Introduction to Statistical Machine Learning

Christfried Webers

Statistical Machine Learning Group
NICTA
and
College of Engineering and Computer Science
The Australian National University

Canberra
February – June 2013

(Many figures from C. M. Bishop, "Pattern Recognition and Machine Learning")

ISML 2013

Outlines

Part XVI

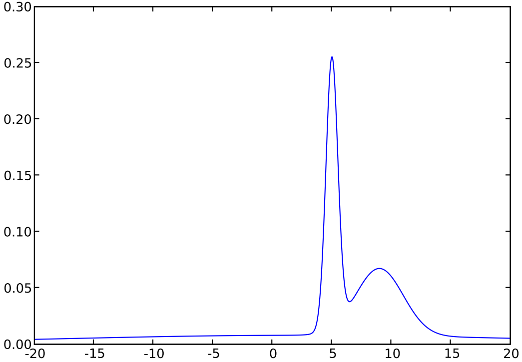
Mixture Models and EM 1

Mixture Models and EM

- We studied how to define joint distributions over observed and latent variables (e.g. graphical models).
- Distribution over observed variables via marginalisation.
- Complex marginal distributions over observed variables can be expressed via more tractable joint distributions over the expanded space of observed and latent variables.
- Mixture Models can also be used to cluster data.
- General technique for finding maximum likelihood estimators in latent variable models: expectation-maximisation (EM) algorithm.
- Later: Variational Interference (Bayesian treatment).

Example - Wallaby Distribution

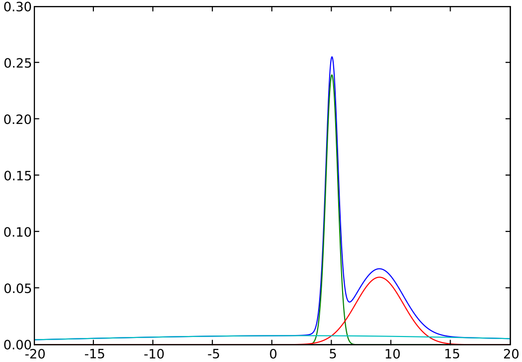
- Introduced very recently to show ...



Example - 'Wallaby' Distribution

- ...that already a mixture of three Gaussian can be fun.

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

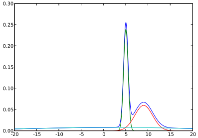


Example - 'Wallaby' Distribution

- Use μ, σ as latent variables and define a distribution

$$p(\mu, \sigma) = \begin{cases} \frac{3}{10} & \text{if } (\mu, \sigma) = (5, 0.5) \\ \frac{3}{10} & \text{if } (\mu, \sigma) = (9, 2) \\ \frac{4}{10} & \text{if } (\mu, \sigma) = (2, 20) \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{aligned} p(x) &= \int_{-\infty}^{\infty} \int_{0}^{\infty} p(x, \mu, \sigma) \, \mathrm{d}\mu \, \mathrm{d}\sigma \\ &= \int_{-\infty}^{\infty} \int_{0}^{\infty} p(x \,|\, \mu, \sigma) \, p(\mu, \sigma) \, \mathrm{d}\mu \, \mathrm{d}\sigma \\ &= \frac{3}{10} \mathcal{N}(x \,|\, 5, 0.5) + \frac{3}{10} \mathcal{N}(x \,|\, 9, 2) + \frac{4}{10} \mathcal{N}(x \,|\, 2, 20) \end{aligned}$$



K-means Clustering

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

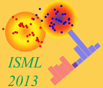
K-means Clustering

- Given a set of data $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ where $\mathbf{x}_n \in \mathbb{R}^D$, $n = 1, \dots, N$.
- Goal: Partition the data into K clusters.
- Each cluster contains points close to each other.
- Introduce a prototype $\boldsymbol{\mu}_k \in \mathbb{R}^D$ for each cluster.
- Goal: Find
  1. a set prototypes $\boldsymbol{\mu}_k$, $k = 1, \dots, K$, each representing a different cluster.
  2. an assignment of each data point to exactly one cluster.

K-means Clustering - The Algorithm

- Start with arbitrary chosen prototypes $\boldsymbol{\mu}_k$, $k = 1, \dots, K$.
  1. Assign each data point to the closest prototype.
  2. Calculate new prototypes as the mean of all data points assigned to each of them.
- In the following, we will formalise this introducing a notation which will be useful later.

K-means Clustering - Notation



- Binary indicator variables

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

  using the 1-of-K coding scheme.
- Define a distortion measure

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

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

K-means Clustering - Notation

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

K-means Clustering - Notation

- Find the values for $\{r_{nk}\}$ and $\{\boldsymbol{\mu}_k\}$ so as to minimise J.
- But $\{r_{nk}\}$ depends on $\{\boldsymbol{\mu}_k\}$, and $\{\boldsymbol{\mu}_k\}$ depends on $\{r_{nk}\}$.
- Iterate until no further change
  1. Minimise J w.r.t. r_{nk} while keeping $\{\boldsymbol{\mu}_k\}$ fixed,

$$r_{nk} = \begin{cases} 1, & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0, & \text{otherwise.} \end{cases} \qquad \forall n = 1, \dots, N$$

     Expectation step

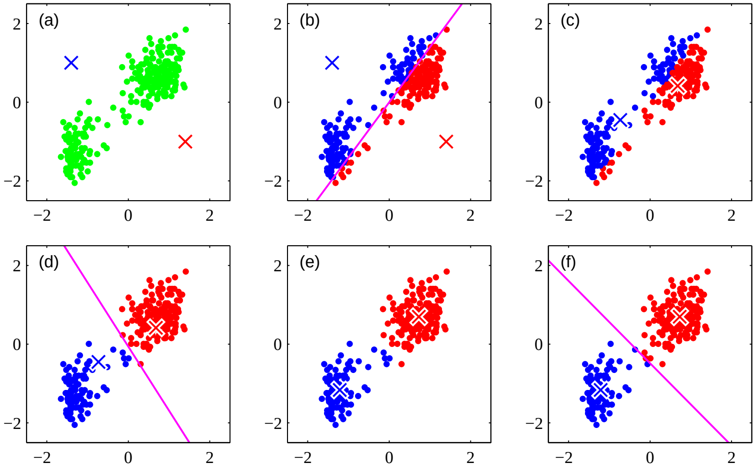
  2. Minimise J w.r.t. $\{\boldsymbol{\mu}_k\}$ while keeping r_{nk} fixed,

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

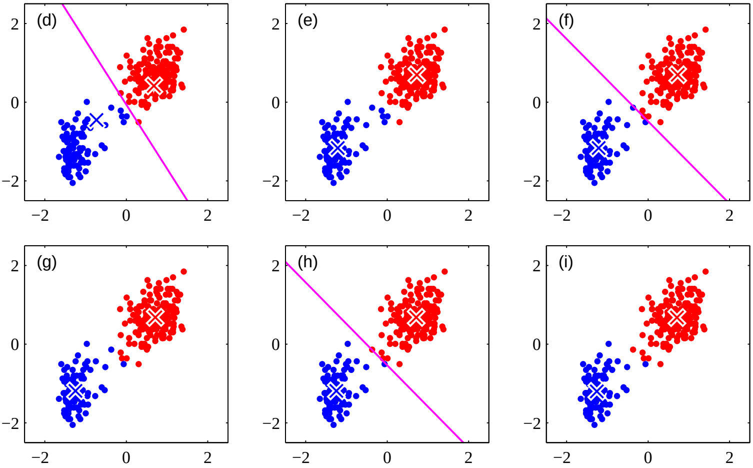
$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} r_{nk} \mathbf{x}_n}{\sum_{n=1}^{N} r_{nk}}$$

     Maximisation step

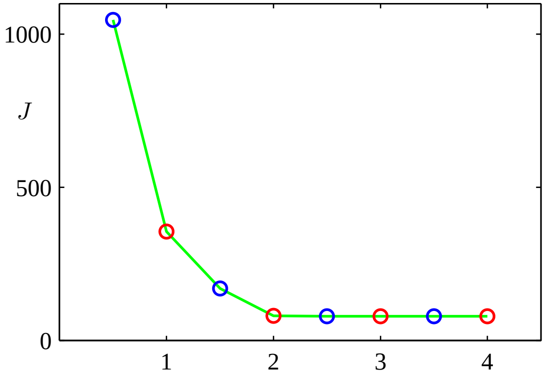
K-means Clustering - Example



K-means Clustering - Example



K-means Clustering - Cost Function



Cost function J after each E step (blue points) and M step (red points).

K-means Clustering - Notes

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

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

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

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

K-means Clustering - Image Segmentationd

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

Illustrating K-means Clustering - Segmentation

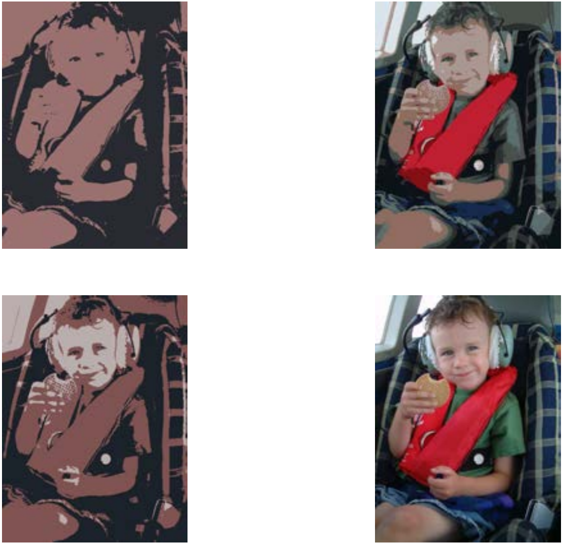
$K = 2$

$K = 10$

$K = 3$

Original image

Illustrating K-means Clustering - Segmentation



K-means Clustering - Compression

- Lossy data compression: accept some errors in the reconstruction as trade-off for higher compression.
- Apply K-means to the data.
- Store the code-book vectors $\boldsymbol{\mu}_k$.
- Store the data in the form of references (labels) to the code-book. Each data points has a label in the range $[1, \dots, K]$.
- New data points are also compressed by finding the closest code-book vector and then storing only the label.
- This technique is also called vector quantisation.

Illustrating K-means Clustering - Compression

$K = 2$

4.2%

$K = 3$

8.3%

$K = 10$

16.7%

Original image

100 %

Mixture of Gaussians

- A Gaussian mixture distribution is a linear superposition of Gaussians of the form

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

- As $\int p(\mathbf{x}) \, \mathbf{dx} = 1$, if follows $\sum_{k=1}^{K} \pi_k = 1$.
- Let us write this with the help of a latent variable $\mathbf{z}$.

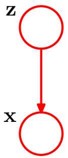
Definition (Latent variables)

Latent variables (as opposed to observable variables), are variables that are not directly observed but are rather inferred (through a mathematical model) from other variables that are observed and directly measured. They are also sometimes called hidden variables, model parameters, or hypothetical variables.

Mixture of Gaussians

- Let $\mathbf{z} \in \{0, 1\}^K$ and $\sum_{k=1}^{K} z_k = 1$. In words, $\mathbf{z}$ is a K-dimensional vector in 1-of-K representation.
- There are exactly K different possible vectors $\mathbf{z}$ depending on which of the K entries is 1.
- Define the joint distribution $p(\mathbf{x}, \mathbf{z})$ in terms of a marginal distribution $p(\mathbf{z})$ and a conditional distribution $p(\mathbf{x} \,|\, \mathbf{z})$ as

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})\, p(\mathbf{x} \,|\, \mathbf{z})$$



Mixture of Gaussians

- Set the marginal distribution to

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

where $0 \leq \pi_k \leq 1$ together with $\sum_{k=1}^{K} \pi_k = 1$.

- Because $\mathbf{z}$ uses 1-of-K coding, we can also write

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

- Set the conditional distribution of $\mathbf{x}$ given a particular $\mathbf{z}$ to

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

or

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

Mixture of Gaussians

- The marginal distribution over $\mathbf{x}$ is now found by summing the joint distribution over all possible states of $\mathbf{z}$

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

- The marginal distribution of $\mathbf{x}$ is a Gaussian mixture.
- For several observations $\mathbf{x}_1, \dots, \mathbf{x}_N$ we need one latent variable $\mathbf{z}_n$ per observation.
- What have we gained? Can now work with the joint distribution $p(\mathbf{x}, \mathbf{z})$. Will lead to significant simplification later, especially for EM algorithm.

Mixture of Gaussians

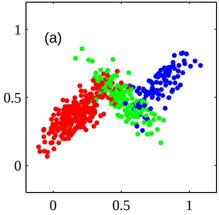
- Conditional probability of $\mathbf{z}$ given $\mathbf{x}$ by Bayes' theorem

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

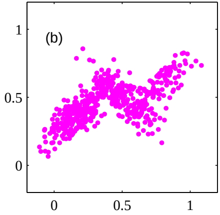
- $\gamma(z_k)$ is the responsibilty of component k to 'explain' the observation $\mathbf{x}$.

Mixture of Gaussians - Ancestral Sampling

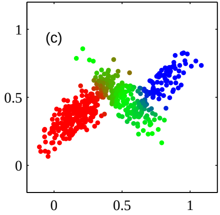
- Goal: Generate random samples distributed according to the mixture model.
  1. Generate a sample $\widehat{\mathbf{z}}$ from the distribution $p(\mathbf{z})$.
  2. Generate a value $\widehat{\mathbf{x}}$ from the conditional distribution $p(\mathbf{x} \,|\, \widehat{\mathbf{z}})$.
- Example: Mixture of 3 Gaussians, 500 points.



Original states of $\mathbf{z}$.



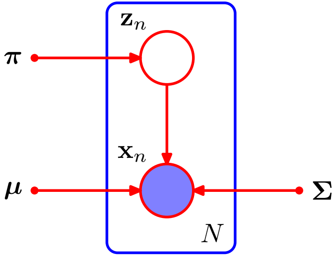
Marginal $p(\mathbf{x})$.



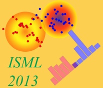
(R, G, B) - colours mixed according to $\gamma(z_{nk})$.

Mixture of Gaussians - Maximum Likelihood

- Given N data points, each of dimension D, we have the data matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$ where each row contains one data point.
- Similarly, we have the matrix of latent variables $\mathbf{Z} \in \mathbb{R}^{N \times K}$ with rows $\mathbf{z}_n^T$.
- Assume the data are drawn i.i.d., the distribution for the data can be represented by a gaphical model.



Mixture of Gaussians - Maximum Likelihood



- The log of the likelihood function is then

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

- Significant problem: If a mean $\boldsymbol{\mu}_j$ 'sits' directly on a data point $\mathbf{x}_n$ then

$$\mathcal{N}(\mathbf{x}_n \,|\, \mathbf{x}_n, \sigma_j^2 \mathbf{I}) = \frac{1}{(2\pi)^{1/2}} \frac{1}{\sigma_j}.$$

- Here we assumed $\boldsymbol{\Sigma}_k = \sigma_k^2 \mathbf{I}$. But problem is general, just think of a main axis transformation for $\boldsymbol{\Sigma}_k$.
- Overfitting (in disguise) occuring again with the maximum likelihood approach.
- Use heuristics to detect this situation and reset the mean of the corresponding component of the mixture.

Mixture of Gaussians - Maximum Likelihood

- A K component mixture has a total of $K!$ equivalent solutions corresponding to the $K!$ ways of assigning K sets of parameters to K solutions.
- Also called identifiability problem. Needs to be considered when the parameters discovered by a model are interpreted.
- Maximising the log likelihood of a Gaussian mixture is more complex then for a single Gaussian. Summation over all K components inside of the logarithm make it harder.
- Setting the derivatives of the log likelihood to zero does not longer result in a closed form.
- May use gradient-based optimisation.
- Or EM algorithm. Stay tuned.