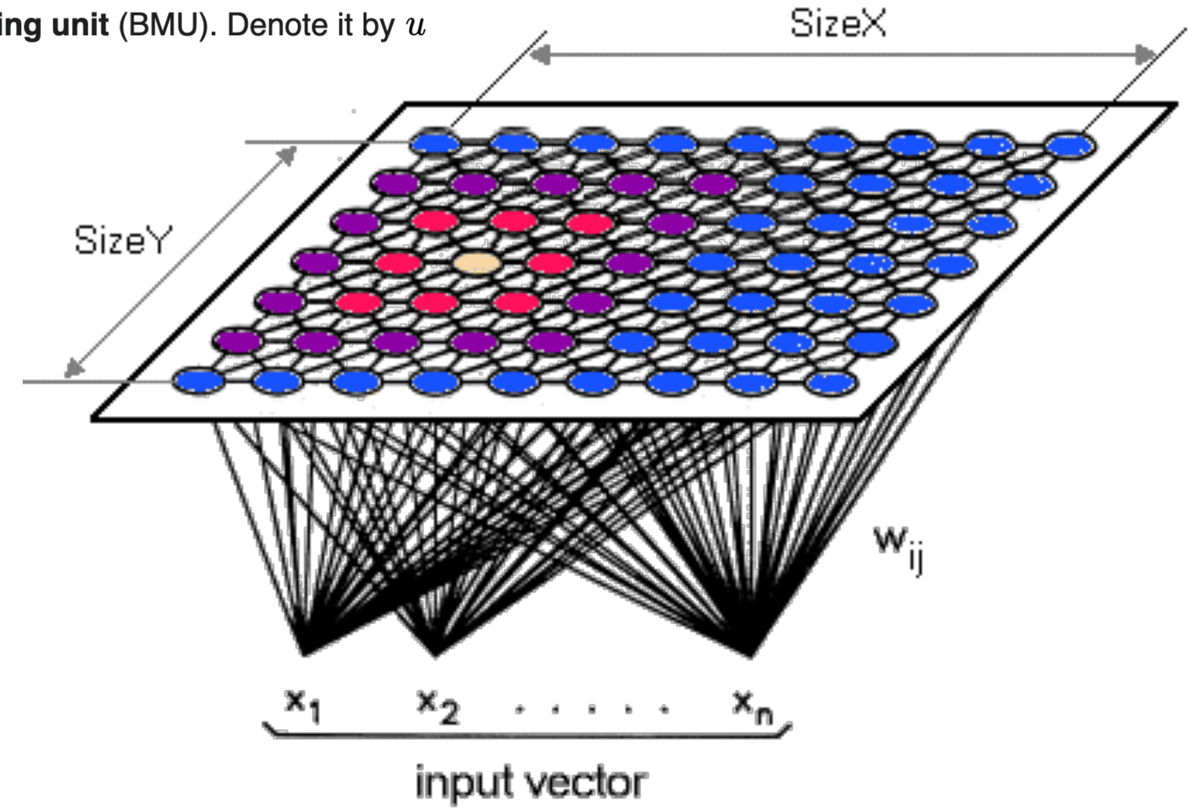
Algorithm [edit]

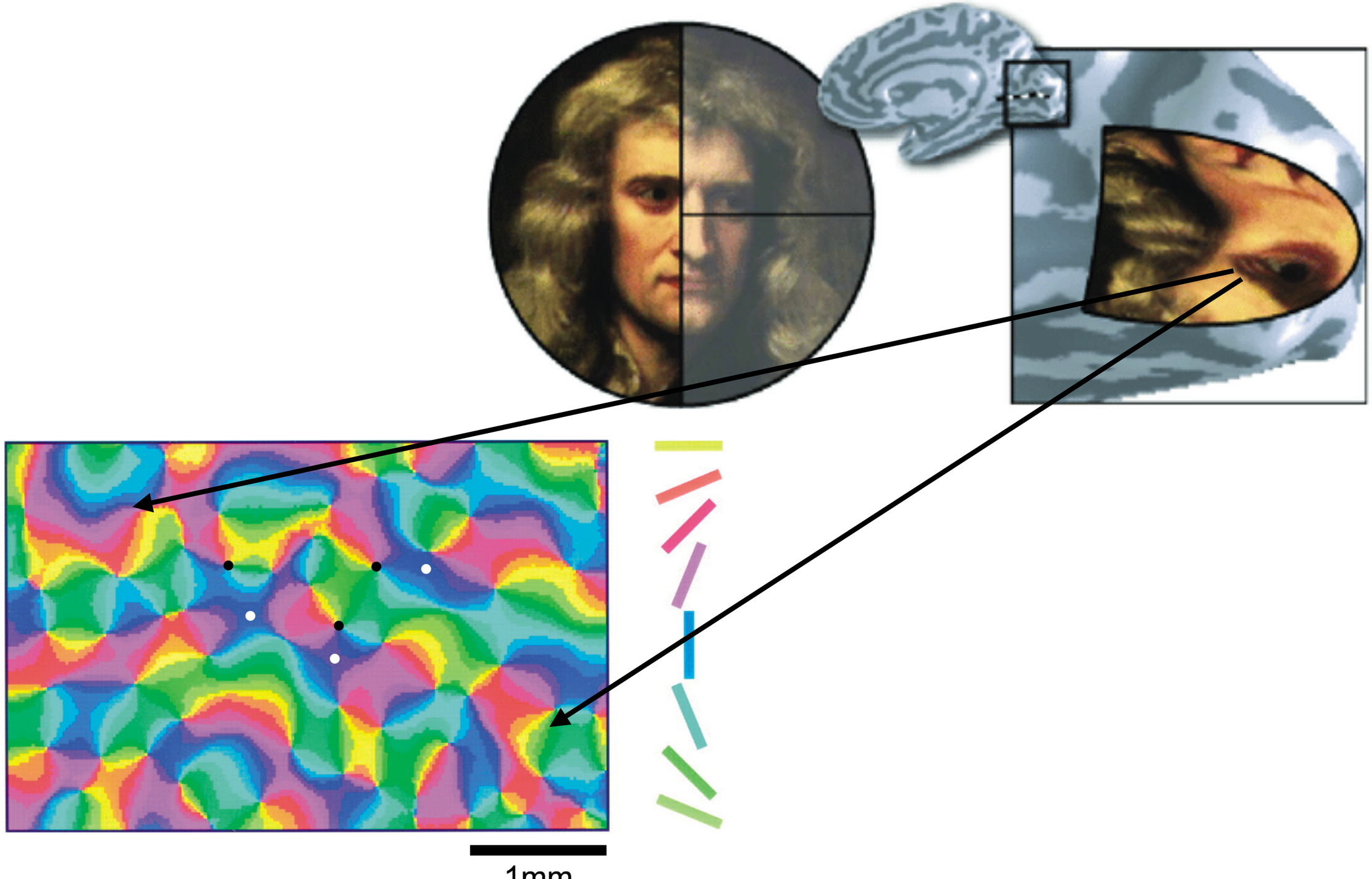
1. Randomize the node weight vectors in a map
2. For $s = 0, 1, 2, \ldots, \lambda$
    1. Randomly pick an input vector $D(t)$
    2. Find the node in the map closest to the input vector. This node is the **best matching unit** (BMU). Denote it by u
    3. For each node v, update its vector by pulling closer to the input vector:

$$W_v(s+1) = W_v(s) + \theta(u, v, s) \cdot \alpha(s) \cdot (D(t) - W_v(s))$$

The variable names mean the following, with vectors in bold,

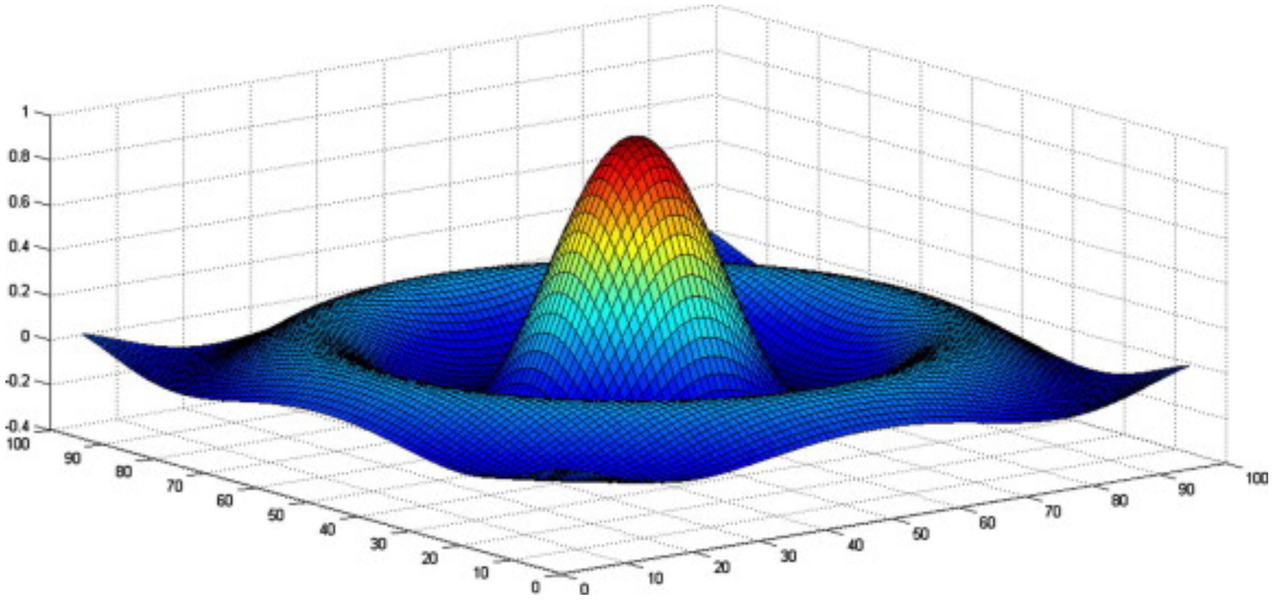
- s is the current iteration
- λ is the iteration limit
- t is the index of the target input data vector in the input data set $\mathbf{D}$
- $D(t)$ is a target input data vector
- v is the index of the node in the map
- $\mathbf{W}_v$ is the current weight vector of node v
- u is the index of the best matching unit (BMU) in the map
- $\theta(u, v, s)$ is the neighbourhood function,
- $\alpha(s)$ is the learning rate schedule.




1mm

Competitive learning

Mexican hat: a competitive neighborhood function for SOM



Algorithm 13.1 *Learning Vector Quantization—LVQ.*

1. Choose R initial prototypes for each class: $m_1(k), m_2(k), \ldots, m_R(k)$, $k = 1, 2, \ldots, K$, for example, by sampling R training points at random from each class.

2. Sample a training point x_i randomly (with replacement), and let (j, k) index the closest prototype $m_j(k)$ to x_i.

   (a) If $g_i = k$ (i.e., they are in the same class), move the prototype towards the training point:

   $$m_j(k) \leftarrow m_j(k) + \epsilon(x_i - m_j(k)),$$

   where ϵ is the *learning rate*.

   (b) If $g_i \neq k$ (i.e., they are in different classes), move the prototype away from the training point:

   $$m_j(k) \leftarrow m_j(k) - \epsilon(x_i - m_j(k)).$$

3. Repeat step 2, decreasing the learning rate ϵ with each iteration towards zero.