Lecture 2, 3: PCA, FA and EM

4F13: Machine Learning

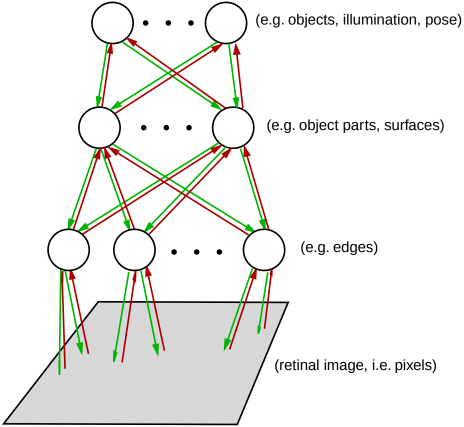
Zoubin Ghahramani and Carl Edward Rasmussen

Department of Engineering, University of Cambridge

January 23rd, 25th, 2008

Latent Variable Models

Explain correlations in **y** by assuming a generative model with latent (hidden) variables **x**


(e.g. objects, illumination, pose)
(e.g. object parts, surfaces)
(e.g. edges)
(retinal image, i.e. pixels)

$$\mathbf{x} \sim p(\mathbf{x}|\theta_x)$$

$$\mathbf{y}|\mathbf{x} \sim p(\mathbf{y}|\mathbf{x}, \theta_y)$$

$$p(\mathbf{x}, \mathbf{y}|\theta_x, \theta_y) = p(\mathbf{y}|\mathbf{x}, \theta_y)p(\mathbf{x}|\theta_x)$$

$$p(\mathbf{y}|\theta_x, \theta_y) = \int p(\mathbf{y}|\mathbf{x}, \theta_y)p(\mathbf{x}|\theta_x)d\mathbf{x}$$

Inference and Learning; latent variables and parameters

Some of the variables in our model are termed latent variables and some are called parameters, why?

- each example has separate latent values; so there are many
- we usually try to integrate out latent values

The process of finding the distribution over latent variables is called inference. The process of finding parameters is called learning.

Ideally, we would want to integrate over all unknown quantities

$$p(\mathbf{y}) \;=\; \iint p(\mathbf{y}|\mathbf{x},\theta)p(\mathbf{x}|\theta)p(\theta)d\mathbf{x}d\theta,$$

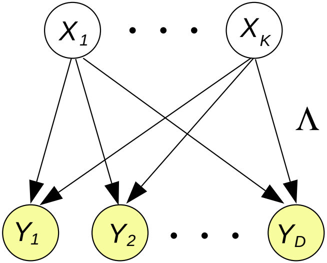
but unfortunately, this is not trivial — but we will show you how to do this approximately in a few weeks time.

Today, we concentrate on the simpler task where parameters are treated as deterministic quantities

$$p(\mathbf{y}) \;\simeq\; p(\mathbf{y}|\hat{\theta}), \;\; \text{where} \;\; \hat{\theta} = \operatorname{argmax} \int p(\mathbf{y}|\mathbf{x},\theta)p(\mathbf{x}|\theta)d\mathbf{x}.$$

Factor Analysis

Latent variable models are useful even when both latent and observed variables are Gaussian.



Linear generative model: $y_d = \sum_{k=1}^{K} \Lambda_{dk}\, x_k + \epsilon_d$

- x_k are independent $\mathcal{N}(0,1)$ Gaussian factors
- ϵ_d are independent $\mathcal{N}(0,\Psi_{dd})$ Gaussian noise
- $K < D$

So, $\mathbf{y}$ is Gaussian with: $p(\mathbf{y}) = \int p(\mathbf{x})p(\mathbf{y}|\mathbf{x})d\mathbf{x} = \mathcal{N}(0, \Lambda\Lambda^\top + \Psi)$

where Λ is a $D \times K$ matrix, and Ψ is diagonal.

Dimensionality Reduction: Finds a low-dimensional projection of high dimensional data that captures the correlation structure of the data.

Inference in the Factor Analysis Model

Given an observation, $\mathbf{y}$, what is the *distribution* of the latent cause?

Use Bayes' rule:

$$p(\mathbf{x}|\mathbf{y}) \;=\; \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \;\propto\; p(\mathbf{y}|\mathbf{x})p(\mathbf{x}).$$

In detail:

$$\begin{aligned}
p(\mathbf{x}|\mathbf{y}) \;&\propto\; p(\mathbf{x})\; p(\mathbf{y}|\mathbf{x}) \\
&=\; (2\pi)^{-K/2}\exp\big[-\tfrac{1}{2}\mathbf{x}^\top\mathbf{x}\big]\; |2\pi\Psi|^{-1/2}\exp\big[-\tfrac{1}{2}(\mathbf{y}-\Lambda\mathbf{x})^\top\Psi^{-1}(\mathbf{y}-\Lambda\mathbf{x})\big] \\
&\propto\; \exp\big(-\tfrac{1}{2}[\mathbf{x}^\top\mathbf{x}+(\mathbf{y}-\Lambda\mathbf{x})^\top\Psi^{-1}(\mathbf{y}-\Lambda\mathbf{x})]\big) \\
&\propto\; \exp\big(-\tfrac{1}{2}[\mathbf{x}^\top(I+\Lambda^\top\Psi^{-1}\Lambda)\mathbf{x}-2\mathbf{x}^\top\Lambda^\top\Psi^{-1}\mathbf{y}]\big) \\
&\propto\; \exp\big(-\tfrac{1}{2}[\mathbf{x}^\top\Sigma^{-1}\mathbf{x}-2\mathbf{x}^\top\Sigma^{-1}\boldsymbol{\mu}+\boldsymbol{\mu}^\top\Sigma^{-1}\boldsymbol{\mu}]\big) \;\propto\; \mathcal{N}(\boldsymbol{\mu},\Sigma),
\end{aligned}$$

where $\Sigma = (I+\Lambda^\top\Psi^{-1}\Lambda)^{-1} = I-\beta\Lambda$, $\boldsymbol{\mu} = \Sigma\Lambda^\top\Psi^{-1}\mathbf{y} = \beta\mathbf{y}$, and $\beta = \Sigma\Lambda^\top\Psi^{-1}$.

Note that $\boldsymbol{\mu}$ is a linear function of $\mathbf{y}$ and Σ does not depend on $\mathbf{y}$.

Learning in the Factor Analysis Model

Integrating over the latent variables, we obtained:

$$p(\mathbf{y}) \;=\; \int p(\mathbf{x})p(\mathbf{y}|\mathbf{x})d\mathbf{x} = \mathcal{N}(0, \Lambda\Lambda^\top + \Psi).$$

Note that Factor Analysis is a restricted form of Gaussian model.

The unrestricted (zero mean) Gaussian model has a closed form solution

$$p(\mathbf{y}) \;\sim\; \mathcal{N}\big(0,\; \tfrac{1}{n}\textstyle\sum_c \mathbf{y}^{(c)}\mathbf{y}^{(c)\top}\big),$$

but is no closed form solution for the Factor Analysis model.

We could try to use the gradients to do maximum likelihood wrt the parameters Λ and Ψ.

It turns out that there is a better algorithm for learning (the EM algorithm).

Eigenvalues and Eigenvectors

λ is an eigenvalue and $\mathbf{x}$ is an eigenvector of A if:

$$A\mathbf{x} = \lambda\mathbf{x}$$

and $\mathbf{x}$ is a unit vector ($\mathbf{x}^\top\mathbf{x} = 1$).

Interpretation: the operation of A in direction $\mathbf{x}$ is a scaling by λ.

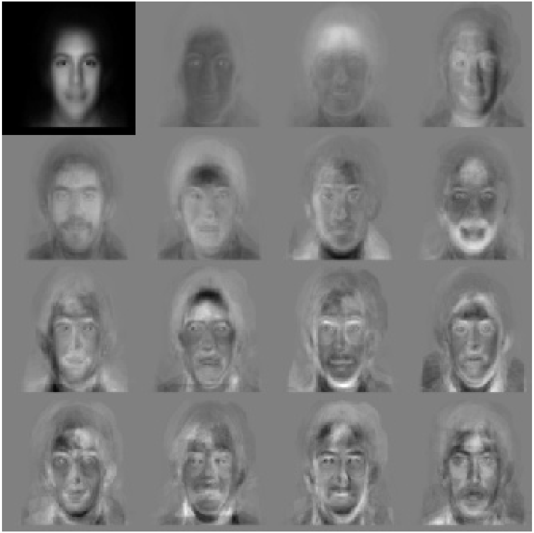
The K Principal Components are the K eigenvectors with the largest eigenvalues of the data covariance matrix (i.e. K directions with the largest variance).

Note: Σ can be decomposed:

$$\Sigma = USU^\top,$$

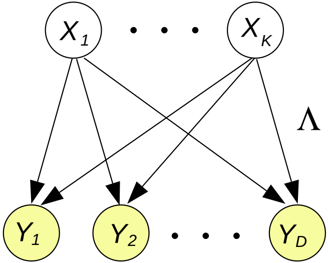
where S is $\text{diag}(\sigma_1^2, \ldots, \sigma_D^2)$ and U is a an orthonormal matrix.

Example of PCA: Eigenfaces



from `http://vismod.media.mit.edu/vismod/demos/facerec/basic.html`

Principal Components Analysis (PCA)



Noise variable becomes infinitesimal compared to the scale of the data:
$\Psi = \lim_{\sigma^2 \to 0} \sigma^2 I$

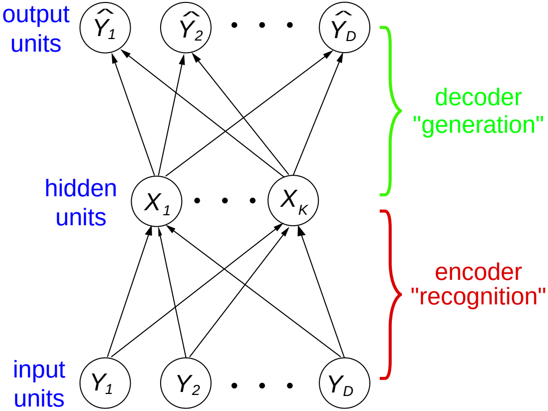
Equivalently: reconstruction cost becomes infinite compared to the cost of coding the hidden units under the prior.

$$p(\mathbf{x}|\mathbf{y}) \;=\; \mathcal{N}(\beta\mathbf{y}, I - \beta\Lambda),$$

where

$$\beta \;=\; \lim_{\sigma^2 \to 0} \Lambda^\top(\Lambda\Lambda^\top + \sigma^2 I)^{-1} \;=\; (\Lambda^\top\Lambda)^{-1}\Lambda^\top.$$

From Supervised Learning to PCA: linear autoencoder

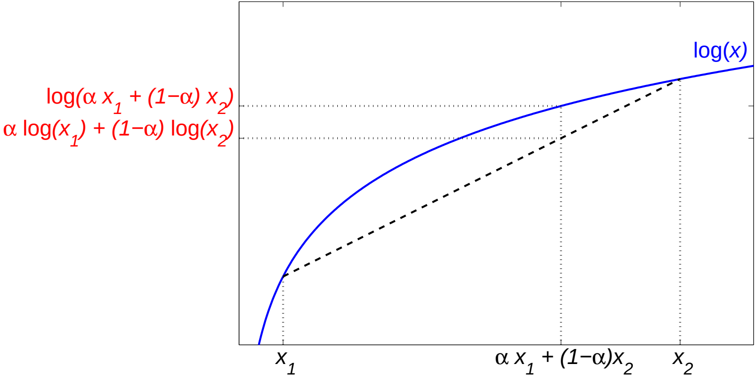


A linear autoencoder neural network trained to minimise squared error learns to perform PCA (Baldi & Hornik, 1989).

FA vs PCA

- PCA is rotationally invariant; FA is not
- FA is measurement scale invariant; PCA is not
- FA defines a probabilistic model; PCA does not
- PCA can be computed in closed form; FA can not

Jensen's Inequality



For $\alpha_i \geqslant 0$, $\sum \alpha_i = 1$ and any $\{x_i > 0\}$

$$\log\Big(\sum_i \alpha_i x_i\Big) \;\geqslant\; \sum_i \alpha_i \log(x_i)$$

Equality if and only if $\alpha_i = 1$ for some i (and therefore all others are 0).

The Expectation Maximization (EM) algorithm

Given a set of observed (visible) variables V, a set of unobserved (hidden / latent / missing) variables H, and model parameters θ, optimize the log likelihood:

$$\mathcal{L}(\theta) \;=\; \log p(V|\theta) \;=\; \log \int p(H, V|\theta) dH, \tag{1}$$

where we have written the marginal for the visibles in terms of an integral over the joint distribution for hidden and visible variables.

Using *Jensen's inequality* for any distribution of hidden states $q(H)$ we have:

$$\mathcal{L} \;=\; \log \int q(H) \frac{p(H, V|\theta)}{q(H)} dH \;\geqslant\; \int q(H) \log \frac{p(H, V|\theta)}{q(H)} dH \;=\; \mathcal{F}(q, \theta), \tag{2}$$

defining the $\mathcal{F}(q, \theta)$ functional, which is a lower bound on the log likelihood.

In the EM algorithm, we alternately optimize $\mathcal{F}(q, \theta)$ wrt q and θ, and we can prove that this will never decrease $\mathcal{L}$.

The E and M steps of EM

The lower bound on the log likelihood:

$$\mathcal{F}(q,\theta) \;=\; \int q(H)\log\frac{p(H,V|\theta)}{q(H)}dH \;=\; \int q(H)\log p(H,V|\theta)dH + \mathcal{H}(q), \qquad (3)$$

where $\mathcal{H}(q) = -\int q(H)\log q(H)dH$ is the entropy of q. We iteratively alternate:

E step: optimize $\mathcal{F}(q,\theta)$ wrt the distribution over hidden variables given the parameters:

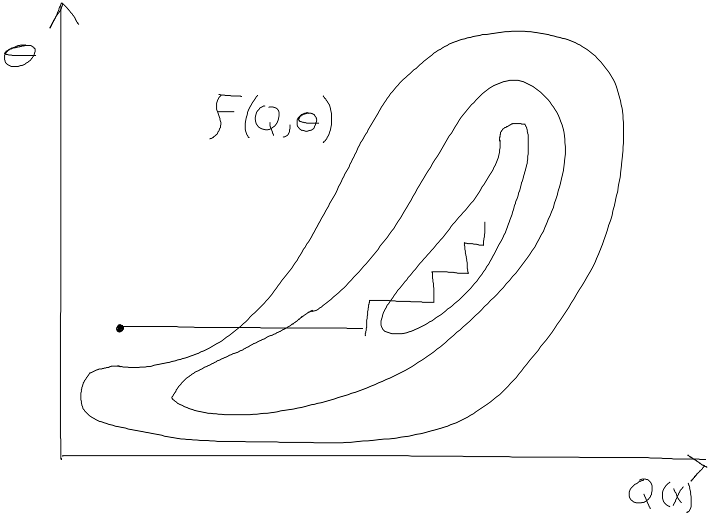
$$q^{(k)}(H) := \underset{q(H)}{\operatorname{argmax}}\; \mathcal{F}\big(q(H), \theta^{(k-1)}\big). \qquad (4)$$

M step: maximize $\mathcal{F}(q,\theta)$ wrt the parameters given the hidden distribution:

$$\theta^{(k)} := \underset{\theta}{\operatorname{argmax}}\; \mathcal{F}\big(q^{(k)}(H), \theta\big) = \underset{\theta}{\operatorname{argmax}}\; \int q^{(k)}(H)\log p(H,V|\theta)dH, \qquad (5)$$

which is equivalent to optimizing the expected complete-data likelihood $p(H,V|\theta)$, since the entropy of $q(H)$ does not depend on θ.

EM as Coordinate Ascent in $\mathcal{F}$



The EM algorithm never decreases the log likelihood

The difference between the cost functions:

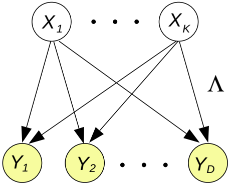
$$\begin{aligned}\mathcal{L}(\theta) - \mathcal{F}(q,\theta) &= \log p(V|\theta) - \int q(H) \log \frac{p(H,V|\theta)}{q(H)} dH \\ &= \log p(V|\theta) - \int q(H) \log \frac{p(H|V,\theta)p(V|\theta)}{q(H)} dH \\ &= -\int q(H) \log \frac{p(H|V,\theta)}{q(H)} dH \;=\; \mathcal{KL}\big(q(H), p(H|V,\theta)\big),\end{aligned}$$

is called the Kullback-Liebler divergence; it is non-negative and only zero if and only if $q(H) = p(H|V,\theta)$ (thus this is the E step). Although we are working with the wrong cost function, the likelihood is still increased in every iteration:

$$\mathcal{L}\big(\theta^{(k-1)}\big) \underset{\text{E step}}{=} \mathcal{F}\big(q^{(k)}, \theta^{(k-1)}\big) \underset{\text{M step}}{\leqslant} \mathcal{F}\big(q^{(k)}, \theta^{(k)}\big) \underset{\text{Jensen}}{\leqslant} \mathcal{L}\big(\theta^{(k)}\big),$$

where the first equality holds because of the E step, and the first inequality comes from the M step and the final inequality from Jensen. Usually EM converges to a local optimum of $\mathcal{L}$ (although there are exceptions).

EM for Factor Analysis



The model for $\mathbf{y}$:
$p(\mathbf{y}|\theta) = \int p(\mathbf{x}|\theta)p(\mathbf{y}|\mathbf{x},\theta)d\mathbf{x} = \mathcal{N}(0, \Lambda\Lambda^\top + \Psi)$
Model parameters: $\theta = \{\Lambda, \Psi\}$.

E step: For each data point $\mathbf{y}_c$, compute the posterior distribution of hidden factors given the observed data: $q_c(\mathbf{x}_c) = p(\mathbf{x}_c|\mathbf{y}_c, \theta^{(t)})$.

M step: Find the $\theta^{(t+1)}$ that maximises $\mathcal{F}(q, \theta)$:

$$\begin{aligned}
\mathcal{F}(q,\theta) &= \sum_c \int q_c(\mathbf{x}_c)\Big[\log p(\mathbf{x}_c|\theta) + \log p(\mathbf{y}_c|\mathbf{x}_c,\theta) - \log q_c(\mathbf{x}_c)\Big]d\mathbf{x}_c \\
&= \sum_c \int q_c(\mathbf{x})\Big[\log p(\mathbf{x}_c|\theta) + \log p(\mathbf{y}_c|\mathbf{x}_c,\theta)\Big]d\mathbf{x} + \text{const.}
\end{aligned}$$

The M step for Factor Analysis

M step: Find $\theta^{(t+1)}$ maximising $\mathcal{F} = \sum_c \int q_c(\mathbf{x})\,[\log p(\mathbf{x}|\theta) + \log p(\mathbf{y}_c|\mathbf{x},\theta)]\,d\mathbf{x}$.

$$\begin{aligned}
&\log p(\mathbf{x}|\theta) + \log p(\mathbf{y}_c|\mathbf{x},\theta) \\
&\quad= \text{const} - \tfrac{1}{2}\mathbf{x}_c^\top\mathbf{x}_c - \tfrac{1}{2}\log|\Psi| - \tfrac{1}{2}(\mathbf{y}_c - \Lambda\mathbf{x}_c)^\top\Psi^{-1}(\mathbf{y}_c - \Lambda\mathbf{x}_c) \\
&\quad= \text{const'} - \tfrac{1}{2}\log|\Psi| - \tfrac{1}{2}[\mathbf{y}_c^\top\Psi^{-1}\mathbf{y}_c - 2\mathbf{y}_c^\top\Psi^{-1}\Lambda\mathbf{x}_c + \mathbf{x}_c^\top\Lambda^\top\Psi^{-1}\Lambda\mathbf{x}_c] \\
&\quad= \text{const'} - \tfrac{1}{2}\log|\Psi| - \tfrac{1}{2}[\mathbf{y}_c^\top\Psi^{-1}\mathbf{y}_c - 2\mathbf{y}_c^\top\Psi^{-1}\Lambda\mathbf{x}_c + \text{trace}\,\Lambda^\top\Psi^{-1}\Lambda\mathbf{x}_c\mathbf{x}_c^\top]
\end{aligned}$$

Taking expectations over $q_c(\mathbf{x}_c)$...

$$= \text{const'} - \tfrac{1}{2}\log|\Psi| - \tfrac{1}{2}\left[\mathbf{y}_c^\top\Psi^{-1}\mathbf{y}_c - 2\mathbf{y}_c^\top\Psi^{-1}\Lambda\boldsymbol{\mu}_c + \text{trace}\,\Lambda^\top\Psi^{-1}\Lambda(\boldsymbol{\mu}_c\boldsymbol{\mu}_c^\top + \Sigma)\right].$$

Note that we don't need to know everything about q, just the expectations of $\mathbf{x}$ and $\mathbf{x}\mathbf{x}^\top$ under q (i.e. the expected sufficient statistics).

The M step for Factor Analysis (cont.)

$$\mathcal{F} = \text{const'} - \frac{N}{2}\log|\Psi| - \tfrac{1}{2}\sum_c \left[\mathbf{y}_c^\top \Psi^{-1}\mathbf{y}_c - 2\mathbf{y}_c^\top\Psi^{-1}\Lambda\boldsymbol{\mu}_c + \text{trace}(\Lambda^\top\Psi^{-1}\Lambda(\boldsymbol{\mu}_c\boldsymbol{\mu}_c^\top + \Sigma))\right]$$

Taking derivatives w.r.t. Λ and Ψ^{-1}, using $\frac{\partial\,\text{trace}\,AB}{\partial B} = A^\top$ and $\frac{\partial \log|A|}{\partial A} = A^{-\top}$:

$$\frac{\partial\mathcal{F}}{\partial\Lambda} = \Psi^{-1}\sum_c \mathbf{y}_c\boldsymbol{\mu}_c^\top - \Psi^{-1}\Lambda\Big(N\Sigma + \sum_c \boldsymbol{\mu}_c\boldsymbol{\mu}_c^\top\Big) = 0$$

$$\Lambda = \Big(N\Sigma + \sum_c \boldsymbol{\mu}_c\boldsymbol{\mu}_c^\top\Big)^{-1}\sum_c \mathbf{y}_c\boldsymbol{\mu}_c^\top$$

$$\frac{\partial\mathcal{F}}{\partial\Psi^{-1}} = \frac{N}{2}\Psi - \frac{1}{2}\sum_c\left[\mathbf{y}_c\mathbf{y}_c^\top - \Lambda\boldsymbol{\mu}_c\mathbf{y}_c^\top - \mathbf{y}_c\boldsymbol{\mu}_c^\top\Lambda^\top + \Lambda(\boldsymbol{\mu}_c\boldsymbol{\mu}_c^\top + \Sigma)\Lambda^\top\right]$$

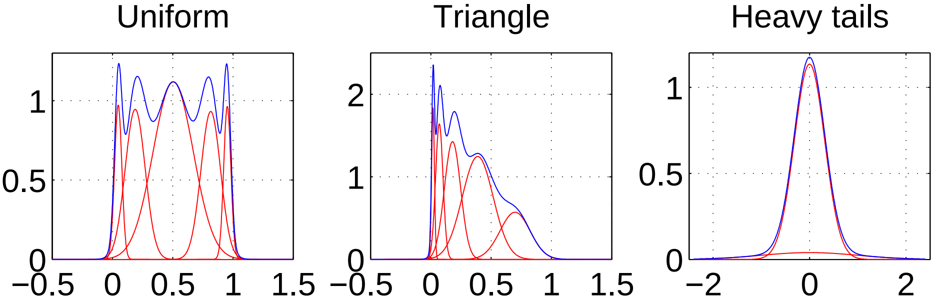
$$\Psi = \frac{1}{N}\sum_c\left[\mathbf{y}_c\mathbf{y}_c^\top - \Lambda\boldsymbol{\mu}_c\mathbf{y}_c^\top - \mathbf{y}_c\boldsymbol{\mu}_c^\top\Lambda^\top + \Lambda(\boldsymbol{\mu}_c\boldsymbol{\mu}_c^\top + \Sigma)\Lambda^\top\right]$$

$$= \Lambda\Sigma\Lambda^\top + \frac{1}{N}\sum_c(\mathbf{y}_c - \Lambda\boldsymbol{\mu}_c)(\mathbf{y}_c - \Lambda\boldsymbol{\mu}_c)^\top \qquad \text{(squared residual s)}$$

Note: we should actually only take derivarives w.r.t. Ψ_{dd} since Ψ is diagonal.
When $\Sigma \to 0$ these become the equations for linear regression!

Limitations and Mixtures

So far, we have assumed that the data is reasonably well approximated by a Gaussian.



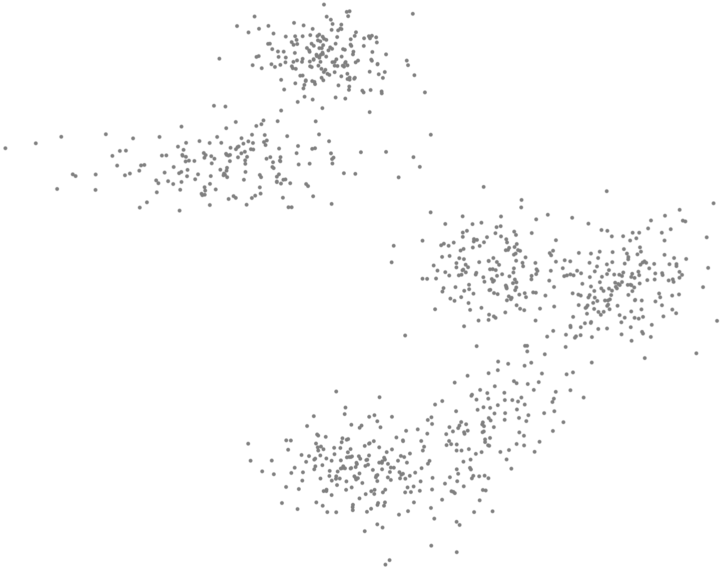
A mixture distribution has a single discrete latent variable:

$$s_i \sim \text{Discrete}[\boldsymbol{\pi}] \qquad \mathbf{y}_i | s_i \sim p(\mathbf{y}_i | \theta_{s_i})$$

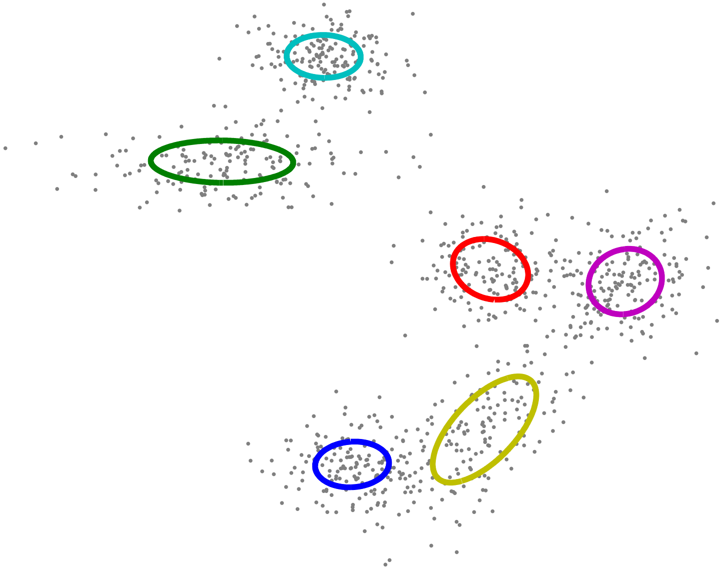
Mixtures arise naturally when observations from different sources have been collated.

They can also be used to *approximate* arbitrary distributions.

Clustering with MoG



Clustering with MoG



Likelihood for the Mixture of Gaussians model

Likelihood:

$$
\begin{aligned}
p(\mathbf{y}_1, \ldots, \mathbf{y}_n | \theta) &= \prod_{c=1}^{n} \sum_{i=1}^{k} \pi_i \, \mathcal{N}(\mathbf{y}_c | \boldsymbol{\mu}_i, \Sigma_i) \\
&= \prod_{c=1}^{n} \sum_{i=1}^{k} \pi_i \, |2\pi\Sigma_i|^{-1/2} \exp\big(- (\mathbf{y}_c - \boldsymbol{\mu}_i)^\top \Sigma_i^{-1} (\mathbf{y}_c - \boldsymbol{\mu}_i)\big).
\end{aligned}
$$

Here, π_i are the mixing proportions, (non-negative, sum to one) and the parameters are collected in $\theta = (\boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k, \Sigma_1, \ldots, \Sigma_k)$.

Mixture of Gaussians, E-step

The likelihood (for a single data point) in the mixture of Gaussians model is

$$p(\mathbf{y}|\theta) \; = \; \sum_{j=1}^{k} p(\mathbf{y}, h|\theta) \; = \; \sum_{j=1}^{k} p(h=j|\theta) p(\mathbf{y}|h=j,\theta),$$

where $\theta = (\pi_1, \ldots, \pi_k, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k, \Sigma_1, \ldots, \Sigma_k)$ are the parameters, and h is the hidden, or latent, variable.

In the E-step we maximize the lower bound functional $\mathcal{F}(q(H), \theta)$ wrt $q(H)$ for fixed θ. We have seen that this is equivalent to setting $q(H)$ equal to the posterior:

$$q(H) \; = \; p(h|\mathbf{y},\theta) \; = \; \frac{p(\mathbf{y}|h,\theta)p(h|\theta)}{p(\mathbf{y}|\theta)} \; \propto \; p(\mathbf{y}|h,\theta)p(h|\theta).$$

Thus, the *responsibilities* are:

$$r_j \; = \; p(h=j|\mathbf{y},\theta) \; \propto \; \pi_j |\Sigma_j|^{1/2} \exp\Big(-(\mathbf{y}-\boldsymbol{\mu}_j)^\top \Sigma_j^{-1}(\mathbf{y}-\boldsymbol{\mu}_j)/2\Big),$$

normalized such that $\sum_j r_j = 1$. For multiple data points, responsibilities are computed analogously: $r_{cj} = p(h_c = j|\mathbf{y}_c, \theta)$.

Mixture of Gaussians, M-step

In the M-step, we maximize the lower bound functional $\mathcal{F}(q(H), \theta)$ wrt. θ. Recall, that this is equivalent to maximizing

$$E(\theta) \;=\; \int q(H) \log \big(p(H, V|\theta)\big) dH \;=\; \int q(H) \sum_c \log \big(p(H_c, V_c|\theta)\big) dH,$$

for a fixed $q(H)$. For mixtures of Gaussians:

$$E(\theta) \;=\; \sum_{c,j} r_{cj} \Big[\log(\pi_j) - \tfrac{1}{2} \log |\Sigma_j| - (\mathbf{y}_c - \boldsymbol{\mu}_j)^\top \Sigma_j^{-1} (\mathbf{y}_c - \boldsymbol{\mu}_j)/2 \Big].$$

Mixture of Gaussians, M-step, cont.

Optimizing wrt the θ parameters (and $S_j = \Sigma_j^{-1}$):

$$\frac{\partial E(\theta)}{\partial \boldsymbol{\mu}_j} = \sum_c r_{cj}\Sigma_j^{-1}(\mathbf{y}_c - \boldsymbol{\mu}) = \mathbf{0} \implies \boldsymbol{\mu}_j = \sum_c r_{cj}\mathbf{y}_c/(\sum_c r_{cj})$$

$$\frac{\partial E(\theta)}{\partial S_j} = \frac{1}{2}\sum_c r_{cj}\Big[S_j^{-1} - (\mathbf{y}_c - \boldsymbol{\mu}_j)(\mathbf{y}_c - \boldsymbol{\mu}_j)^\top\Big] = \mathbf{0}$$

$$\implies \Sigma_j = \sum_{c=1}^{n} r_{cj}(\mathbf{y}_c - \boldsymbol{\mu}_j)(\mathbf{y}_c - \boldsymbol{\mu}_j)^\top/(\sum_c r_{cj}),$$

$$\frac{\partial E(\theta)}{\partial \pi_j} = \sum_c \frac{r_{cj}}{\pi_j} - \lambda = 0 \implies \pi_j = \sum_c r_{cj}/(\sum_{c,j} r_{cj}),$$

where λ is the Lagrange multiplier which ensures that the mixture normalizes $\sum_j \pi_j = 1$.

Mixture of Gaussian issues

The EM algorithm converges to a *local* maximum of the likelihood. There may be many local maxima.

There could be many bad local maxima, ie with low values of the likelihood.

In fact, we are not interested in the global optimum!

Another problem: “How many mixture components?”