4F13: Machine Learning

Lecture 3-4: Unsupervised Learning

Zoubin Ghahramani
zoubin@eng.cam.ac.uk

Department of Engineering
University of Cambridge

Michaelmas, 2006

http://learning.eng.cam.ac.uk/zoubin/ml06/

Key Ingredients of Machine Learning

Data

Let $\mathbf{y} = (y_1, y_2, \ldots, y_D)$ denote a data point, and $\mathcal{D} = \{\mathbf{y}_1, \mathbf{y}_2 \ldots, \mathbf{y}_N\}$, a data set

Predictions

We are generally interested in predicting something based on the observed data set.

Given $\mathcal{D}$ what can we say about $\mathbf{y}_{N+1}$?

Given $\mathcal{D}$ and $y_{N+1,1}, y_{N+1,2}, \ldots, y_{N+1,D-1}$, what can we say about $y_{N+1,D}$?

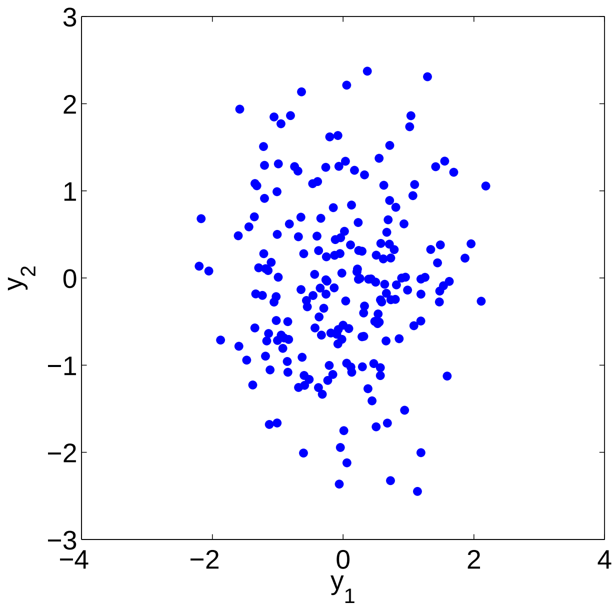
Model

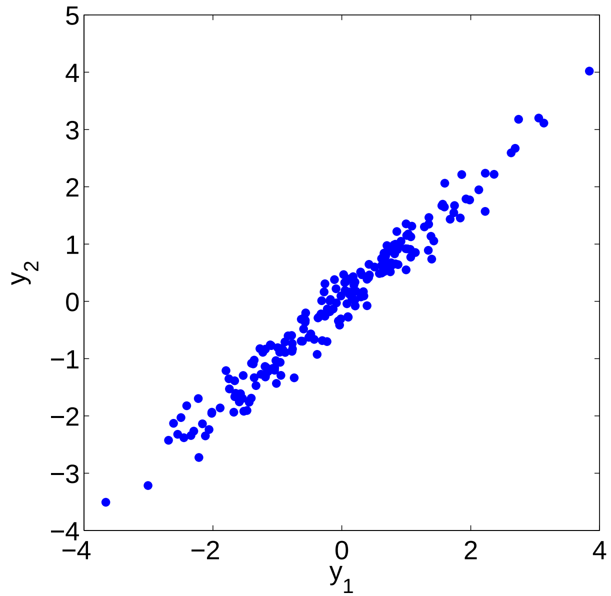
To make predictions, we need to make some *assumptions*. We can often express these assumptions in the form of a model, with some parameters, $\boldsymbol{\theta}$

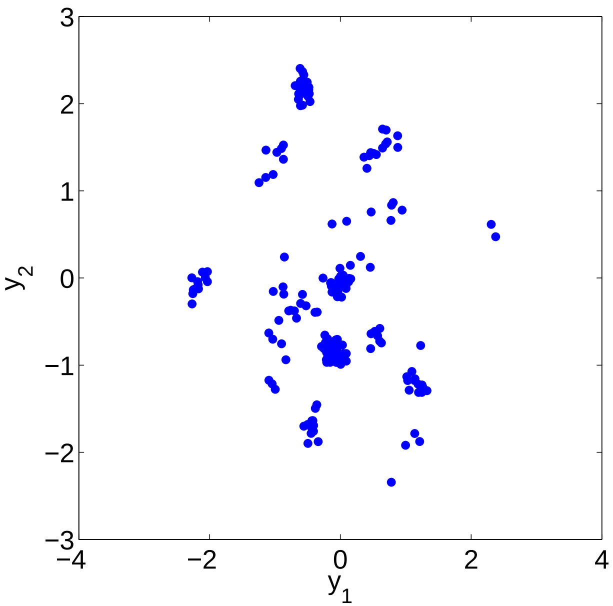
Given data $\mathcal{D}$, we learn the model parameters $\boldsymbol{\theta}$, from which we can predict new data points.

The model can often be expressed as a *probability distribution over data points*

A few simple data sets

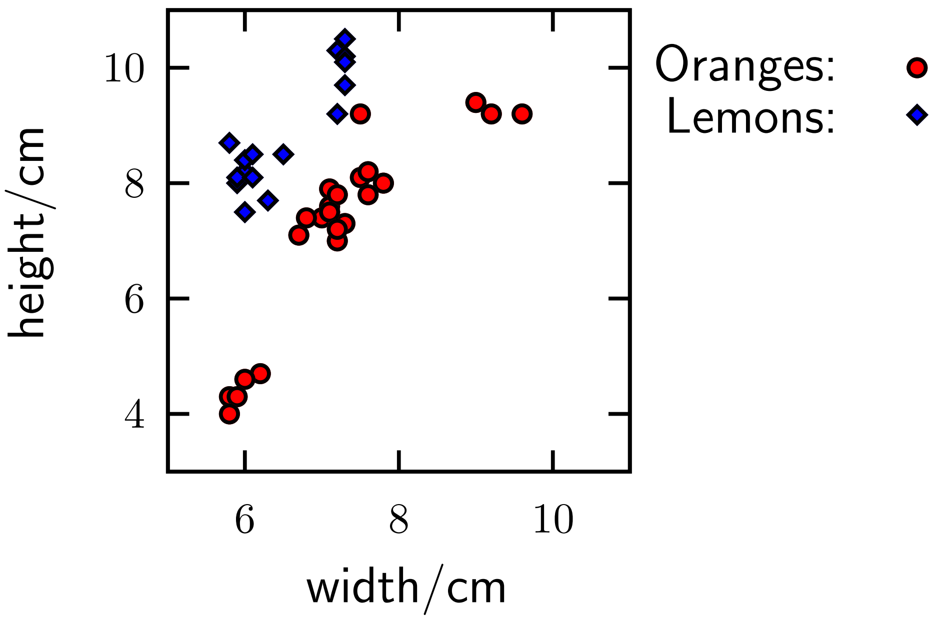






A more interesting data set:

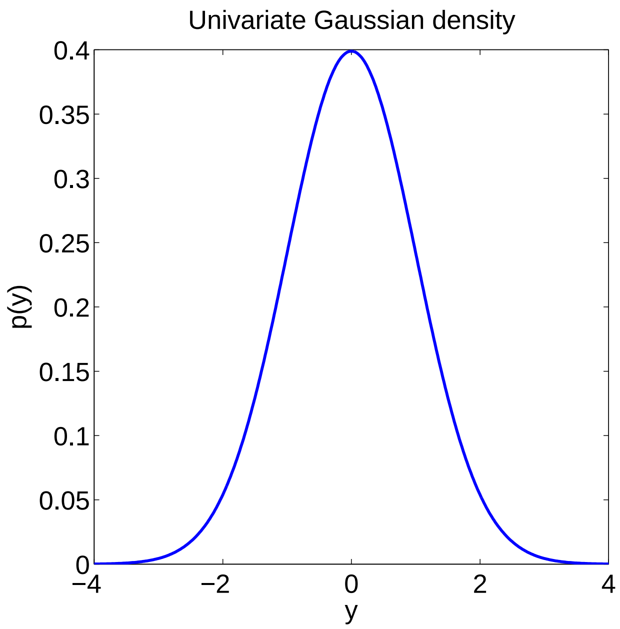
Here $D = 2$, $\mathbf{y} \in \mathbb{R}^2$.



A very simple model

Univariate Gaussian density ($y \in \mathbb{R}$):

$$p(y|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(y-\mu)^2}{2\sigma^2}\right\}$$



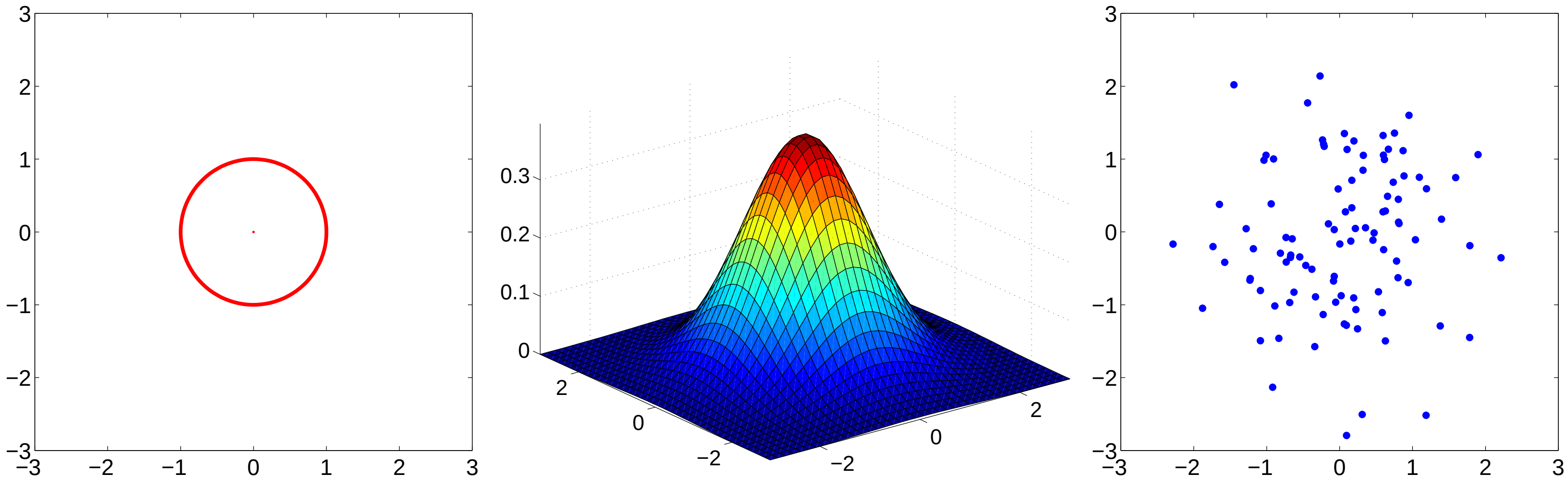
This model has parameters $\boldsymbol{\theta} = \{\mu, \sigma\}$ which model the mean and standard deviation of the data, respectively.

A slighly more complicated model

Multivariate Gaussian density ($\mathbf{y} \in \mathbb{R}^D$):

$$p(\mathbf{y}|\boldsymbol{\mu}, \Sigma) = |2\pi\Sigma|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{y}-\boldsymbol{\mu}) \right\}$$

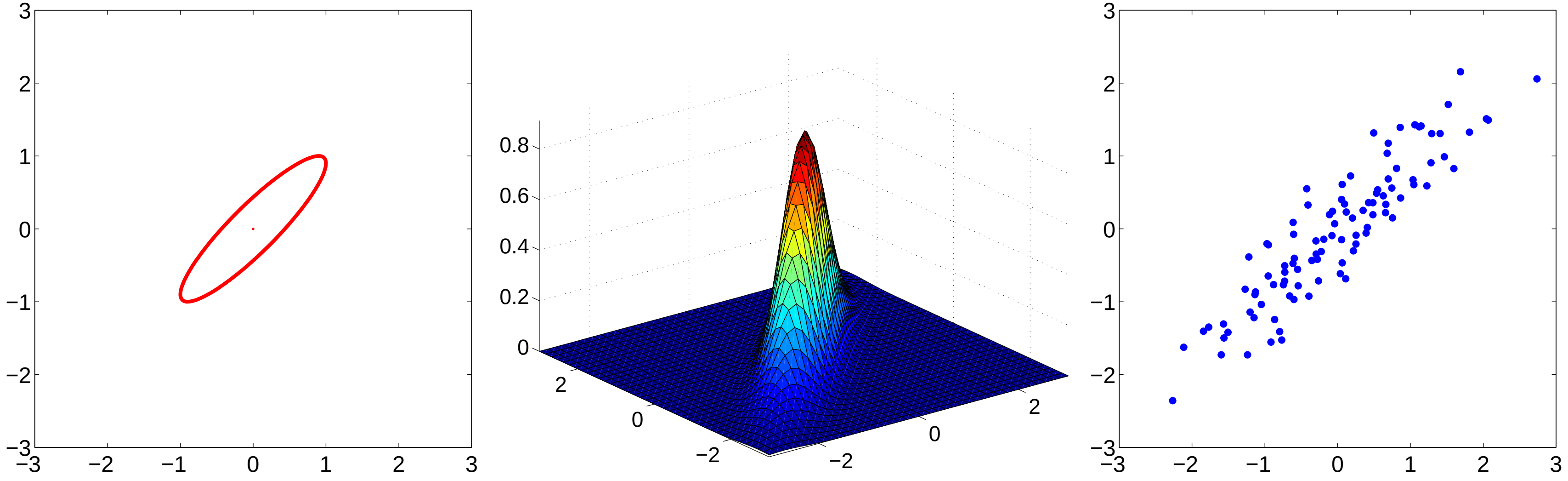
$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



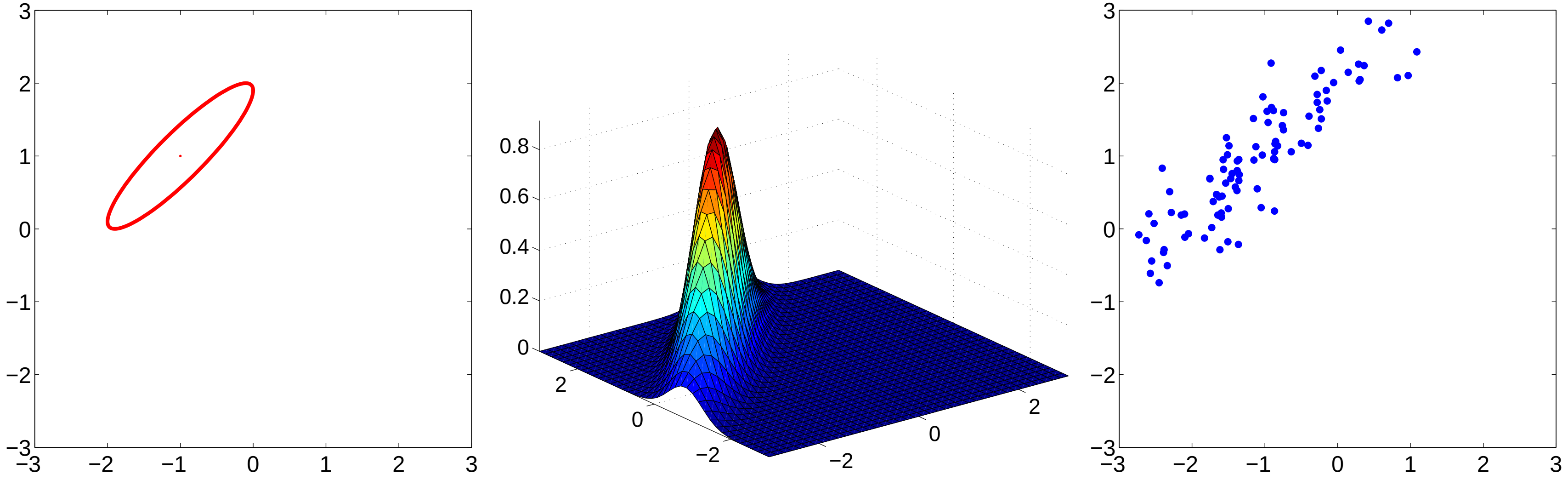
This model has parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \Sigma\}$ which model the mean and covariance matrix of the data.

The multivariate Gaussian density

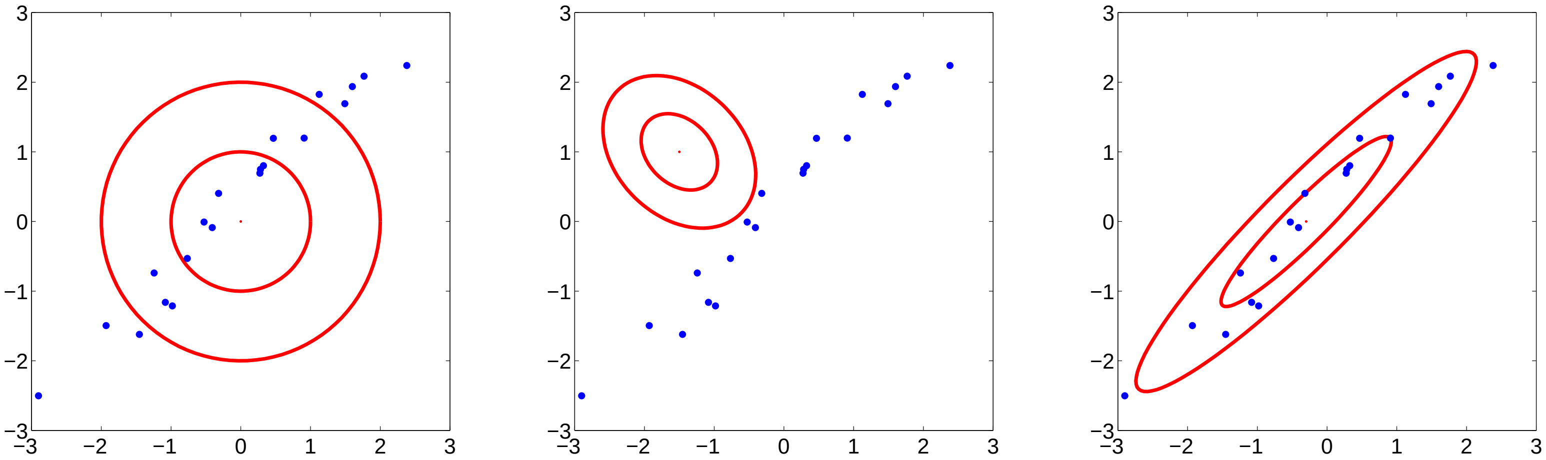
$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$$



$$\boldsymbol{\mu} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$$



Fitting the model to data



Assume the data were generated independently from the model.
We can measure the likelihood of the model:

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\mathbf{y}_n|\boldsymbol{\theta})$$

Clearly, the third model is a better fit to the data than the others:

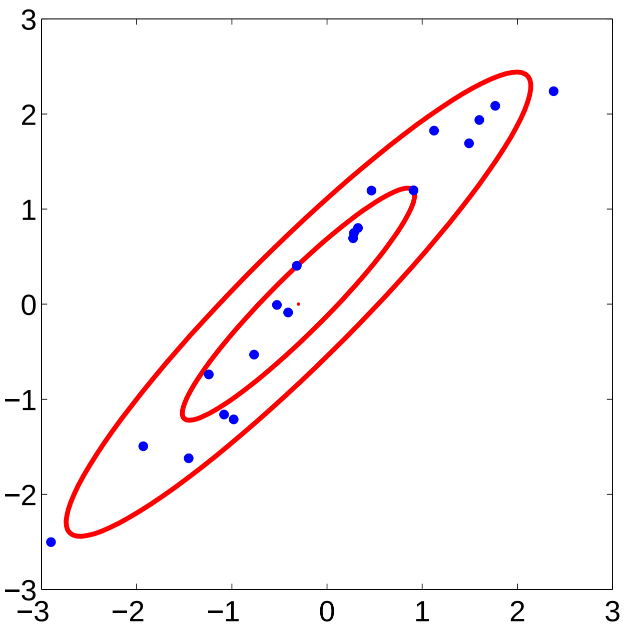
$$\begin{aligned}
\log p(\mathcal{D}|\boldsymbol{\theta}_1) &= -55.38 \\
\log p(\mathcal{D}|\boldsymbol{\theta}_2) &= -238.29 \\
\log p(\mathcal{D}|\boldsymbol{\theta}_2) &= -22.14
\end{aligned}$$

The likelihood function

Data set $\mathcal{D} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, the likelihood: $p(\mathcal{D}|\boldsymbol{\mu}, \Sigma) = \prod_{n=1}^{N} p(\mathbf{y}_n|\boldsymbol{\mu}, \Sigma)$ is a function of the model parameters

The maximum likelihood (ML) procedure finds parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \Sigma\}$ such that:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \mathrm{argmax}_{\boldsymbol{\theta}}\, p(\mathcal{D}|\boldsymbol{\theta})$$



Finding Maximum Likelihood Estimate for a Gaussian

Data set $\mathcal{D} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, likelihood: $p(\mathcal{D}|\boldsymbol{\mu}, \Sigma) = \prod_{n=1}^{N} p(\mathbf{y}_n|\boldsymbol{\mu}, \Sigma)$

Maximise likelihood $\Leftrightarrow$ maximise log likelihood

Goal: find $\boldsymbol{\mu}$ and Σ that maximise log likelihood:

$$\begin{aligned}\mathcal{L} = \log \prod_{n=1}^{N} p(\mathbf{y}_n|\boldsymbol{\mu}, \Sigma) &= \sum_n \log p(\mathbf{y}_n|\boldsymbol{\mu}, \Sigma) \\ &= -\frac{N}{2} \log |2\pi\Sigma| - \frac{1}{2} \sum_n (\mathbf{y}_n - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{y}_n - \boldsymbol{\mu})\end{aligned}$$

Note: equivalently, minimise $-\mathcal{L}$, which is *quadratic* in $\boldsymbol{\mu}$

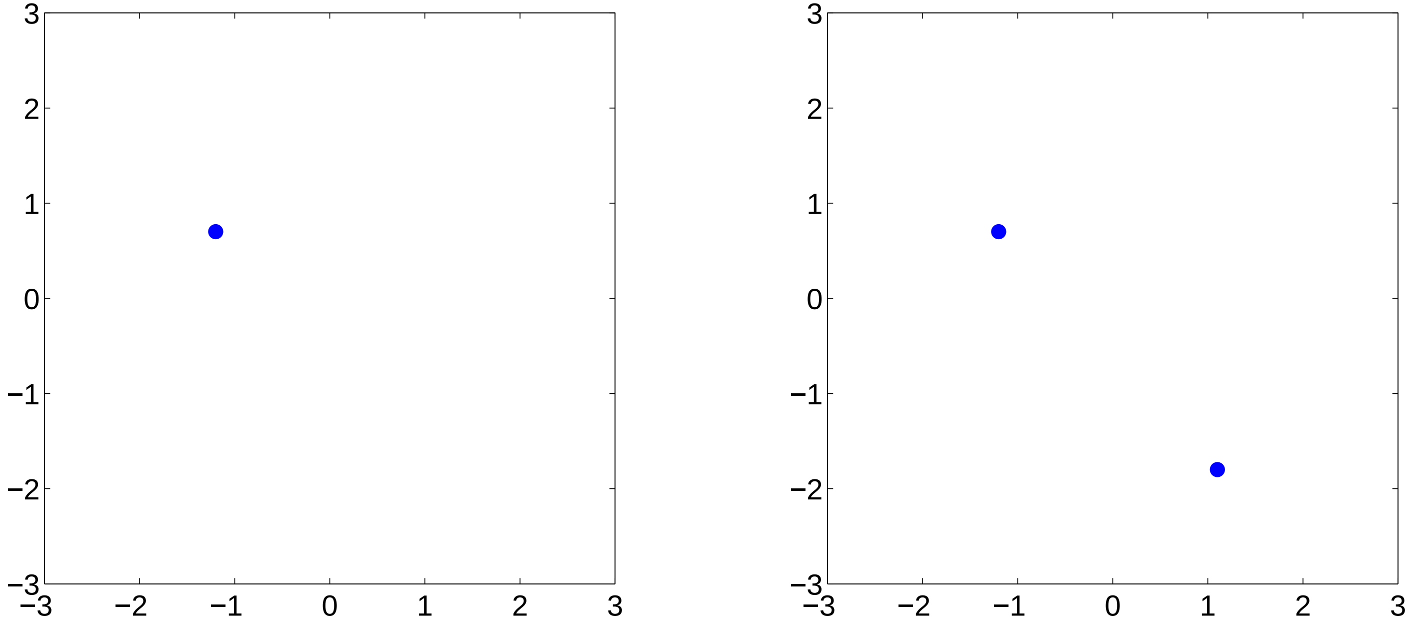
Procedure: take derivatives and set to zero:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = 0 \quad \Rightarrow \quad \hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_n \mathbf{y}_n \quad \text{(sample mean)}$$

$$\frac{\partial \mathcal{L}}{\partial \Sigma} = 0 \quad \Rightarrow \quad \hat{\Sigma} = \frac{1}{N} \sum_n (\mathbf{y}_n - \hat{\boldsymbol{\mu}})(\mathbf{y}_n - \hat{\boldsymbol{\mu}})^\top \quad \text{(sample covariance)}$$

Two *very* simple data sets

What are the maximum likelihood estimates of θ for these data sets?



Does this make sense?

Bayesian Learning

Apply the basic rules of probability to learning from data.
Use probability distributions to represent uncertainty.

Data set: $\mathcal{D} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$
Model parameters: $\boldsymbol{\theta}$

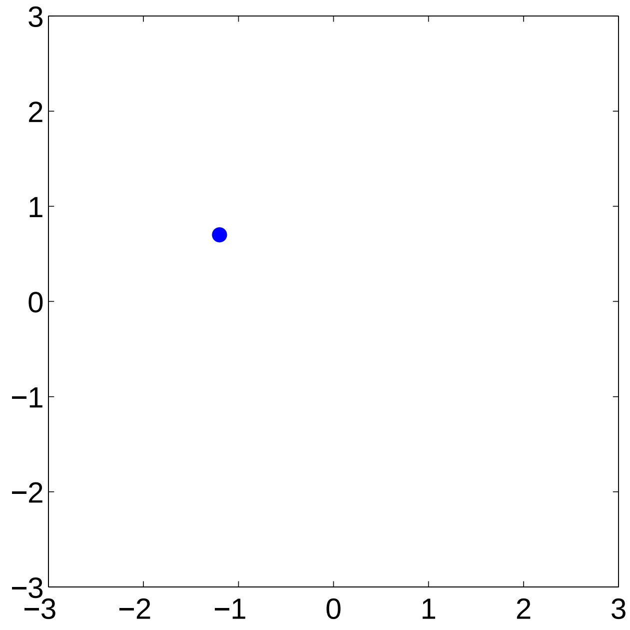
Prior probabilities of model parameters: $P(\boldsymbol{\theta})$
Model of data given parameters (likelihood model): $P(\mathbf{y}|\boldsymbol{\theta})$

If the data are independently and identically distributed then:

$$P(\mathcal{D}|\boldsymbol{\theta}) = \prod_{n=1}^{N} P(\mathbf{y}_n|\boldsymbol{\theta})$$

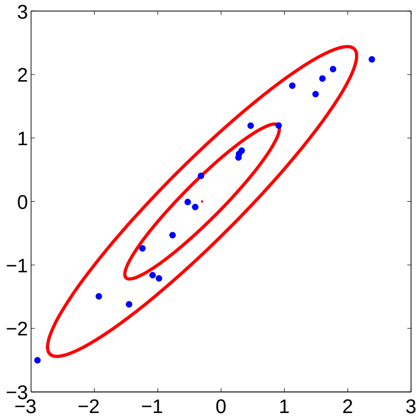
Posterior probability of model parameters:

$$P(\boldsymbol{\theta}|\mathcal{D}) = \frac{P(\mathcal{D}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\mathcal{D})}$$



Limitations of the Multivariate Gaussian

Gaussians are fundamental and widespread, but not every distribution of interest is Gaussian.



- Some processes produce outliers.

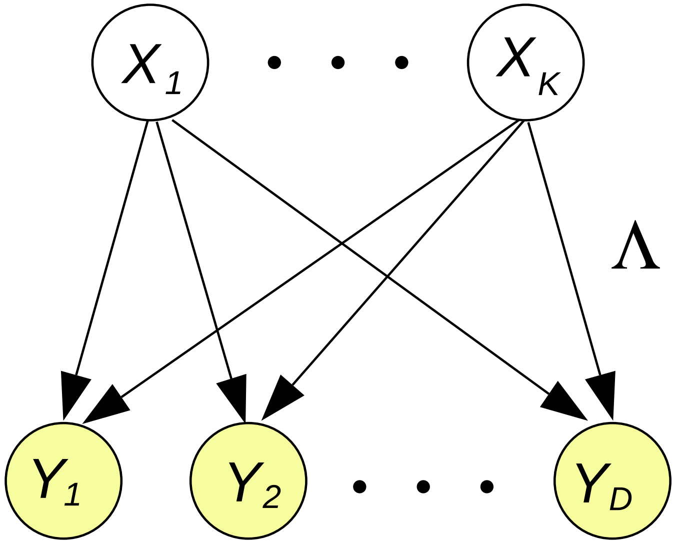
- Some data has higher-order or non-linear structure.

- Not all random processes fit the central limit theorem.

- Even if data are Gaussian, if D is large the full multivariate Gaussian model may be difficult to handle. There are $D(D+1)/2$ parameters in the covariance matrix.

Factor Analysis

Factor analysis models high dimensional data $\mathbf{y}$ in terms of a linear transformation of some smaller number of latent factors, $\mathbf{x}$.



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
- $K < D$

Properties:

- $p(\mathbf{x}) = \mathcal{N}(0, I)$ and $\mathbf{y} = \Lambda\mathbf{x} + \epsilon$
- Since $p(\epsilon) = \mathcal{N}(0, \Psi)$, we get that $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\Lambda\mathbf{x}, \Psi)$
- $p(\mathbf{y}) = \int p(\mathbf{x})p(\mathbf{y}|\mathbf{x})d\mathbf{x} = \mathcal{N}(0, \Lambda\Lambda^\top + \Psi)$ where Λ is a $D \times K$ matrix, and Ψ is diagonal.

latent = hidden = unobserved = missing

Ways of thinking about Factor Analysis (FA)

- FA models high dimensional data in terms of a linear transformation of some smaller number of latent factors.

- FA is a method for parameterizing a $D \times D$ covariance matrix Σ in terms of $D \times K + D$ parameters, $\Lambda\Lambda^\top + \Psi$. Since K can be chosen by the user, this means that factor analysis can be applied to *very* high dimensional datasets.

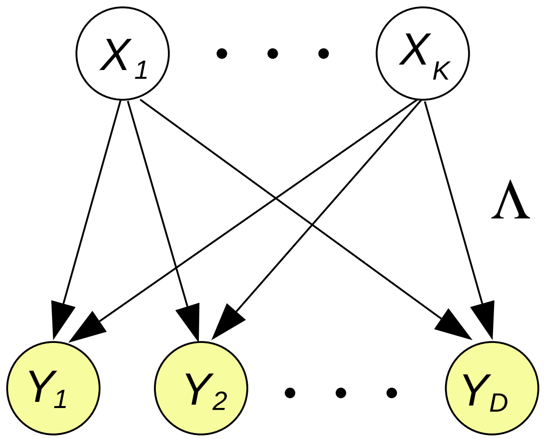
- FA is a method for modelling correlations among the observed variables.

- FA is a linear regression model, where the inputs are assumed to be hidden.

- FA is a method for doing dimensionality reduction. Given $\mathbf{y}$ we can represent it by the mean of $\mathbf{x}$. FA finds a low-dimensional projection of high dimensional data that captures the correlation structure of the data.

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x})p(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})} = \mathcal{N}(\beta\mathbf{y}, I - \beta\Lambda) \quad \text{where} \quad \beta = \Lambda^\top(\Lambda\Lambda^\top + \Psi)^{-1}$$

Factor Analysis



$$\mu = 0$$

$$\Sigma \approx \Lambda\Lambda^\top + \Psi$$

- ML learning for FA aims to fit Λ and Ψ given data. There is no closed form solution for ML parameters.

- Number of free parameters (corrected for symmetries):

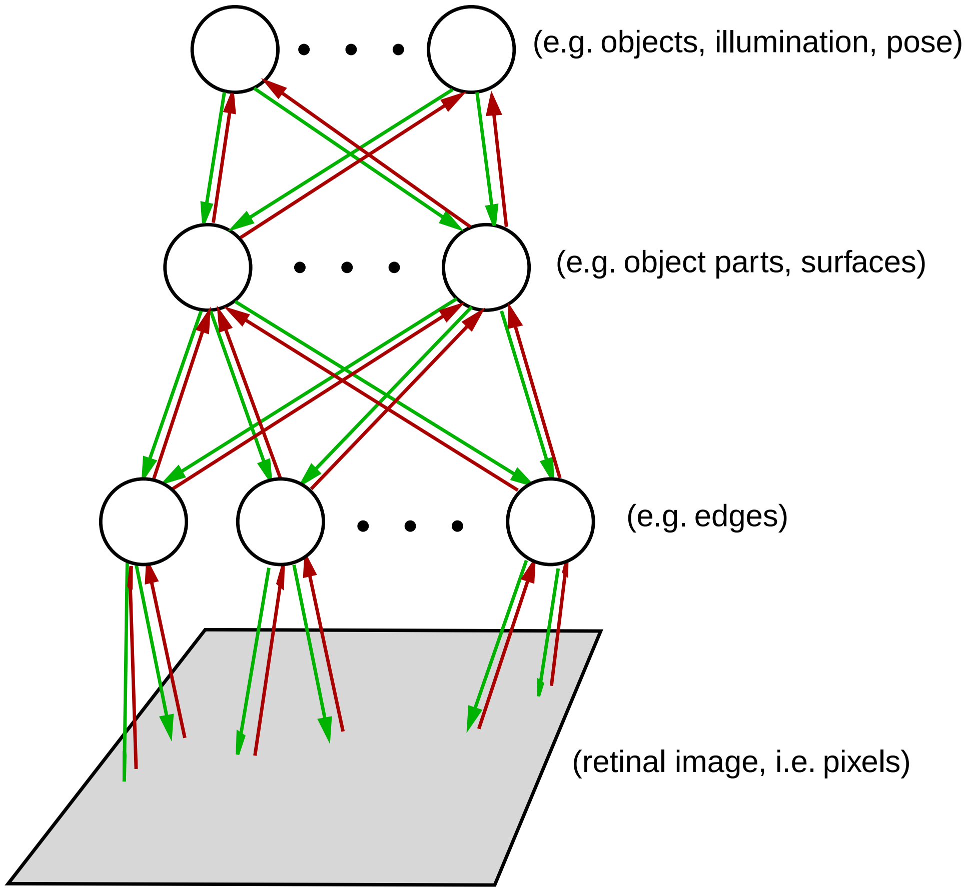
$$DK + D - \frac{K(K-1)}{2} < \frac{D(D+1)}{2}$$

- A Bayesian treatment would start with priors over Λ and Ψ and infer their posterior given the data.

$$p(\Lambda, \Psi|\mathcal{D}) = \frac{p(\mathcal{D}|\Lambda, \Psi)p(\Lambda, \Psi)}{p(\mathcal{D})}$$

Latent Variable Models

Explain correlations in $\mathbf{y}$ by assuming some latent variables $\mathbf{x}$



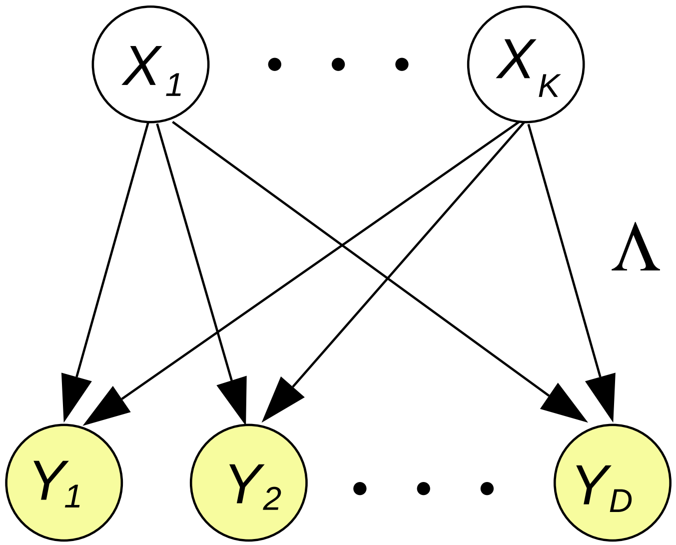
$$\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta}_x)$$

$$\mathbf{y}|\mathbf{x} \sim p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_y)$$

$$p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}_x, \boldsymbol{\theta}_y) = p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_y) p(\mathbf{x}|\boldsymbol{\theta}_x)$$

$$p(\mathbf{y}|\theta_x, \boldsymbol{\theta}_y) = \int d\mathbf{x}\, p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_y) p(\mathbf{x}|\boldsymbol{\theta}_x)$$

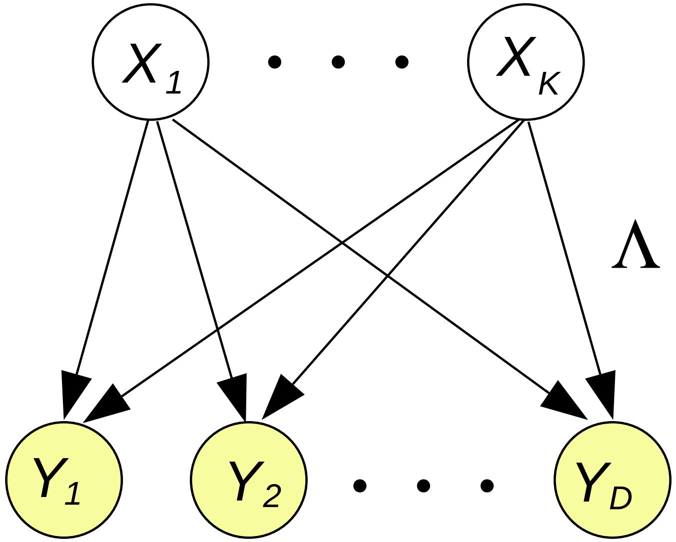
Probabilistic Principal Components Analysis (pPCA)



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 $\mathcal{N}(0,\sigma^2)$ Gaussian noise
- $K < D$
- pPCA is factor analysis with isotropic noise: $\Psi = \sigma^2 I$
- pPCA finds same principal subspace as PCA, but is a well-defined probabilistic model.

Principal Components Analysis (PCA)



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

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

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

Usually in PCA we choose columns of Λ to be orthonormal, i.e. $\Lambda^\top \Lambda = I$, therefore:

$$\beta = \Lambda^\top$$

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.

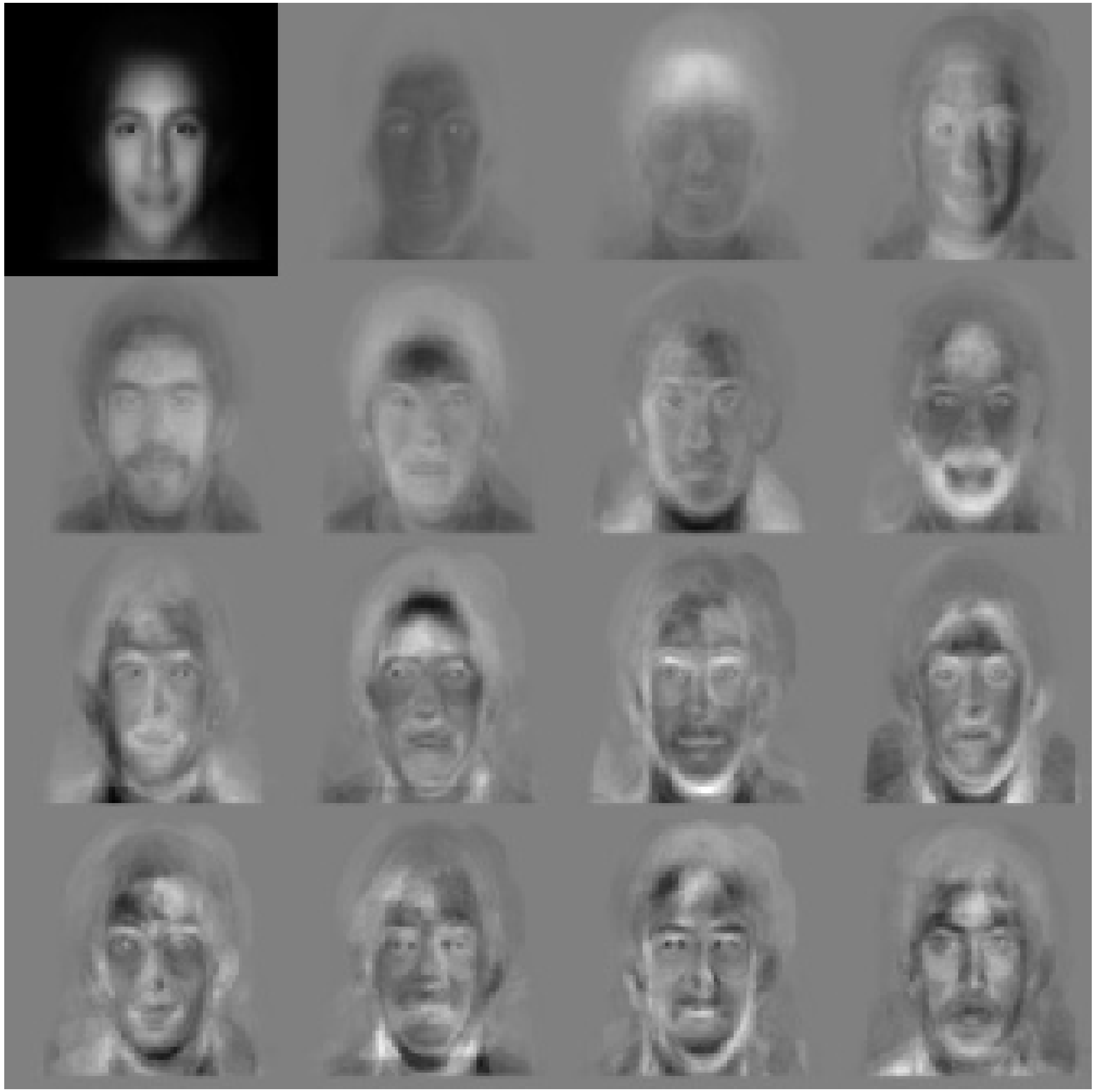
The K Principal Components are the K eigenvectors with the largest eigenvalues of the data covariance matrix (i.e. K directions with the largest variance).

Note: Σ can be decomposed:

$$\Sigma = USU^\top$$

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

Example of PCA: Eigenfaces



from www-white.media.mit.edu/vismod/demos/facerec/basic.html

Mutual Information and PCA

Problem: Given $\mathbf{y}$, find $\mathbf{x} = A\mathbf{y}$ with columns of A unit vectors, s.t. $I(\mathbf{x};\mathbf{y})$ is maximised (assuming that $P(\mathbf{y})$ is Gaussian).

$$I(\mathbf{x};\mathbf{y}) = H(\mathbf{x}) + H(\mathbf{y}) - H(\mathbf{x},\mathbf{y}) = H(\mathbf{x})$$

So we want to maximise the entropy of $\mathbf{x}$. What is the entropy of a Gaussian?

$$H(\mathbf{z}) \quad = \quad -\int d\mathbf{z}\, p(\mathbf{z}) \ln p(\mathbf{z}) = \frac{1}{2}\ln|\Sigma| + \frac{D}{2}(1 + \ln 2\pi) \qquad (1)$$

Therefore we want the distribution of $\mathbf{x}$ to have largest volume (i.e. det of covariance matrix).

$$\Sigma_x = A\Sigma_y A^\top = AUS_yU^\top A^\top$$

So, A should be aligned with the columns of U which are associated with the largest eigenvalues (variances).

Gradient Methods for Learning FA

Write down negative log likelihood:

$$\frac{1}{2}\log|2\pi(\Lambda\Lambda^\top + \Psi)| + \frac{1}{2}\mathbf{y}^\top(\Lambda\Lambda^\top + \Psi)^{-1}\mathbf{y}$$

Optimise w.r.t. Λ and Ψ (need matrix calculus) subject to constraints

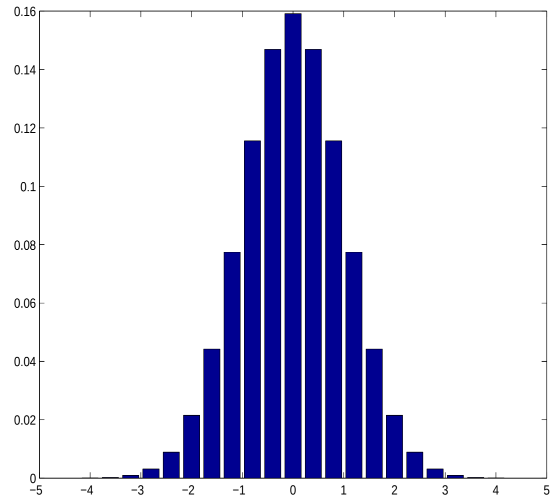
We will soon see an easier way to learn latent variable models...

Appendix: Source Coding Under a Gaussian Model

Consider coding real valued numbers x under a Gaussian model of the data.

- How many bits should we use for each x?

- Clearly we need to limit the precision of our code, otherwise we will need infinitely many bits. Let's use precision Δ.



- Remember, from Shannon's source coding theorem.

$$
\begin{aligned}
l(x) &= -\log P(x) \approx -\log[p(x)\Delta] = -\log p(x) - \log \Delta \\
&= \frac{(x-\mu)^2}{2\sigma^2} + \frac{1}{2}\log 2\pi + \log \sigma - \log \Delta
\end{aligned}
$$

- Note as $\Delta \Rightarrow 0$ then $l(x) \Rightarrow \infty$.

So we need $l(x)$ bits to code x, which grows quadratically with distance from x to μ.

Appendix: FA vs PCA

- PCA is rotationally invariant; FA is not
- FA is measurement scale invariant; PCA is not
- FA and pPCA define valid probabilistic models; PCA does not