COMP9417 – Machine Learning and Data Mining

W12 – Probabilistic Graphical Models

Instructor: Edwin V. Bonilla

School of Computer Science and Engineering

May 23rd, 2017

Acknowledgements

Material derived from:

- [Barber, BRML, 2012] Bayesian Reasoning and Machine Learning, David Barber, 2012

- [Bishop, PRML, 2006] Pattern Recognition and Machine Learning, Christopher Bishop, 2006

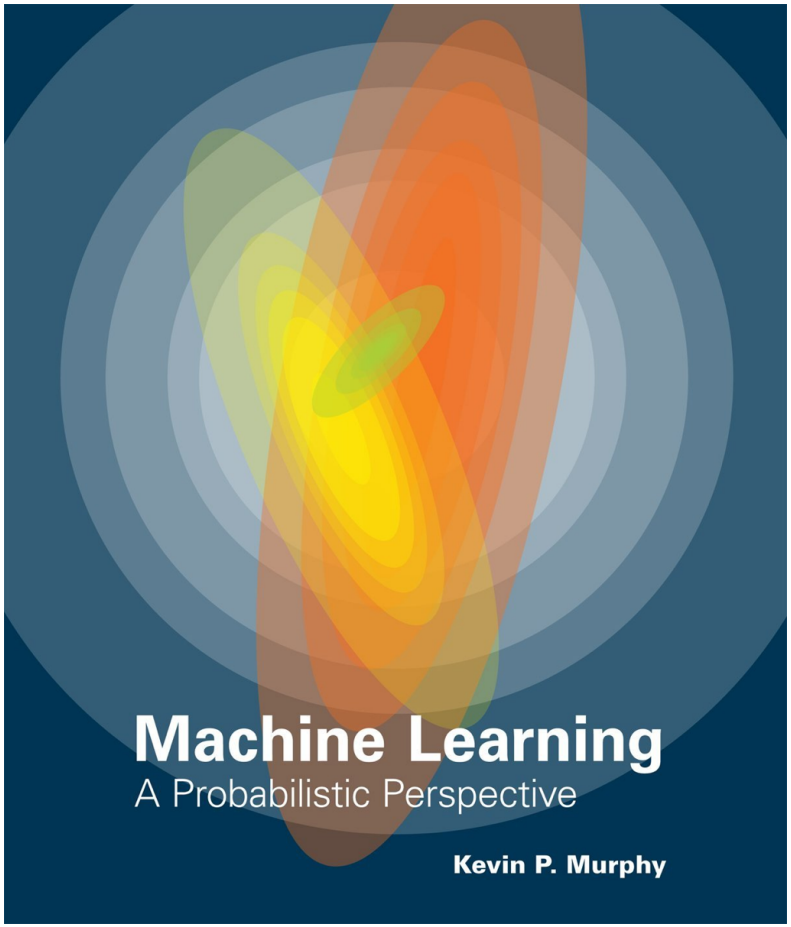
- [Bonilla, PML, 2017] Slides from course to industry, Edwin V. Bonilla, © 2010 – 2017

Aims

This lecture will reinforce the main concepts in popular machine learning methods for dimensionality reduction, clustering and classification and will explain these methods from a probabilistic graphical model perspective. Following it you should be able to:

- Apply linear dimensionality reduction methods such as PCA and understand them as the limiting case of a linear Gaussian model (PPCA).
- Apply k-means to clustering problems and its probabilistic counterpart Gaussian Mixture Models.
- Differentiate generative and discriminative approaches to probabilistic classification.
- Apply Naive Bayes and class-conditional Gaussian models to classification problems.
- Estimate the parameters of the above probabilistic models.

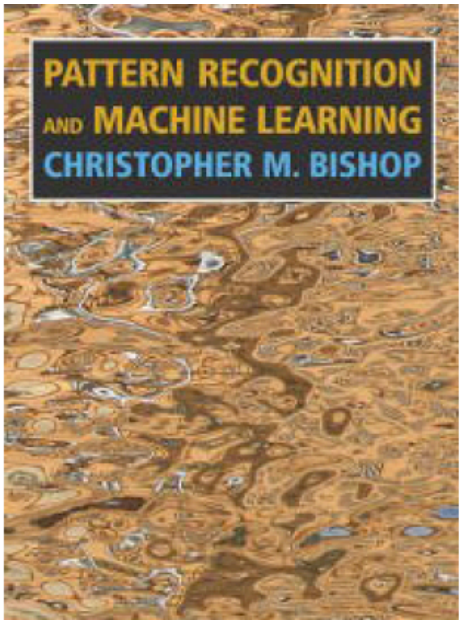
Probabilistic Graphical Models – Best Books Available

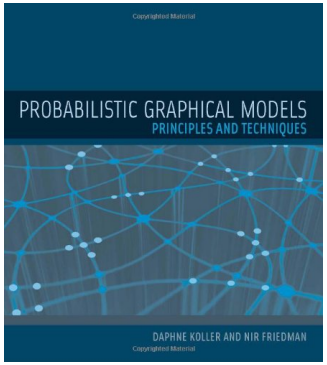


Code available

Fully available online

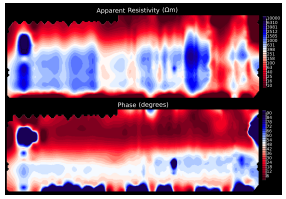
Code available



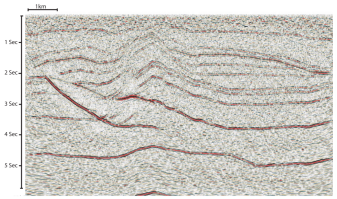


Why Probabilistic Models: Example

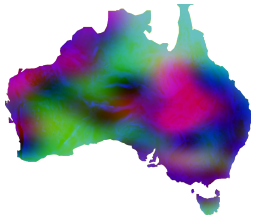
Data Fusion for Geothermal Energy Exploration



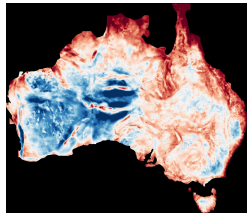
Magnetotellurics



Seismic

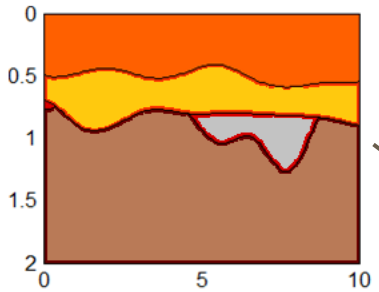


Temperature

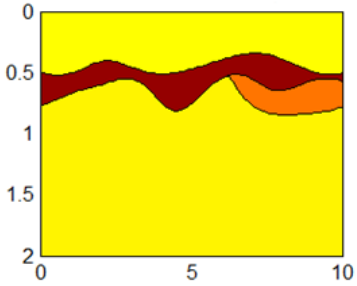


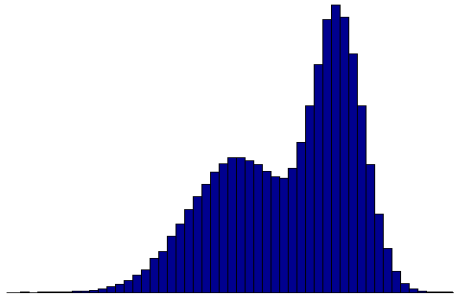
Gravity

. . .



Prior





Likelihood

Posterior distribution

Decision Support

Types of Learning

- Supervised Learning: There is a “teaching signal”
  - Classification, regression, time series prediction

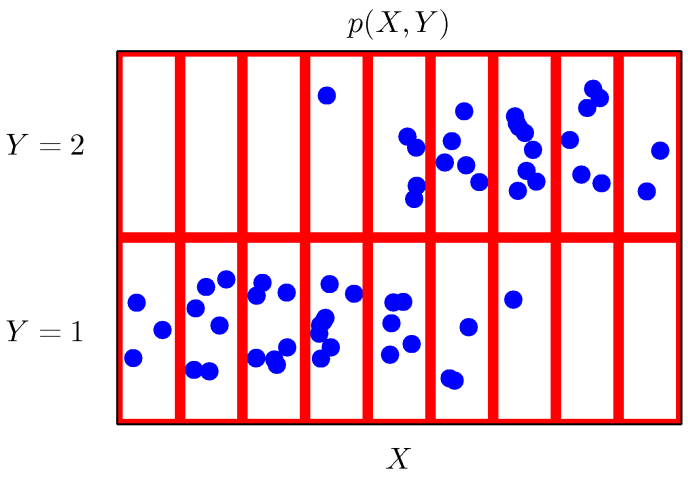
- Unsupervised Learning: Descriptive modelling
  - Dimensionality reduction, visualisation, clustering, latent variable modelling, density estimation, time series modelling

- Reinforcement Learning: Delayed reward (indirect supervisory signal)

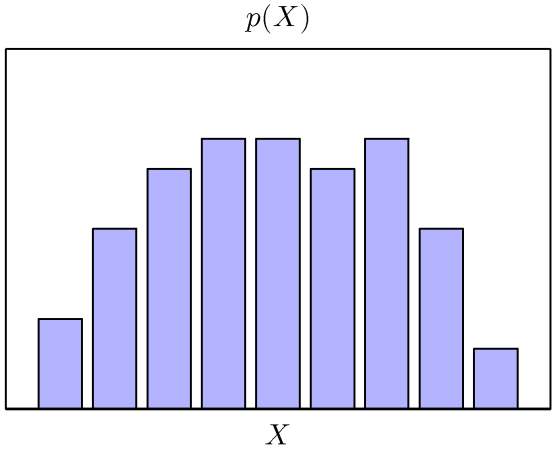
We will study these topics from a (probabilistic) graphical models perspective

Preliminaries

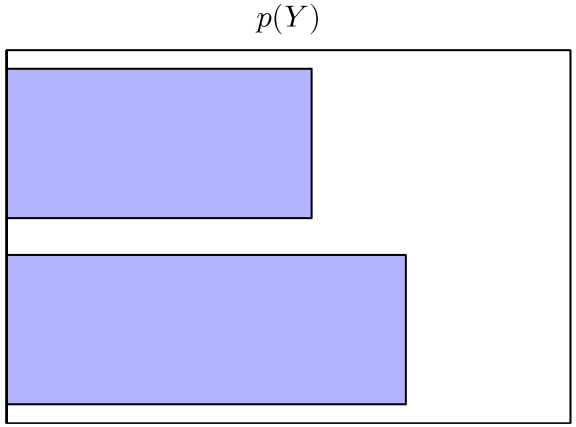
Probability Distributions: Illustration



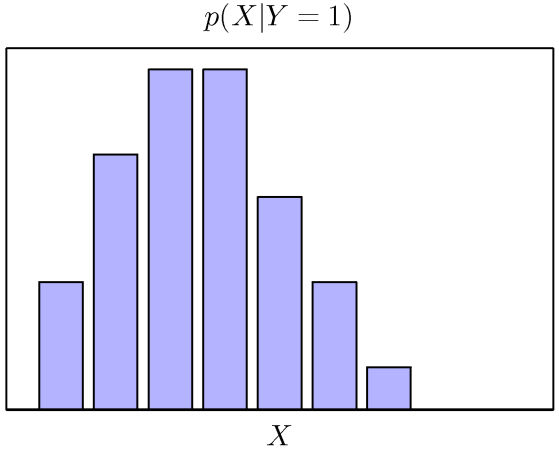
Joint



Marginal



Marginal



Conditional

The Rules of Probability and Terminology

Discrete Case

- Sum Rule: $P(X) = \sum_{Y} P(X, Y)$
- Product Rule: $P(X, Y) = P(Y|X)P(X)$
- By symmetry: $P(Y, X) = P(X, Y)$
- Therefore: $P(X) = \sum_{Y} P(X, Y) = \sum_{Y} P(X|Y)P(Y)$
- Normalisation, marginalisation

Bayes' Theorem

Bayesian Inference

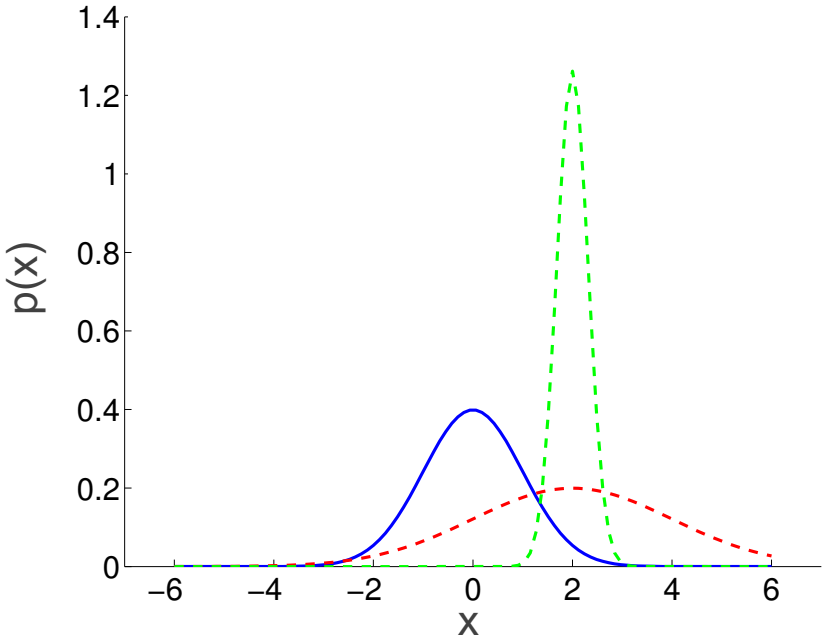
Bayesian inference provides us with a a mathematical framework explaining how to change our (prior) beliefs in the light of new evidence.

$$\underbrace{p(Z|X)}_{\text{posterior}} = \frac{\overbrace{p(X|Z)}^{\text{likelihood}}\overbrace{p(Z)}^{\text{prior}}}{\underbrace{p(X)}_{\text{evidence}}}$$

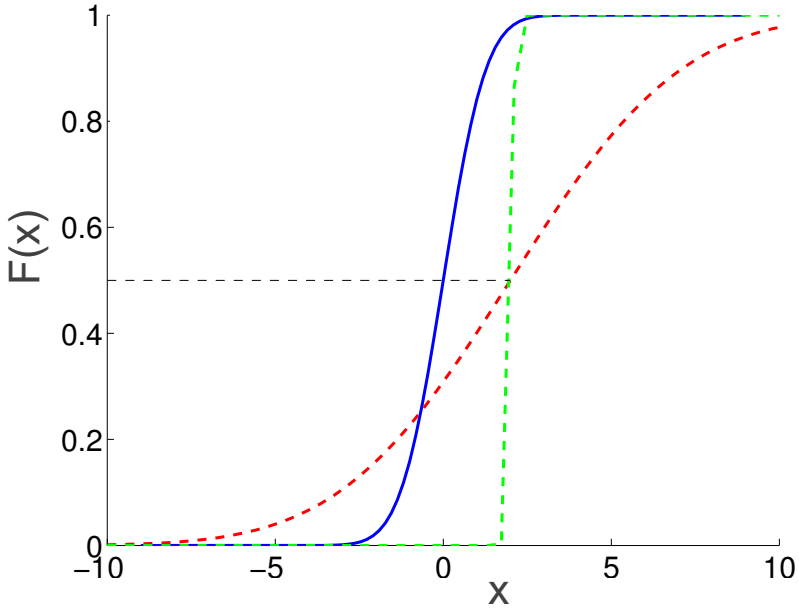
$$p(Z|X) = \frac{p(X|Z)p(Z)}{\sum_{Z'} p(X|Z')p(Z')}$$

Continuous Random Variables

Probability Density Function (pdf)



Cumulative Distribution Function (cdf)



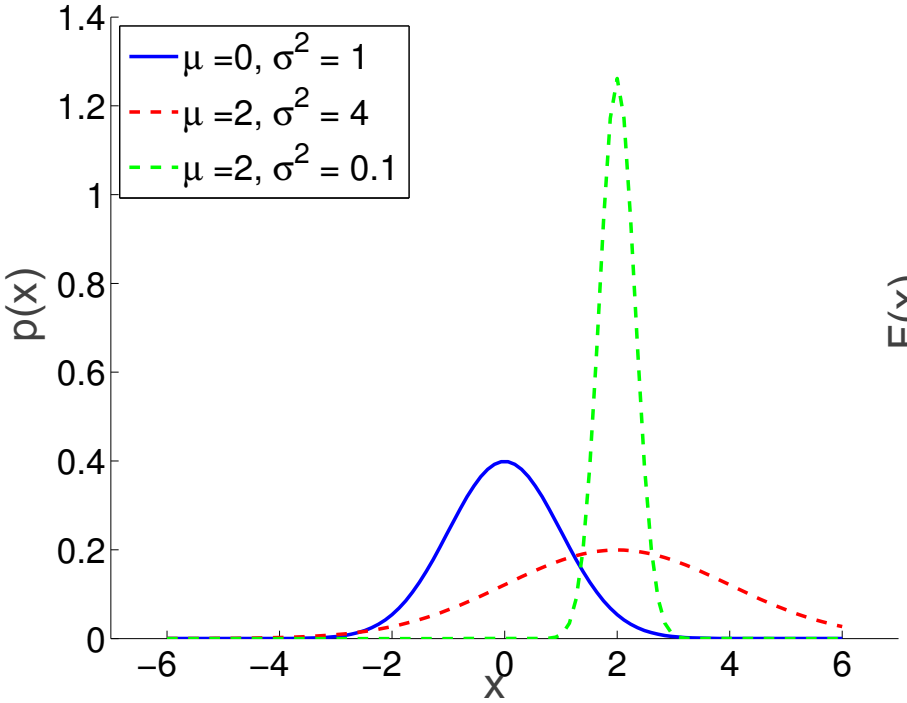
$$p(x) \geq 0, \qquad \int_{-\infty}^{\infty} p(x)dx = 1$$

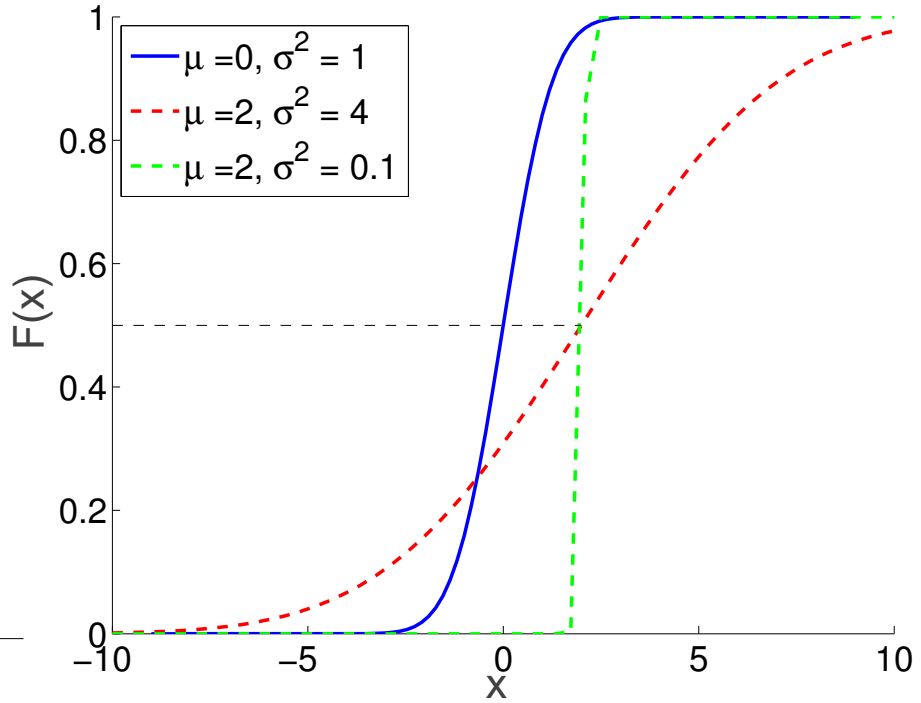
$$p(a < x < b) = \int_{a}^{b} p(x)dx$$

$$F(x) = p(X \leq x) = \int_{-\infty}^{x} p(z)dz$$

All rules of probability apply where we replace sums with integrals

The Gaussian Distribution – 1D Example





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

Parameters

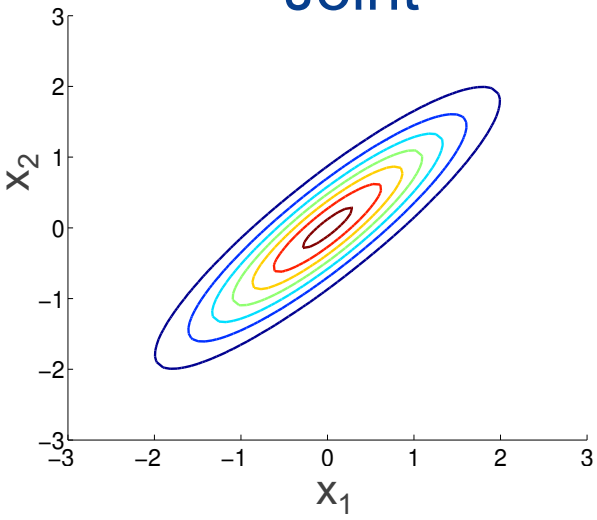
$$F(x) = \int_{-\infty}^{x} \mathcal{N}(z|\mu, \sigma^2)dz$$

In D >1 dimensions:

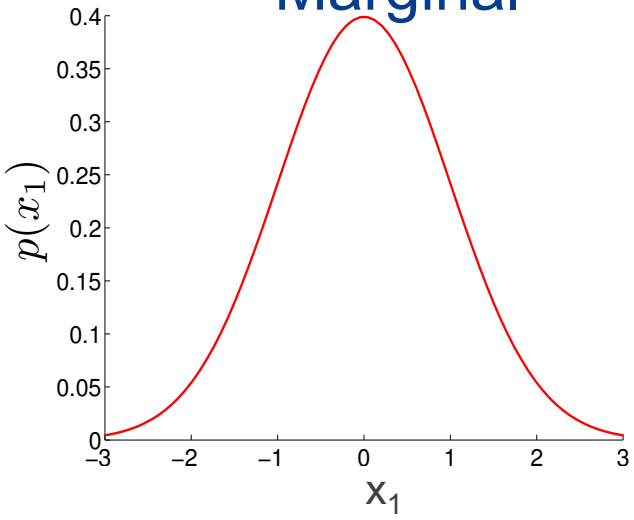
$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{|2\pi\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

The Gaussian Distribution – 2D Example

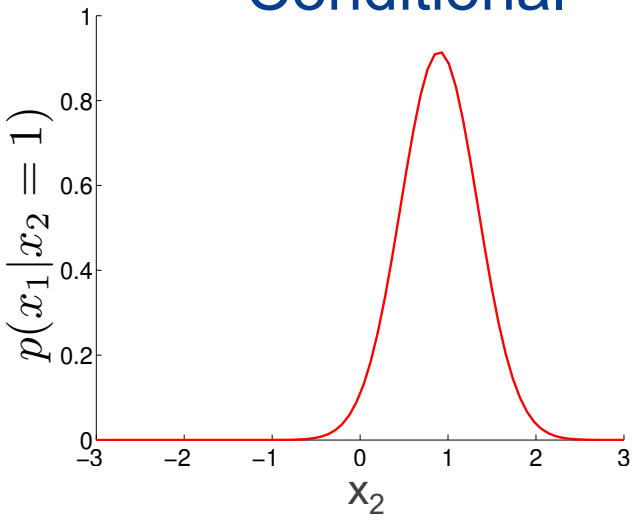
Joint



Marginal



Conditional



$$p(x_1, x_2) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \longrightarrow \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}^T & \boldsymbol{\Sigma}_{22} \end{bmatrix} \right)$$

The marginal and the conditional distributions are also Gaussians:

$$p(\mathbf{x}_1) = \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11})$$

$$p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{12}^T)$$

Covariance and Precision Matrices

$\boldsymbol{\Sigma}$: is the covariance matrix

$\boldsymbol{\Sigma}^{-1}$: is the precision matrix

Are these two variables statistically independent?

- An entry $\boldsymbol{\Sigma}_{ij} = 0$ indicates that the variables i and j are marginally independent
- An entry $(\boldsymbol{\Sigma}^{-1})_{ij} = 0$ indicates that the variables i and j are conditionally independent given all the other variables
- Marginalizing out a variable leaves $\boldsymbol{\Sigma}$ unchanged but changes $\boldsymbol{\Sigma}^{-1}$

In general, there is a difference between statistical independence and correlation

Parameter Estimation for The Gaussian Distribution – Maximum Likelihood Training (1)

We are given a dataset $\mathcal{D} = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$ with $\mathbf{x} \in \mathbb{R}^D$ and we want to fit a multivariate Gaussian $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Goal: Find the best parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ that fit the data.

Assuming iid observations, the data log-likelihood can be written as:

$$L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} p(\mathbf{x}^n|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \log L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \log p(\mathbf{x}^n|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$= -\frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}^n - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}^n - \boldsymbol{\mu}) - \frac{N}{2}\log|2\pi\boldsymbol{\Sigma}|$$

Parameter Estimation for The Gaussian Distribution – Maximum Likelihood Training (2)

The optimal **μ** and **Σ** can be obtained by computing the corresponding derivatives of L(**μ**, **Σ**) and setting them to zero:

- For the mean we have:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = \sum_{n=1}^{N} \boldsymbol{\Sigma}^{-1}(\mathbf{x}^n - \boldsymbol{\mu})$$

$$\widehat{\boldsymbol{\mu}}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}^n$$

- Similarly, for the covariance:

$$\widehat{\boldsymbol{\Sigma}}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{x}^n - \boldsymbol{\mu})(\mathbf{x}^n - \boldsymbol{\mu})^T$$

These are simply the sample mean and sample covariance of the data!

Outline

I. Dimensionality Reduction
- PCA and PPCA

II. Clustering
- K-means and Gaussian Mixture Models

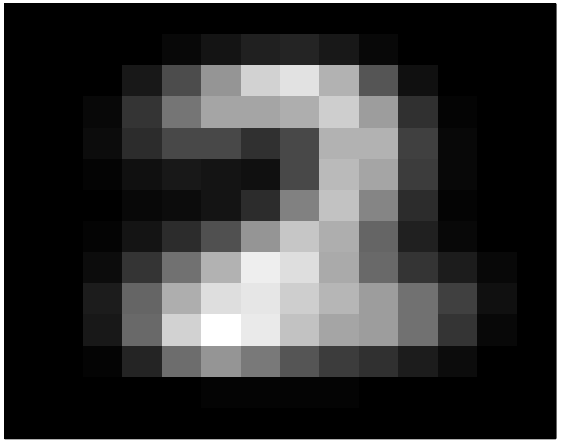
III. Probabilistic Classification – Generative Approaches
- Naïve Bayes and class-conditional Gaussian models

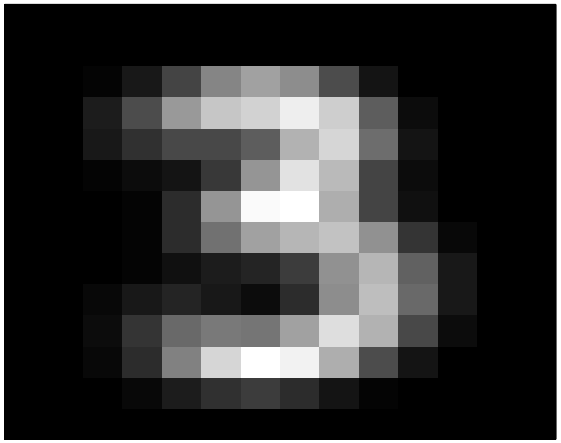
I. Dimensionality Reduction: Probabilistic Principal Component Analysis (PPCA)

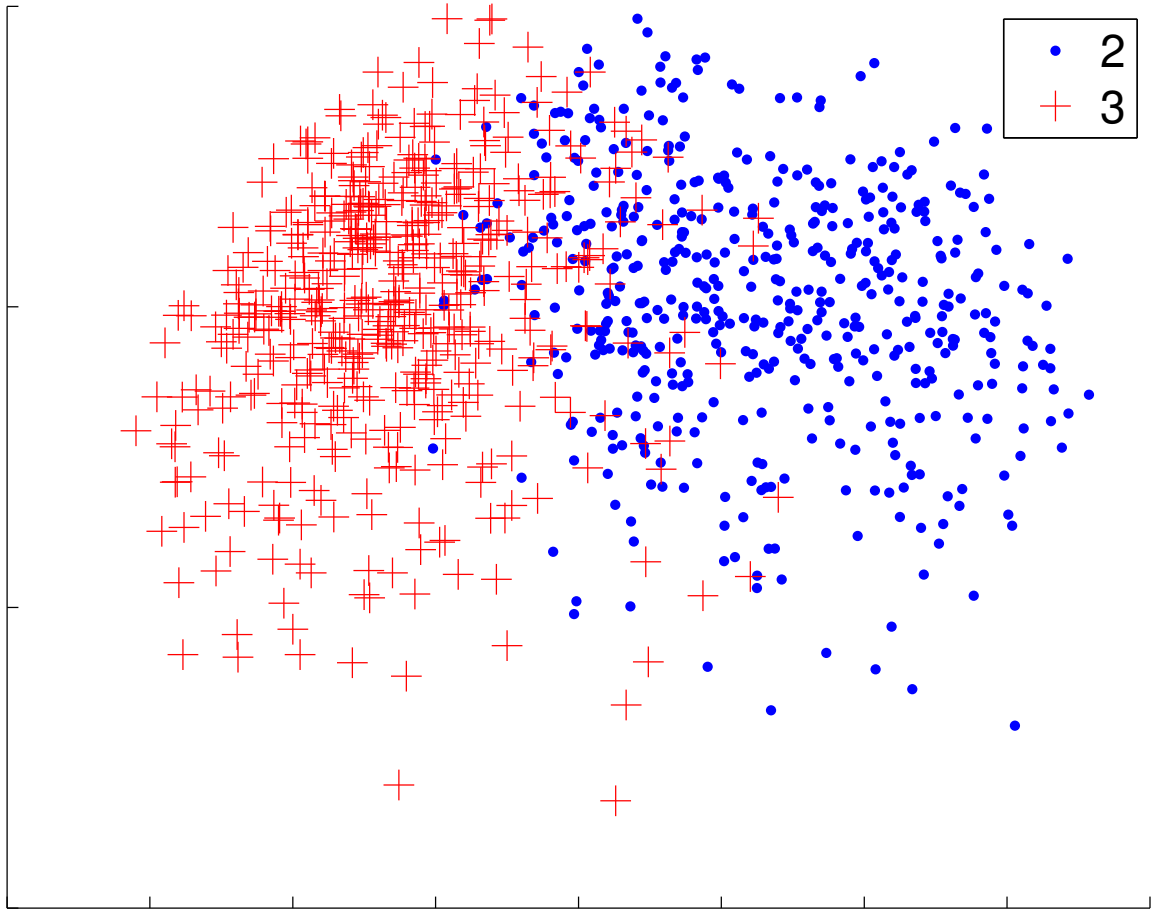
Low Dimensional Manifolds

High-dimensional data may lie in low-dimensional manifolds

- Only a small number of dimensions (degrees of freedom) may be sufficient to explain well the data







Why Dimensionality Reduction?

- Learning in high-dimensional spaces may be very hard
- We can use low-dimensional representations for visualisation
- These representations can be more discriminative
- We will focus on unsupervised and linear techniques for dimensionality reduction
  - Note that we may be "throwing" away useful information

Principal Component Analysis (PCA)
Problem Definition

Consider a set of N datapoints $\{\mathbf{x}^{(n)}\}_{n=1}^{N}$ with $\mathbf{x} \in \mathbb{R}^D$.

- Our goal is to find a low-dimensional linear representation $\mathbf{z}^{(n)} \in \mathbb{R}^K$ with K < D that explains the data well.

- We can have (at least) two criteria:
  1. Find an orthogonal projection such as the variance of the projected data is maximized.
  2. Find a linear projection that minimizes the reconstruction error (squared error).

- These criteria lead to the same solution given by PCA

Principal Component Analysis (PCA)

The Algorithm

Input: D-dimensional data $\{\mathbf{x}^{(n)}\}_{n=1}^{N}$, K

1. Compute sample mean $\bar{\mathbf{x}}$ and covariance $\mathbf{S}$
2. Find K eigenvectors $\mathbf{e}^1, \ldots, \mathbf{e}^K$ corresponding to the largest K eigenvalues $\lambda_1, \ldots, \lambda_K$ and construct the matrix

$$\mathbf{E} = (\mathbf{e}^1, \ldots, \mathbf{e}^K)$$

3. Output: Lower dimensional representation is given by:

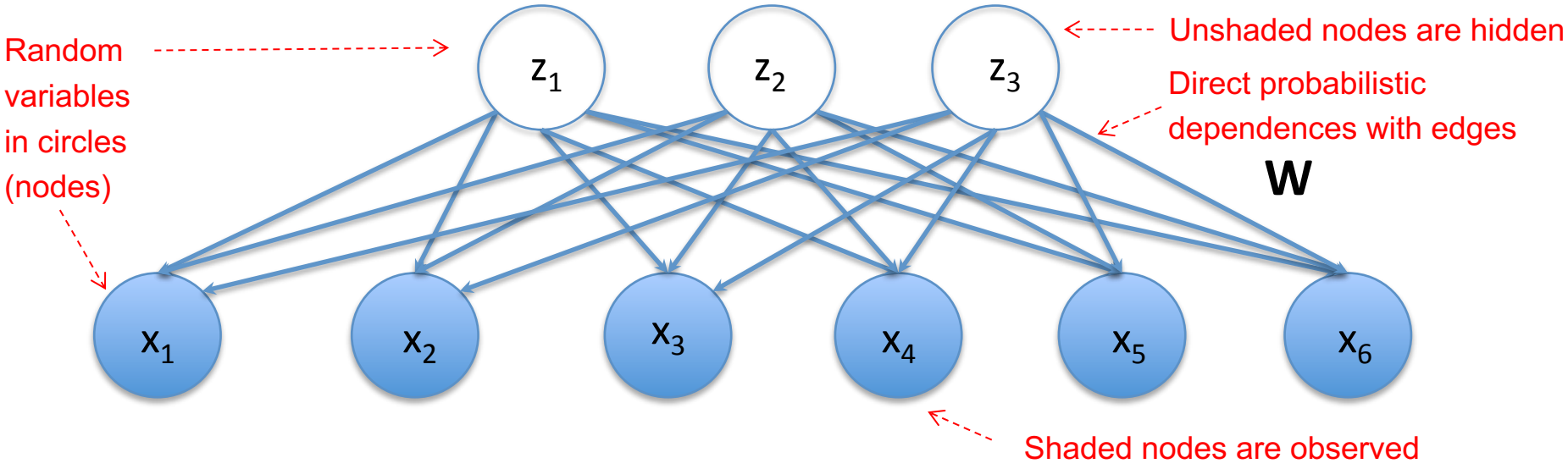
$$\mathbf{z}^n = \mathbf{E}^T(\mathbf{x}^n - \bar{\mathbf{x}})$$

How much variance have we explained by selecting K components?

- HINT: We can use the fact that the total variance is given by $\sum_{i=1}^{D} \lambda_i$
- We can use this to establish a criterion to select K in an unsupervised setting.

Probabilistic Principal Component Analysis (PPCA)

In fact, PCA can be derived from a proper probabilistic latent variable model:



We observe $\mathbf{x} \in \mathrm{R}^D$ and assume that it is generated by a latent low-dimensional vector $\mathbf{z} \in \mathrm{R}^K$:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}), \qquad p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$$

- The D × K matrix **W**, the D-dimensional bias vector **μ** and the noise variance σ^2 are the model parameters
  - How do we learn these parameters?

PPCA: Parameter Estimation

Maximum Likelihood Parameters

PPCA defines p(**z**,**x**) but **z** are latent as we only observe $\{\mathbf{x}^{(n)}\}_{n=1}^{N}$

- Need to marginalise **z** and get p(**x**) → Analytical solution!

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}),$$

Assuming iid observations, the data log-likelihood is:

$$\mathcal{L}(\mathbf{W}, \boldsymbol{\mu}, \sigma^2) = \sum_{n=1}^{N} \log p(\mathbf{x}^n|\mathbf{W}, \boldsymbol{\mu}, \sigma^2)$$

and Maximizing $\mathcal{L}$ wrt **μ**, **W** and **σ**^{2} we obtain:

$$\widehat{\boldsymbol{\mu}}_{\mathrm{ML}} = \bar{\mathbf{x}} \quad \widehat{\mathbf{W}}_{\mathrm{ML}} = \mathbf{E}(\boldsymbol{\Lambda} - \sigma^2_{\mathrm{ML}}\mathbf{I})^{1/2}\mathbf{R} \quad \widehat{\sigma}^2_{\mathrm{ML}} = \frac{1}{D-K}\sum_{i=K+1}^{D}\lambda_i$$

Ordered eigenvalues and eigenvectors of data covariance

Rotation matrix, e.g. **R=I**

Average variance of Discarded dimensions

PPCA

How to Do Dimensionality Reduction

Probabilistic inference problem

- Compute p(**z** | **x**) → Analytical solution!

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mathbf{M}^{-1}\mathbf{W}^T(\mathbf{x}-\boldsymbol{\mu}), \sigma^{-2}\mathbf{M}), \quad \mathbf{M} = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}.$$

Using the ML estimators, the posterior mean is:

$$\mathbb{E}[\mathbf{z}|\mathbf{x}] = (\widehat{\mathbf{W}}_{\mathrm{ML}}^T\widehat{\mathbf{W}}_{\mathrm{ML}} + \widehat{\sigma}_{\mathrm{ML}}^2\mathbf{I})^{-1}\widehat{\mathbf{W}}_{\mathrm{ML}}^T(\mathbf{x}-\bar{\mathbf{x}})$$

- and taking the limit $\sigma^2 \rightarrow 0$ we obtain PCA, an orthogonal projection of the datapoints into the latent space.

Advantages of probabilistic approach:

- Proper likelihood function
  - Mixture models and Bayesian extensions
- Can find dimensionality automatically

PCA & PPCA

Final Remarks

- PPCA is a constrained Gaussian model
  - $\mathbf{\Sigma} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}$
  - if K = D then $\mathbf{\Sigma} = \mathbf{S}$, the sample covariance
  - The number of independent parameters only grows linearly with D (cf. full Gaussian, isotropic Gaussian)
  - But we still capture the K most significant correlations

- Caution in supervised scenarios: We may be throwing away the important information

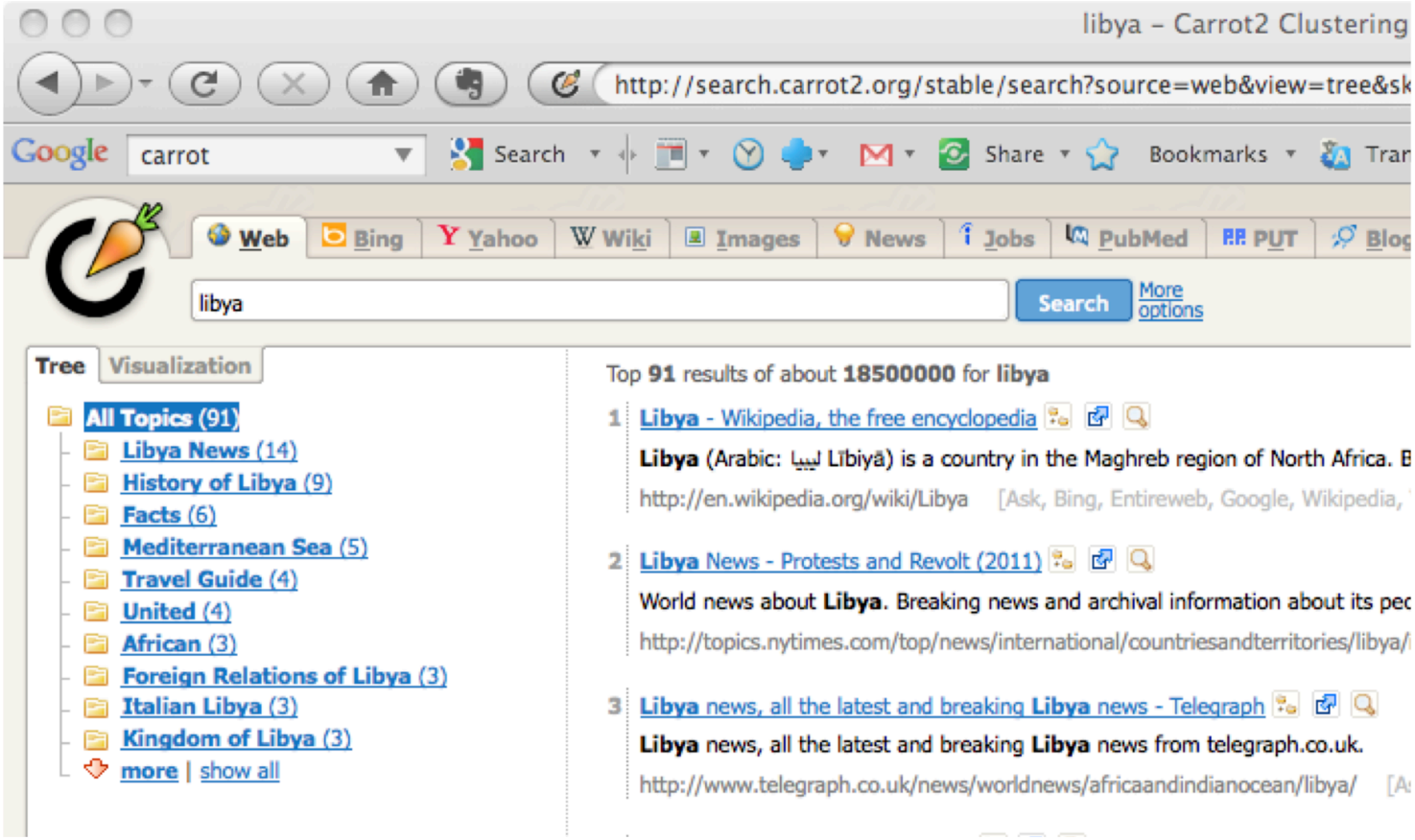
- We are assuming the data lies close to a hyperplane (linear technique)
  - Manifold may be curved or data can be clustered

- PCA is a very old technique with many different names, e.g. the Karhunen-Loève transform
  - Most commonly used dimensionality reduced method

II. Clustering and Mixture Models: K-Means and Gaussian Mixtures

Clustering

Goal: Given (e.g. documents), assign "similar" ones to one of K groups (clusters)



Clustering with K-means

Each cluster $\mathcal{C}_k$ is associated with a prototype vector (or centre) $\boldsymbol{\mu}_k$ and each $\mathbf{x}^n$ with $\mathbf{z}^n$ where $z_k^n = 1$ iff $\mathbf{x}^n$ belongs to cluster $\mathcal{C}_k$.

Repeat until convergence:

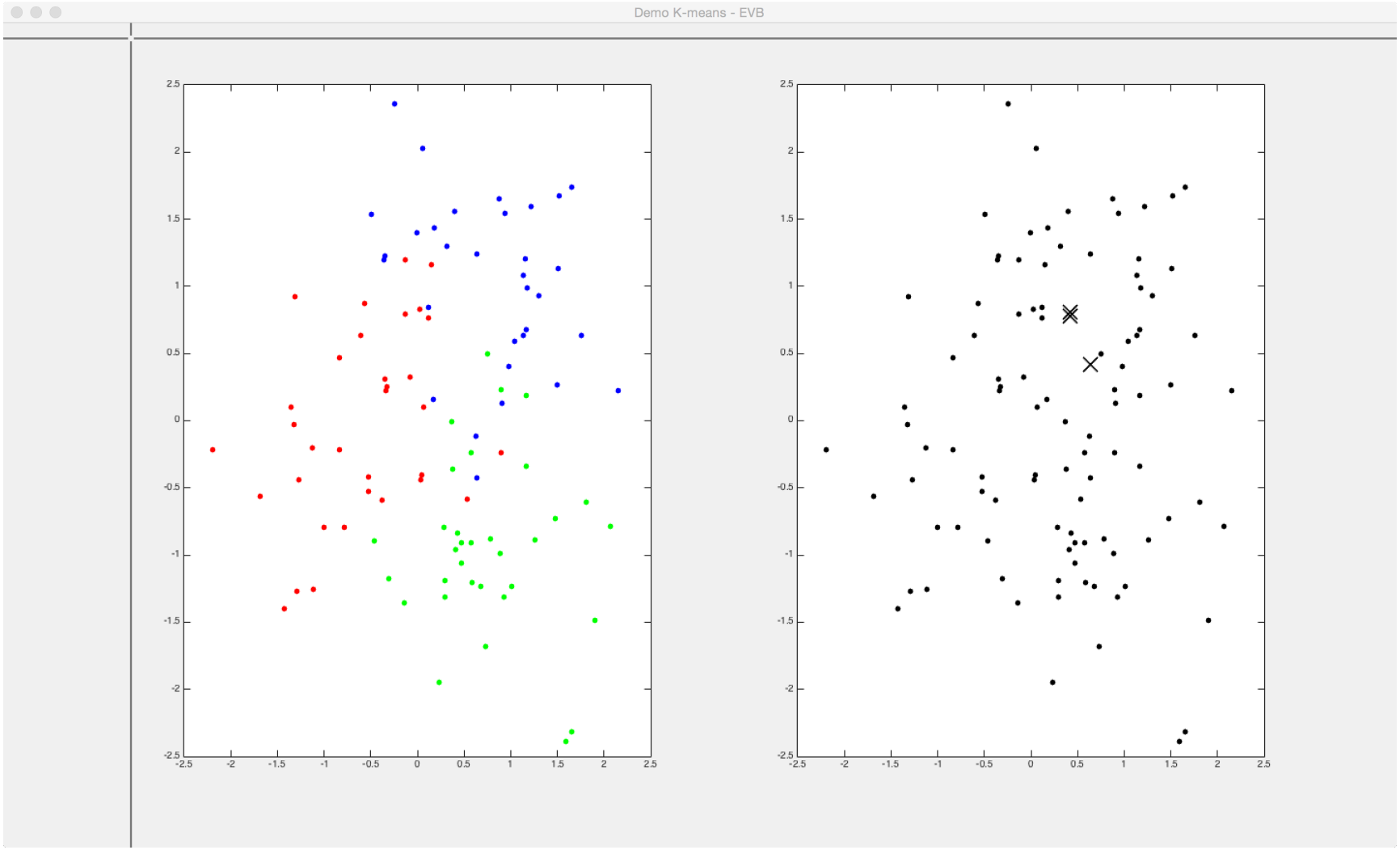
1. Initialize centres $\{\boldsymbol{\mu}_k\}_{k=1}^K$ (e.g. using K data vectors)
2. Set $z_k^n = 1$ iff $k = \underset{j}{\operatorname{argmin}} \|\mathbf{x}^n - \boldsymbol{\mu}^j\|^2$ and $z_k^n = 0$ otherwise
3. Set $\boldsymbol{\mu}_k = \frac{1}{|\mathcal{C}_k|} \sum_{n=1}^{N} z_k^n \mathbf{x}^n$

- This procedure can be shown to minimize the error function:

$$\mathcal{E} = \sum_{k=1}^{K} \sum_{n=1}^{N} z_k^n \|\mathbf{x}^n - \boldsymbol{\mu}^k\|^2$$

  - Convergence only to a local minimum

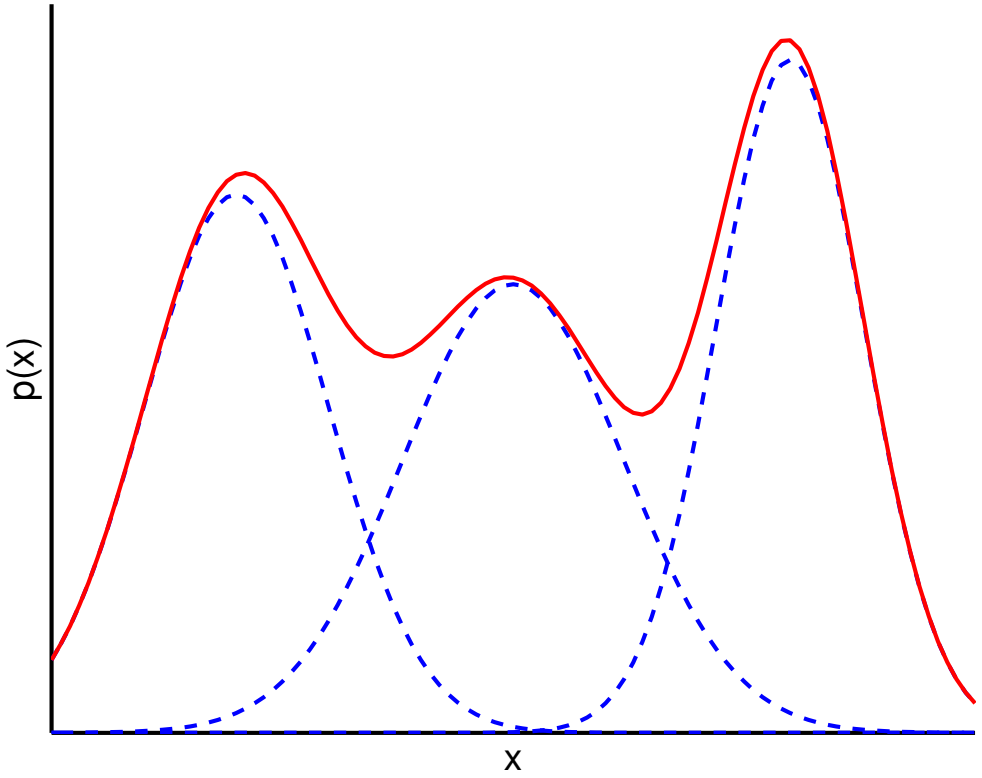
K-means Clustering Demo



demo_kmeans.m

Gaussian Mixture Models (GMMs)

Modelling Complex Densities



In blue: original Gaussians scaled by their priors $p(z_k = 1)$ with k = 1, 2, 3

In red: the resulting mixture

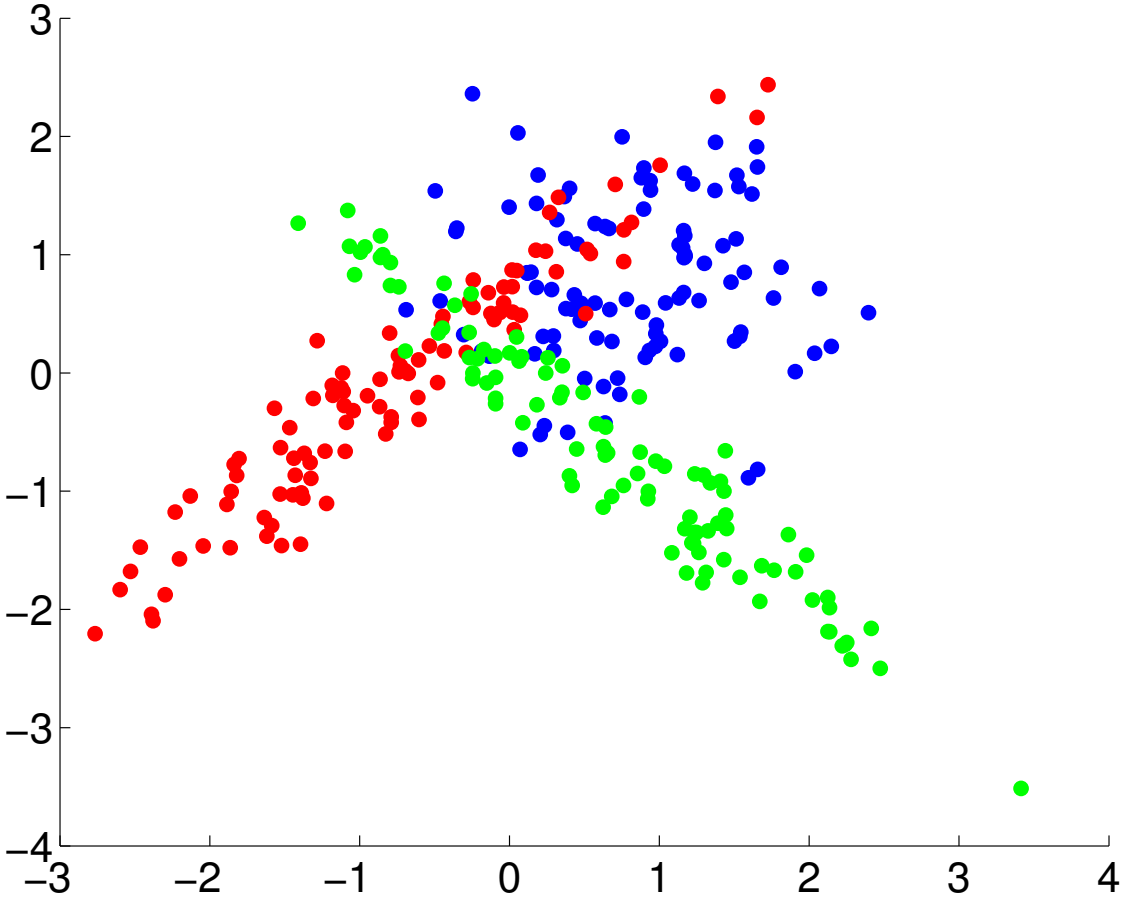
The final density is multimodal

Generative Process: For k = 1, 2, 3

1. Pick component (cluster) k with probability π_k
2. Draw $\mathbf{x} \sim N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

Gaussian Mixture Models (GMMs)

Data Generation – How can mixtures be used for clustering?



$$p_k = \frac{1}{3},\ k = 1, \ldots, 3$$

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$$

$$\boldsymbol{\Sigma}_2 = \begin{pmatrix} 1 & 0.95 \\ 0.95 & 1 \end{pmatrix}$$

$$\boldsymbol{\Sigma}_3 = \begin{pmatrix} 1 & -0.95 \\ -0.95 & 1 \end{pmatrix}$$

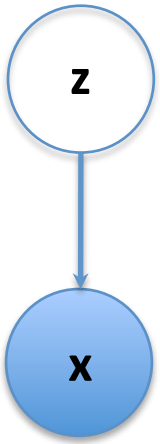
How can we represent this mixture of Gaussians with a graphical model?

Gaussian Mixture Models (GMMs)

Probabilistic Latent Variable model

In fact, K-means can be seen as a limiting case of a Gaussian mixture model:

- Given $z_k \in \{0,1\}$ with $\sum_{k=1}^{K} z_k = 1$



$$p(z_k = 1) = \pi_k \text{ with } 0 \leq \pi_k \leq 1, \sum_{k=1}^{K} \pi_k = 1, \quad p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

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

Marginal:

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

How can we learn $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$?

Gaussian Mixture Models (GMMs)

Parameter Estimation – Direct Likelihood Maximisation

- We can write down the data log-likelihood:

$$\mathcal{L}(\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}^n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)$$

- We can do gradient-based optimization directly but:
  - Note logarithm of the sum, no closed form solution
  - Constraints on π_k and on $\boldsymbol{\Sigma}_k$

- We can address these by having suitable parameterizations:
  - Softmax for π_k and Cholesky decomposition for $\boldsymbol{\Sigma}_k$

- An alternative solution is obtained via the EM algorithm
  - "closed-form" updates
  - The above constraints are automatically satisfied

Gaussian Mixture Models (GMMs)

Parameter Estimation – The Expectation-Maximisation (EM) Algorithm

Consider the case that **z** had been observed. Therefore:

$$\mathcal{L}^{\text{comp}}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(\mathbf{x}^n, \mathbf{z}^n | \boldsymbol{\theta}),$$

where $\mathcal{L}^{\text{comp}}$ refers to the complete data log-likelihood and $\boldsymbol{\theta}$ are the model parameters.

- The log of sum does not appear!
  - We would not have to marginalize the latent variable **z**

But life is hard and **z** has not been observed. What can we do?

1. Compute its expectation over the posterior $\langle \mathcal{L}^{\text{comp}} \rangle_{p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta})}$
2. Maximize this expectation wrt $\boldsymbol{\theta}$

The EM algorithm iterates the above steps and converges to a local minimum of the (incomplete) data log-likelihood

Gaussian Mixture Models (GMMs)

The EM Algorithm: E-step

In the E-step we need to compute the posterior $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$

$$p(z_k = 1|\mathbf{x}, \boldsymbol{\theta}^{\text{old}}) = \frac{p(z_k = 1|\boldsymbol{\theta}^{\text{old}})p(\mathbf{x}|z_k = 1, \boldsymbol{\theta}^{\text{old}})}{\sum_{j=1}^{K} p(z_j = 1|\boldsymbol{\theta}^{\text{old}})p(\mathbf{x}|z_j = 1, \boldsymbol{\theta}^{\text{old}})}$$

$$= \frac{\pi_k^{\text{old}} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^{\text{old}}, \boldsymbol{\Sigma}_k^{\text{old}})}{\sum_{j=1}^{K} \pi_j^{\text{old}} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j^{\text{old}}, \boldsymbol{\Sigma}_j^{\text{old}})} \stackrel{\text{def}}{=} \gamma_k$$

π_k is the prior probability of component *k* and γ_k is the posterior probability of component *k* given that we have observed **x**.

- γ_k can be viewed as the responsibility that component *k* takes for explaining observation **x**.

Gaussian Mixture Models (GMMs)

The EM Algorithm: M-step

In the M-step we need to maximise the objective function:

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \left\langle \sum_{n=1}^{N} \log p(\mathbf{x}^n, \mathbf{z}^n | \boldsymbol{\theta}) \right\rangle_{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})}$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_k^n \left[\log \pi_k + \log \mathcal{N}(\mathbf{x}^n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right]$$

Optimising wrt $\boldsymbol{\theta}$ and defining $N_k = \sum_{n=1}^{N} \gamma_k^n$ we have:

$$\pi_k = \frac{N_k}{N} \quad \boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^n \mathbf{x}^n \quad \boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^n (\mathbf{x}^n - \boldsymbol{\mu}_k)(\mathbf{x}^n - \boldsymbol{\mu}_k)^T$$

- Interpretation of $N_k, \pi_k, \boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$?

EM for Gaussian Mixture Models (GMMs)

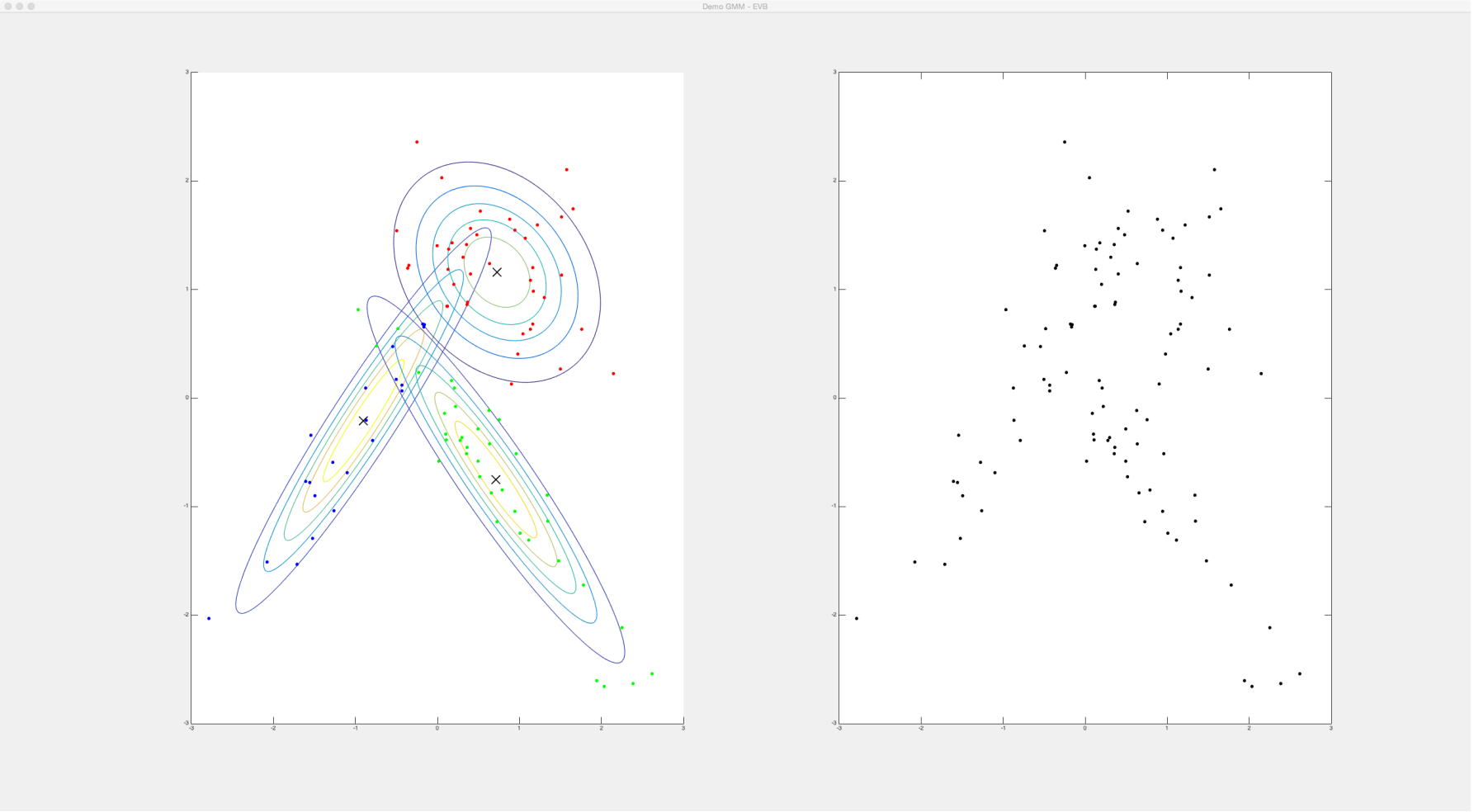
The Algorithm

Goal: Given $\mathcal{D} = \{\mathbf{x}^n\}_{n=1}^N$ (e.g. documents) learn a GMM

Input: $\mathcal{D} = \{\mathbf{x}^n\}_{n=1}^N$, K

1. Initialise $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ (how?) and evaluate data log-likelihood
2. Repeat until convergence of parameters or data log-likelihood
   a. E-step: Compute responsibilities γ_k^n using current parameters $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$
   b. M-step: Re-estimate $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ using current (fixed) responsibilities γ_k^n
   c. Compute data log-likelihood with current parameters and evaluate convergence

Learning GMMs: Demo



demo_gmm.m

Why EM Works (2) – Illustration

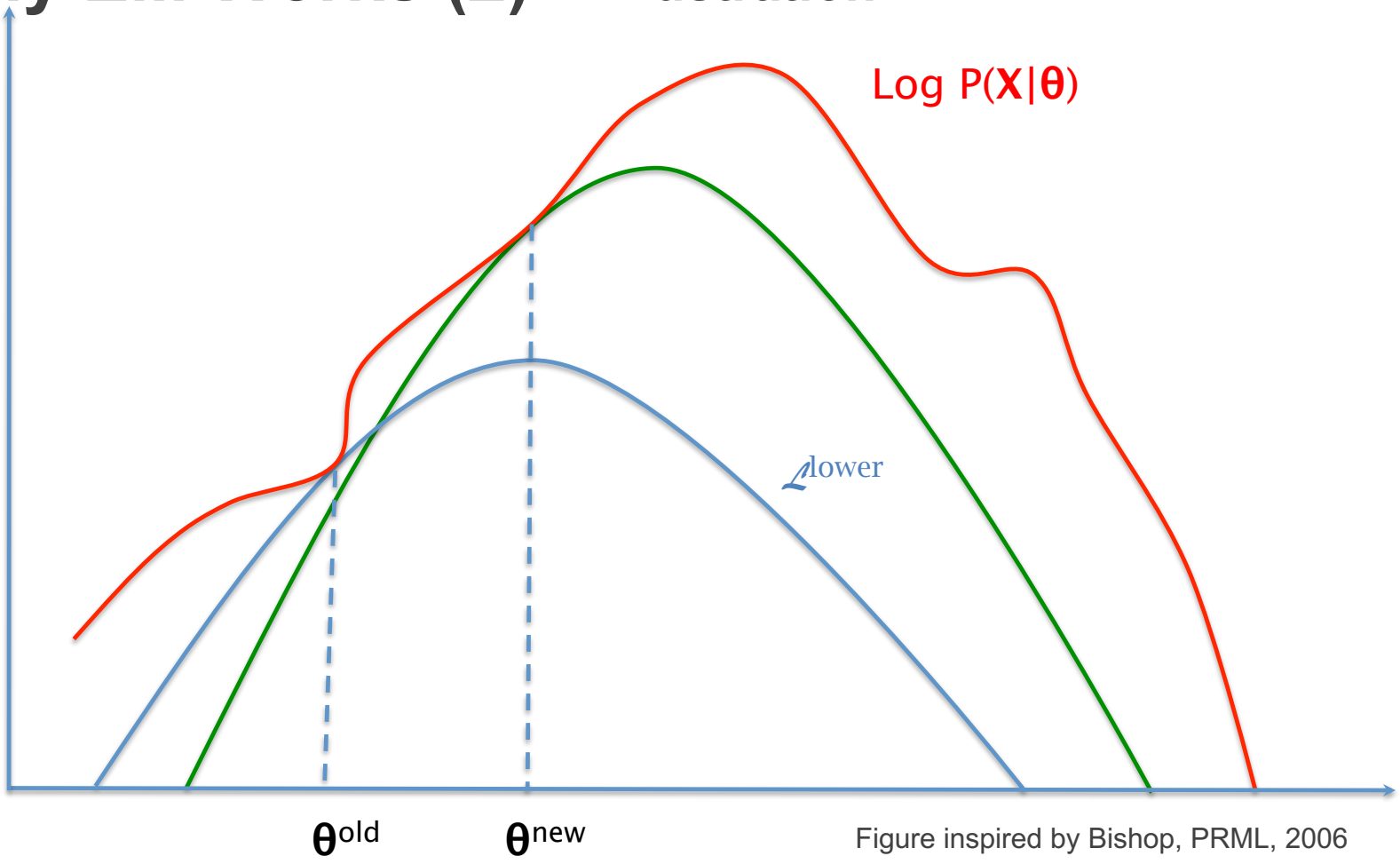


Figure inspired by Bishop, PRML, 2006

Iteratively, compute lower bound for fixed parameters θ^{old} and then maximize this lower bound obtaining θ^{new}

Relation Between GMMs and K-means

- GMM does soft clustering - we have posterior $p(z_k^n = 1|\mathbf{x})$
- Recall documents example
- K-means does hard clustering
- Consider a GMM model with $\boldsymbol{\Sigma}_k = \sigma^2\mathbf{I}$
  - We obtain a softmax function for the responsibilities:

$$\gamma_k^n = \frac{\pi_k \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}^n - \boldsymbol{\mu}_k\|^2\right)}{\sum_{j=1}^{K} \pi_j \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}^n - \boldsymbol{\mu}_j\|^2\right)}$$

- The (hard) max is obtained by making $\sigma^2 \to 0$
  - Then $\gamma_k^n \to z_k^n$ (in the K-means algorithm, i.e. hard assignment)
  - Same update for the centres
  - Optimizing complete data log-likelihood same as optimizing error function in K-means

K-means is the hard limit of GMMs

GMMs and EM
Final Remarks

Singularities In GMMs: Consider the simple case $\boldsymbol{\Sigma}_k = \sigma_k^2 \mathbf{I}$

- Centre one of the components to a single datapoint $\boldsymbol{\mu}_k = x^n$ and Make $\sigma_k \rightarrow 0$
- The log-likelihood will tend to infinity!
- Can use heuristics or be Bayesian (priors over variances)

Identifiability: K! ways to assign K parameters to K components

- These solutions will have the same likelihood
- Problems with interpretability (not with density estimation)
- Possible initial confusion as components try to explain same data and symmetry eventually is broken

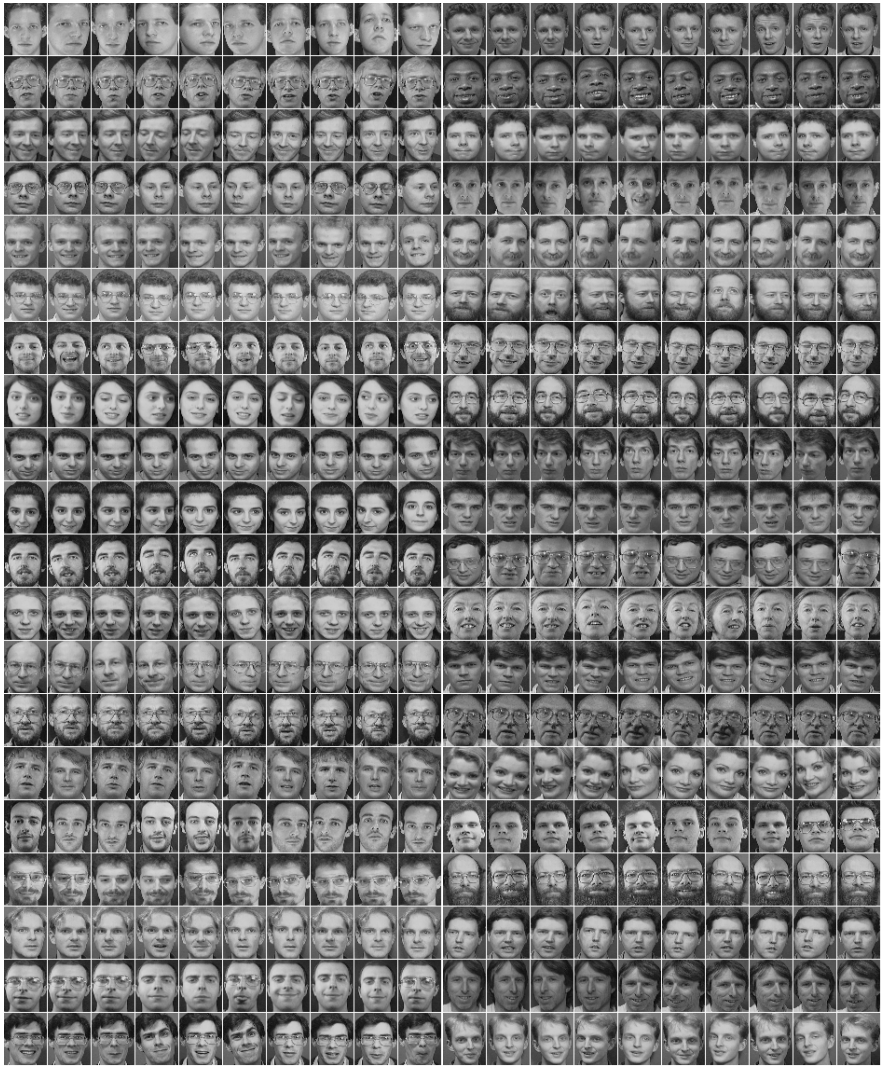
Local optima: EM can get stuck in local optima

- susceptible to different initializations (we can use K-means)

III. Probabilistic Classification: Generative Approaches – Naïve Bayes

Classification Problems

Facial Recognition



Examples of faces and their identity

What is her identity?

Classification Problems

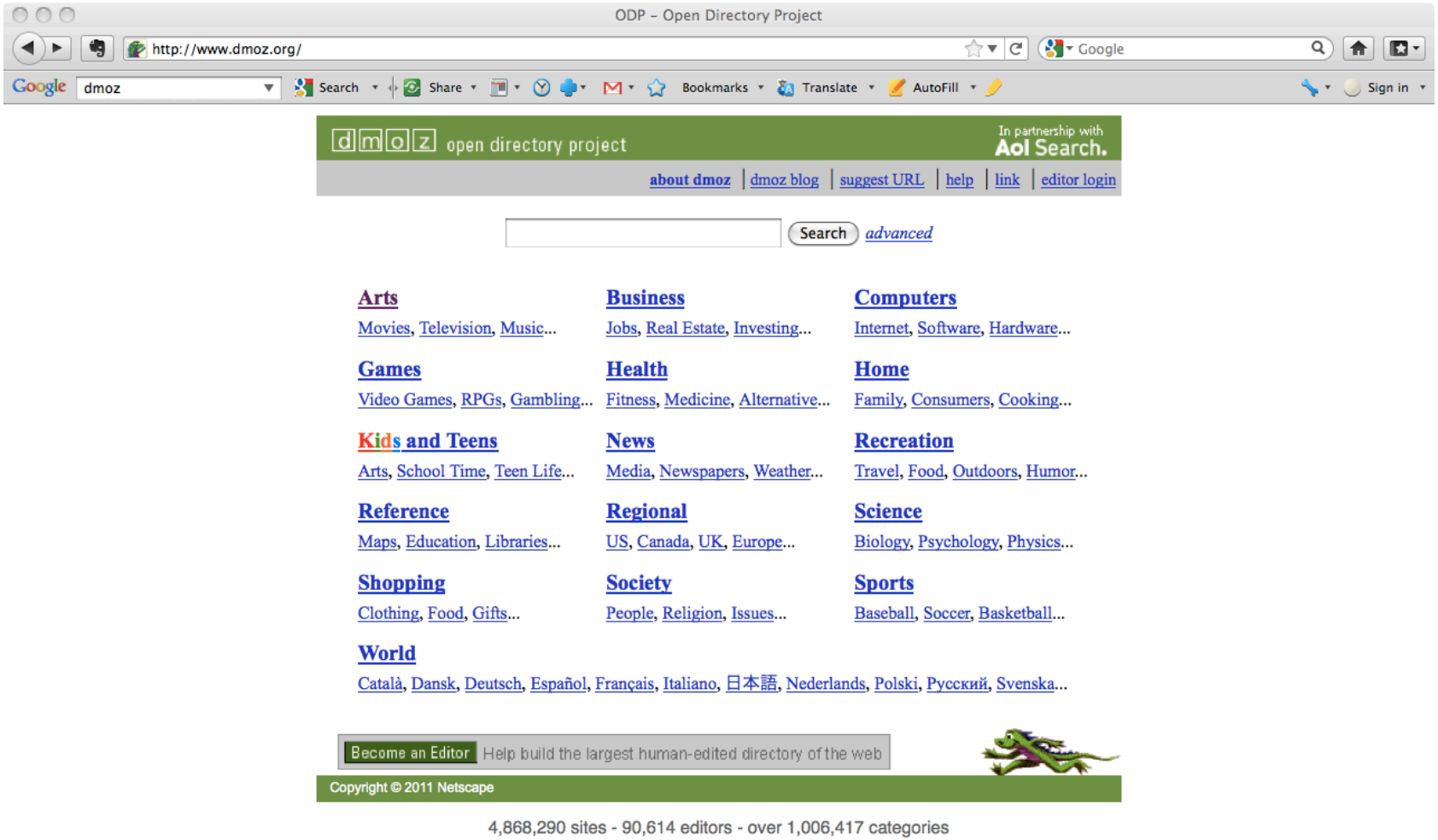
Handwritten Digit Recognition

Examples of images and their corresponding digit

What is the number in the image?

Classification Problems

Supervised Document Classification



Difference with clustering search engines example?

The Classification Problem

Problem Definition (Refresher)

In all previous examples we are dealing with discrete targets.

Given a set of input-output pairs $\mathcal{D} = \{\mathbf{x}^n, y^n\}_{n=1}^N$

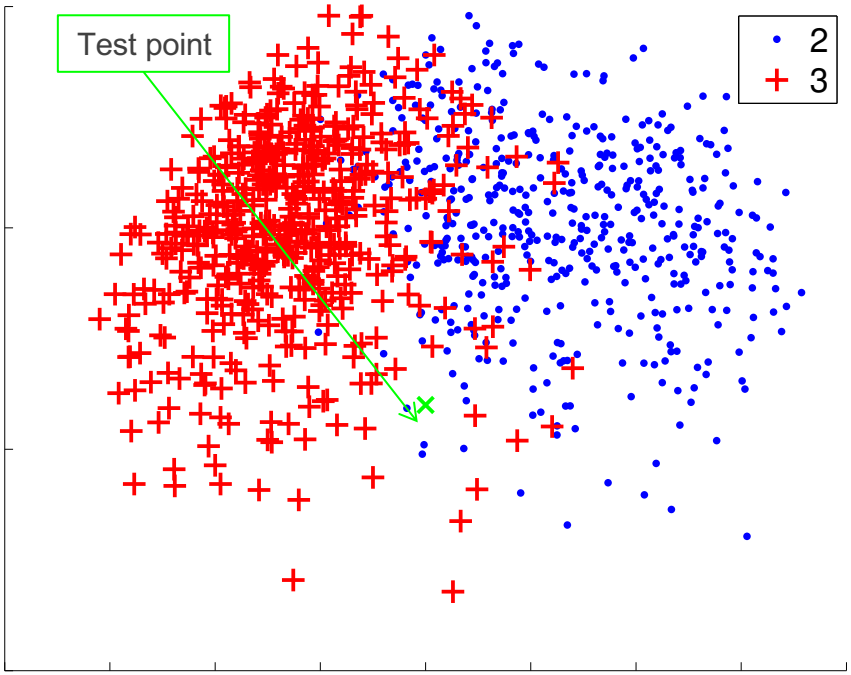
- **x**: D-dimensional feature vector
- y $\in \{\mathcal{C}_1, \ldots, \mathcal{C}_M\}$

Goal: Learn a mapping

- f (**x**) : **x** $\rightarrow$ y
- Make predictions at unseen datapoints **x***.

Probabilistic methods

- Confidence on our predictions
  - Decision theory
  - Reject option



Decision Theory for Classification

- A probabilistic classifier provides an elegant framework for decision theory
- Suppose that we have a predictive probability p(y = c | **x**)
- Let $\mathcal{L}(c, c')$ be the loss incurred by making a decision c′ when the true class is c.

Then for a new $\mathbf{x}^*$ we predict c^* that solves:

Expected loss

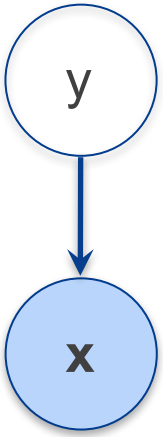
$$c^* = \underset{c'}{\operatorname{argmin}} = \sum_{c} \mathcal{L}(c, c') p(c|\mathbf{x}^*)$$

- For the case of the 0-1 loss (a unit penalty is paid for a misclassification), the optimal decision maximizes p(c|$\mathbf{x}^*$)
  - » This optimal classifier is know as Bayes classifier

How would we make predictions in probabilistic binary classification?

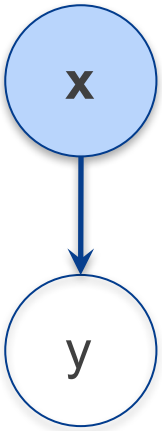
Generative vs. Discriminative Approaches

Generative



- Model the joint p(**x**,y) via models for p(**x**|y) and p(y).
- Predictions p(y|**x**) via Bayes rule
  - Amenable to incorporation of prior knowledge
  - Indirect approach

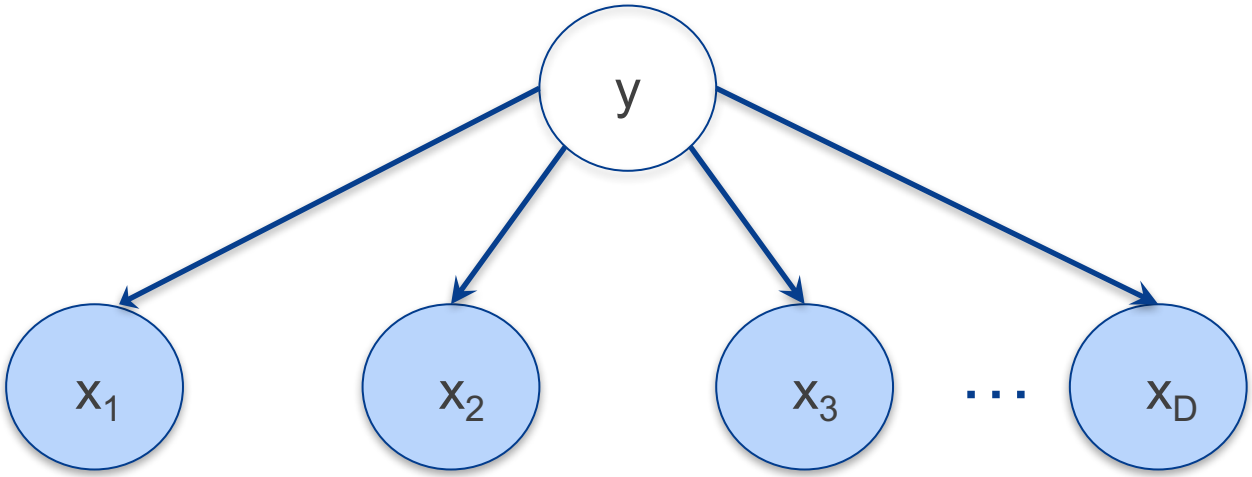
Discriminative



- Model p(y|**x**) directly
- Direct prediction
  - Difficult to incorporate prior information
  - Focus on the task at hand

Generative Approaches

Naïve Bayes



Generative model: p(**x**,y) = p(**x** | y) p(y)

- Where the features are conditionally independent given the class label:

$$p(\mathbf{x}|y) = \prod_{i=1}^{D} p(x_i|y)$$

- Inference problem: Predict the class for a new test point (use Baye's rule):

$$p(y|\mathbf{x}^*) = \frac{p(\mathbf{x}^*|y)p(y)}{\sum_{y'} p(\mathbf{x}^*|y')p(y')}$$ ← Marginal p(**x***)

Naïve Bayes

Binary Classification and Binary Features: Example (1)

- Consider the case of $y \in \{C_1, C_2\}$ and each $x_i \in \{0, 1\}$
  - We want to classify documents as being about sports (C_1) or politics (C_2).

- Bag-of-words
  - Describe each document as a binary vector indicating the presence/absence of a word in a vocabulary V.
  - V = {**f**ootball, **d**efence, **s**trategy, **g**oal, **o**ffice}

- The conditional probability tables(CPTs) are given by:

$$p(\mathcal{C}_1) = 0.5$$

$$\begin{aligned} p(f = 1|\mathcal{C}_1) &= 0.8 & p(f = 1|\mathcal{C}_2) &= 0.1 \\ p(d = 1|\mathcal{C}_1) &= 0.7 & p(d = 1|\mathcal{C}_2) &= 0.7 \\ p(s = 1|\mathcal{C}_1) &= 0.2 & p(s = 1|\mathcal{C}_2) &= 0.8 \\ p(g = 1|\mathcal{C}_1) &= 0.7 & p(g = 1|\mathcal{C}_2) &= 0.3 \\ p(o = 1|\mathcal{C}_1) &= 0.2 & p(o = 1|\mathcal{C}_2) &= 0.7 \end{aligned}$$

Naïve Bayes

Binary Classification and Binary Features: Example (2)

A new document arrives and is described by $\mathbf{x}* = (0, 1, 1, 1, 0)$

- Is it about sports or politics? any guess?

$$p(\mathcal{C}_1|\mathbf{x}^*) = \frac{(0.2)(0.7)(0.2)(0.7)(0.8)}{(0.2)(0.7)(0.2)(0.7)(0.8) + (0.9)(0.7)(0.8)(0.3)(0.3)}$$
$$\approx 0.26$$

- What happened with the priors $p(C_1)$ and $p(C_2)$?
- If were to minimize the 0-1 loss we would classify this document as politics.

- How can we learn the model parameters (i.e. the CPTs) from data?
  - $\theta_i^y \stackrel{\text{def}}{=} p(x_i = 1|y)$ and $\theta^y \stackrel{\text{def}}{=} p(y)$ for y $\in$ {C_1, C_2} and i = 1, …, D
- Intuitively, what would be reasonable estimates for these parameters?

Naïve Bayes

Parameter Estimation via Maximum Likelihood (1)

At training time we observe $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$

- Assuming iid observations, the data log-likelihood is:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(\mathbf{x}^n, y^n) = \sum_{n=1}^{N} \log p(y^n) + \sum_{n=1}^{N} \log p(\mathbf{x}^n | y^n)$$

To optimise $\mathcal{L}(\boldsymbol{\theta})$ wrt $\theta^y \stackrel{\text{def}}{=} p(y)$ we need to maximise:

$$\mathcal{L}(\boldsymbol{\theta}^y) = \sum_{n=1}^{N} \log p(y^n) = N^{\mathcal{C}_1} \log \theta^{\mathcal{C}_1} + N^{\mathcal{C}_2} \log \theta^{\mathcal{C}_2} \quad \text{s.t.} \sum_{y \in \{\mathcal{C}_1, \mathcal{C}_2\}} \theta^y = 1$$

- Where $N^{\mathcal{C}_j}$ is the number of times class $\mathcal{C}_j$ occurs in the training data

- The solution to the above optimisation problem is: $\widehat{\theta}^{\mathcal{C}_1} = \dfrac{N^{\mathcal{C}_1}}{N}$

Naïve Bayes

Parameter Estimation via Maximum Likelihood (2)

To optimize $\mathcal{L}(\boldsymbol{\theta})$ wrt $\theta_i^y \stackrel{\text{def}}{=} p(x_i = 1|y)$ we need to consider only:

$$\mathcal{L}(\boldsymbol{\theta}_i^y) = \sum_{n=1}^{N} \log p(\mathbf{x}^n|y^n)$$

- Due to the independence assumption and binary nature of x_i:

$$p(\mathbf{x}|y) = \prod_{i=1}^{D} (\theta_i^y)^{x_i} (1 - \theta_i^y)^{1-x_i}$$

- Therefore, we want to maximise (wrt each θ_i^y):

$$\sum_n \sum_i \left(x_i^n \log \theta_i^{y^n} + (1 - x_i^n) \log(1 - \theta_i^{y^n}) \right)$$

- Which yields $\theta_i^{\mathcal{C}_j} = p(x_i = 1|y = \mathcal{C}_j) = \frac{\#(x_i = 1, \mathcal{C}_j)}{N^{\mathcal{C}_j}}$

- Where $\#(x_i = 1, \mathcal{C}_j)$ is the number of times x_i=1 for class $\mathcal{C}_j$

Naïve Bayes
Additional Remarks

- Naive Bayes is a simple, computationally efficient yet effective method in many practical applications
  - e.g. document classification
  - Conditional independence (not marginal independence)
  - Linear decision boundary

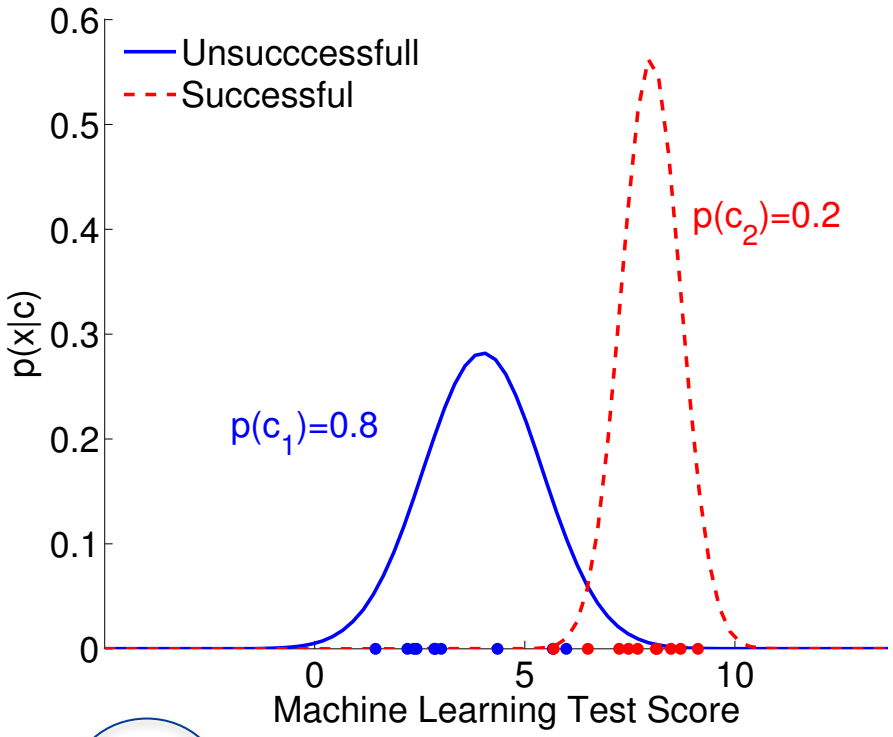
- Maximum likelihood estimates may be poor if data is very sparse
  - What is p(sport | **x***) if the word *goal* is never seen in training (and the new document does contain the world goal)?
  - Need to "smooth" estimates or be Bayesian

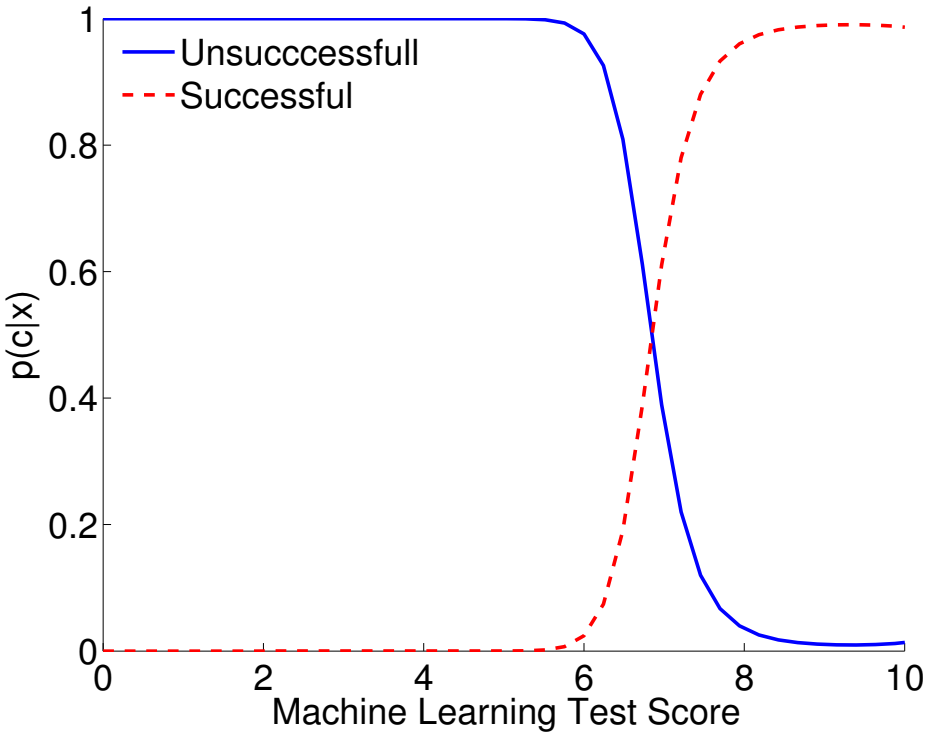
- What about features with more than two states?
  - Need to consider a categorical distribution
  - What parameter estimates would we get?

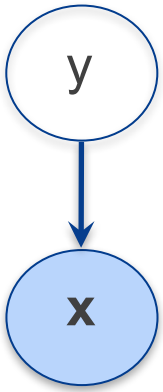
- What about continuous features?
  - Need a model for $p(x_i \mid C_j)$, Gaussian?

Generative Approaches

Class-conditional Gaussian Models







Generative: Modelling p(**x**,y) via p(**x**|y) and p(**y**)

Predictions via Bayes rule p(y | **x**) α p(**x**|y) p(**y**)

- Need to model the density p(**x**|y)

Class-conditional Gaussian Models

Inference and Learning

- Consider the binary case of y $\in$ {C_1, C_2}:

- Learning: Fitting a Gaussian to each class: How?

$$p(\mathbf{x}|\mathcal{C}_1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \qquad p(\mathbf{x}|\mathcal{C}_2) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

  - And estimating p(C_1), p(C_2) How?

- Predictions: For the 0-1 loss, we can compare the log of the (un-normalised posteriors):

$$\log p(\mathbf{x}^*|\mathcal{C}_1) + \log p(\mathcal{C}_1) > \log p(\mathbf{x}^*|\mathcal{C}_2) + \log p(\mathcal{C}_2)$$

- This gives us a quadratic decision boundary

Class-conditional Gaussian Models
Additional Remarks

- High-dimensional data Need to evaluate $\log|\Sigma_j|$ and $\mathbf{\Sigma}_j^{-1}$
  - What can we do?
- How can we model more complex densities? i.e. to address multi-modality?
- Relation to Naïve Bayes: For continuous variables in Naïve Bayes
  - you can make: $p(x_i|C_j) = N(x_i \mid \mu_{ji}, \sigma_{ji})$.
- This is equivalent to having a class-conditional Gaussian model with:
  - $p(\mathbf{x} \mid C_j) = N(\mathbf{x} \mid \boldsymbol{\mu}_j, \mathbf{\Sigma}_j)$ where $\mathbf{\Sigma}_j = \text{diag}(\sigma_{j1},\ldots,\sigma_{jD})$

Summary & Conclusions

- Popular machine learning methods can be seen as probabilistic graphical models
- PCA vs. PPCA
- K-means vs GMMs
- Naïve Bayes

- Discriminative vs generative

- Inference → prediction

- Learning → parameter estimation
  - All data visible (maximum likelihood)
  - Hidden variables (EM)

Research Projects at all Levels

- I'm looking for outsanding students to work with me in research projects (PhD, Masters, Honours, fun)
  - Creative
  - Strong math and stats background (or willing to climb the hill)
  - Excellent programming skills
- Project on *structured prediction with deep learning*, selected for the prestigious 2018 UNSW Scientia Scholarship Scheme
  - Stipend of $40,000 per annum for 4 years and a support package of up to $10,000 per annum
- Other topics:
  - Bayesian deep learning, probabilistic programming, reinforcement learning
  - Spatio-temporal modelling, causal inference, inverse problems
  - Privacy-preserving machine learning
- For a sample of my research look at http://ebonilla.github.io/
  - Drop me an email if interested (e.bonilla@unsw.edu.au)