Advanced Machine Learning
Course VI - Mixture Models, EM and Model Order Selection

L. Omar Chehab[(1)] and Frédéric Pascal[(2)]

[(1)] Parietal Team, Inria
[(2)] Laboratory of Signals and Systems (L2S), CentraleSupélec, University Paris-Saclay

l-emir-omar.chehab@inria.fr, frederic.pascal@centralesupelec.fr,

http://fredericpascal.blogspot.fr

Dominante MDS (Mathématiques, Data Sciences)

Sept. - Dec., 2020

CentraleSupélec

Contents

Key references for this course

- Bishop, C. M. *Pattern Recognition and Machine Learning.* Springer, 2006.
- Hastie, T., Tibshirani, R. and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Second edition. Springer, 2009.
- James, G., Witten, D., Hastie, T. and Tibshirani, R. *An Introduction to Statistical Learning, with Applications in R.* Springer, 2013

Course 6.1

Mixture models

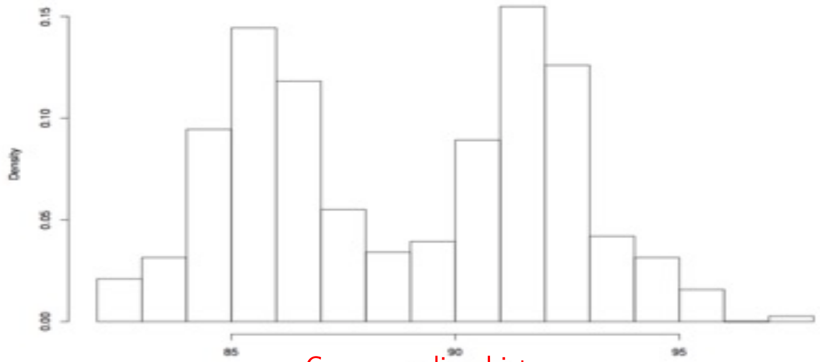
What it is useful for?

- Data-to-knowledge
  - Statistical models fitting ⇒ models learning
  - Features extraction for data, e.g. behavior, shapes...
  - Data characterisation ⇒ Complex modelling
- Complex estimation problems, e.g. many parameters, non parametric estimation...
- Clustering / Classification: Modes ≃ clusters / classes
- Dealing with missing (latent) data: unknown labels can be generalized to unobserved data...

Gaussian Mixture Model

Example: Weight of small animals coming from two different regions

Length	82	83	84	85	86	87	88	89
Observations	5	3	12	36	55	45	21	13
Length	90	91	92	93	94	95	96	98
Observations	15	34	59	48	16	12	6	1



Corresponding histogram

Gaussian Mixture Model with two components

To understand / intuite the process, continue with this simple example

$$\begin{aligned} Y_1 &\sim \mathcal{N}(\mu_1, \sigma_1^2) \\ Y_2 &\sim \mathcal{N}(\mu_2, \sigma_2^2) \\ Z &\sim \mathcal{B}(1, p) \end{aligned}$$

That is $P(Z=1) = p$ and $P(Z=0) = 1-p$. In this context, the observations are as follows:

$$X = Z\, Y_1 + (1-Z)\, Y_2$$

Meanings

data *follows the first distribution / belongs to the first cluster* with a probability p.

Denote $\phi_\theta(x)$ the Gaussian PDF with parameters $\theta = (\mu, \sigma^2)$, one has the following PDF for X: $f_X(x) = p\phi_{\theta_1}(x) + (1-p)\,\phi_{\theta_2}(x)$ leading to the log-likelihood for n observations $(X_1, \dots, X_n)$

$$l(\theta; \mathbf{x}) = \sum_{i=1}^{n} \log\left(p\phi_{\theta_1}(x_i) + (1-p)\,\phi_{\theta_2}(x_i)\right)$$

Gaussian Mixture Model with two components

Difficult estimation problem for $\theta = (p, \theta_1, \theta_2)$, 5 unknown parameters for the simplest case... Problem with the sum in the log.

Solution: consider unobserved latent variables $(Z_1, \ldots, Z_n)$ where $Z_i = 1$ when X_i comes from the first model and $Z_i = 0$ when X_i comes from the second model. Let us now assume we knew the value of each Z_i. In that case, MLEs can be trivially obtained...

$$l(\theta; \mathbf{x}, \mathbf{z}) = \sum_{i=1}^{n} \left(z_i \log(\phi_{\theta_1}(x_i)) + (1 - z_i) \log(\phi_{\theta_2}(x_i)) \right) + \sum_{i=1}^{n} \left(z_i \log(p) + (1 - z_i) \log(1 - p) \right)$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{z} = (z_1, \ldots, z_n)$.

Derive the MLEs pour $\theta = (p, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$!

Gaussian Mixture Model with two components

In practice, the values of the Z_i's are **unknown**!

Idea: Replace for each Z_i, its expected value (conditional to the observed data X_i)

$$\gamma_i(\theta) = E[Z_i|\theta,\mathbf{x}] = P(Z_i = 1|\theta,\mathbf{x})$$

called the responsibility for model 1 of observation i. ⇒ iterative algorithm, Expectation-Maximization (EM) algo

Algorithm (EM algo for two-component Gaussian Mixture)

- *Randomly initialization of* $\theta^{(0)}$
- *Repeat until CV for* $t = 0, 1, \dots$
  - (a) ***E-Step:*** *Compute the responsibilities* $\hat{\gamma}_i = \dfrac{\hat{p}\phi_{\hat{\theta}_1}(x_i)}{\hat{p}\phi_{\hat{\theta}_1}(x_i) + (1-\hat{p})\phi_{\hat{\theta}_2}(x_i)}$, $i = 1, \dots, n$
  - (b) ***M-Step:*** *Compute the parameters...* $\hat{\mu}_1 = \dfrac{\sum_i \hat{\gamma}_i x_i}{\sum_i \hat{\gamma}_i}, \hat{\sigma}_1^2 = \dfrac{\sum_i \hat{\gamma}_i (x_i - \hat{\mu}_1)^2}{\sum_i \hat{\gamma}_i}, \dots$ *and* $\hat{p} = \sum_i \hat{\gamma}_i / n$.

Discussion

Gaussian Mixture Model

Idea: One aims at modelling the statistical behaviour from several populations, groups or classes...

Notations:

- n observations of i.i.d. random variables/vectors, denoted $(X_1, \ldots, X_n)$
- K different clusters containing n_k observations. Of course, $n = \sum_{k=1}^{K} n_k$
- p_k the probability of belonging to the k^{th} class and f_k the PDF of r.v. in this class.

e.g.,:

- different objects in an image (or a patch) containing N pixels, denoted x_i
- population of ducks: x_i corresponds to the size of the i^{th} duck. Different classes corresponding to the animal age/sex/origin (young, old, female, male).
- ...

Gaussian Mixture Model

Statistical modelling of a mixture: with previous notations, one can defined the following PDF:

$$f(x) = \sum_{k=1}^{K} p_k \times f_k(x)$$

Particular case of Gaussian Mixture Models:

$$f(x) = \sum_{k=1}^{K} p_k \times \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right)$$

Problem: estimation of many unknown parameters

$$\theta = \left(p_k, \mu_k, \sigma_k\right)_{k=1,\ldots,K}$$

with $\sum_{k=1}^{K} p_k = 1$ and $\forall k \in \{1,\ldots,K\}, \mu_k \in \mathbb{R}, \sigma_k \in \mathbb{R}_+^*$.

What about K ? Known, unknown ?

Interest of GMM

GMM allow to model many various distributions

(a) $\frac{1}{5}\mathcal{N}(0,1) + \frac{1}{5}\mathcal{N}(1/2,(2/3)^2) + \frac{3}{5}\mathcal{N}(13/15,(5/9)^2)$,

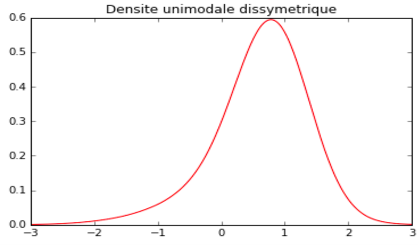
(b) $\sum_{k=0}^{7}\mathcal{N}(3((2/3)^k - 1),(2/3)^{2k})$

(c) $\frac{1}{2}\mathcal{N}(-1,(2/3)^2) + \frac{1}{2}\mathcal{N}(1,(2/3)^2)$

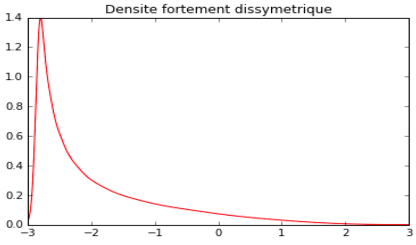
(d) $\frac{3}{4}\mathcal{N}(0,1) + \frac{1}{4}\mathcal{N}(3/2,(1/3)^2)$

(e) $\frac{9}{2}0\mathcal{N}(-6/5,(3/5)^2) + \frac{9}{2}0\mathcal{N}(6/5,(3/5)^2) + \frac{1}{1}0\mathcal{N}(0,(1/4)^2)$

(f) $\frac{1}{2}\mathcal{N}(0,1) + \sum_{k=-2}^{2}\frac{2^{1-k}}{31}\mathcal{N}(k+1/2,(2^{-k}/10)^2)$

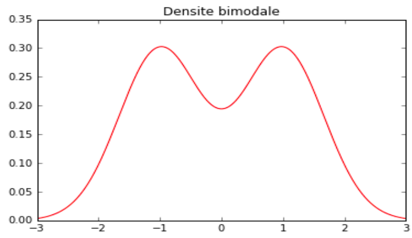


(a) *Asymmetric unimodal PDF*

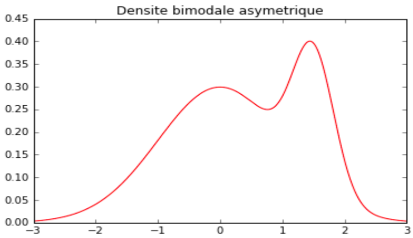


(b) *Strongly asymmetric unimodal PDF*

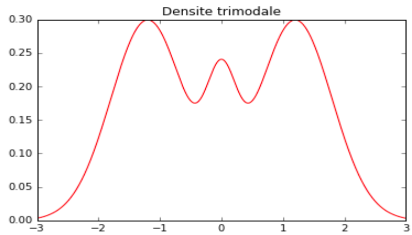
Interest of GMM



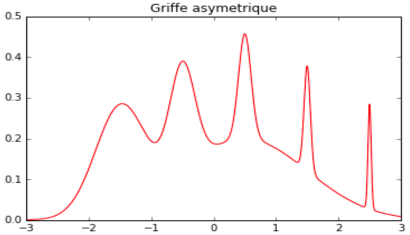
(c) Bimodal PDF



(d) Asymmetric bimodal PDF



(e) Tri-modal PDF



(f) More complex PDF

Reminders in Bayesian probabilities/statistics

For two events (or r. v. ...), one has:

- Conditional probabilities

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

- Bayes rule

$$P(B|A) = \frac{P(A|B)\, P(B)}{P(A)}$$

- if $B_1, \ldots, B_n$ is a partition of Ω, i.e. $\bigcup_{i=1}^{n} B_i = \Omega$ and $\forall i \neq j, B_i \cap B_j = \emptyset$, then

$$P(A) = \sum_{i=1}^{n} P(A \cap B_i)$$

GMM simulations

To simulate the mixture $f(x) = \sum_{k=1}^{K} p_k \times \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right)$, one needs to introduce a latent variable Z (or missing data) that corresponds to the class of the variable X.

Now, the complete data $T = (X, Z)$ is defined by:

- Z follows a discrete distribution $(p_1, \ldots, p_K)$ on $\{1, \ldots, K\}$ such that $\forall k$, one has (Multinomial distribution)

$$P(Z = k) = p_k, \text{ with } \sum_k p_k = 1$$

- $\forall k \in \{1, \ldots, K\}$, conditionally to $\{Z = k\}$, X has a PDF f_k:

$$\mathscr{L}(x|Z = k) = f_k(x)$$

Goal: estimation of $\theta = \left(p_k, \mu_k, \sigma_k\right)_{k=1,\ldots,K}$

2 cases for : one knows latent variables (unrealistic scenario) or not...

EM algorithm - preliminaries

Simple case: Z is known

$\Rightarrow$ one observes $(x_i, z_i)_{i=1,\dots,n}$ instead of (only) $(x_i)_{i=1,\dots,n}$.

Maximum Likelihood approach

Theorem (ML estimates of θ)

Let the observations $(x_i, z_i)_{i=1,\dots,n}$, then $\forall k \in \{1,\dots,K\}$, one has

$$\hat{p}_k = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{z_i=k} \tag{1}$$

$$\hat{\mu}_k = \frac{1}{n\,\hat{p}_k}\sum_{i|z_i=k} x_i \tag{2}$$

$$\hat{\sigma}_k^2 = \frac{1}{n\,\hat{p}_k}\sum_{i|z_i=k} (x_i - \hat{\mu}_k)^2 \tag{3}$$

General EM algorithm - k-means, SEM...

General idea: One only observes $(x_1, \ldots, x_n) \Rightarrow$ analyse the log-likelihood

$$l_{obs}(x_1, \ldots, x_n; \theta) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} p_k \times f_k(x_i)\right), \text{ where } \theta = \left(p_k, \mu_k, \sigma_k\right)_{k=1,\ldots,K}$$

Difficult to maximize!!!

BUT one can make assumptions of the unobserved $(Z_1, \ldots, Z_n)$:

Lemma (Conditional distribution of the Z_i's)

For $\theta \in \Theta, x \in \mathbb{R}$ and $k \in \{1, \ldots, K\}$, one has

$$P_\theta\left(Z = k | X = x\right) = \frac{p_k \times f_k(x)}{\sum_{l=1}^{K} p_l \times f_l(x)} \quad (4)$$

Intuition: thanks to some θ_{old}, one can assign to each x_i some z_i (Lemma) and thanks to previous theorem, one can compute a θ_{new}...

General EM algorithm - k-means, SEM...

Several possible approaches:

- [k-means] Assign a class to each x_i according to

$$z_i = \arg\max_k P_{\theta_{old}}(Z = k | X_i = x_i)$$

  Natural approach but not flexible

- [SEM] *Randomly* assign a class to each x_i according to the distribution

$$P_{\theta_{old}}(Z = . | X_i = x_i)$$

  More flexible

- [N-SEM] *Randomly* assign N classes to each x_i
- [EM] Limit of N-SEM when $N \to \infty$ Very flexible and robust!

k-means

One has to assume that (Very strong assumptions!)

- $p_1 = \ldots = p_K = \dfrac{1}{K}$ and $\sigma_1 = \ldots = \sigma_K$.

Lemma

$\forall \theta, \forall x \in \mathbb{R}$

$$\mathit{arg}\max_k P_\theta\left(Z = k | X = x\right) = \mathit{arg}\min_k |x - \mu_k|$$

Algorithm (k-means)

- *Randomly initialize* $(z_1, \ldots, z_K)$
- *Repeat until CV:*
  - *for* $k \in \{1, \ldots, K\}$, $\mu_k = \dfrac{1}{n} \sum_{i=1}^{n} x_i \, \mathbb{1}_{z_i = k}$
  - *for* $i \in \{1, \ldots, n\}$, $z_i = \mathit{arg}\min_k |x - \mu_k|$

Advantages / Drawbacks ...

Stochastic EM

General idea: Stochastic version of the k-means algorithm...

Algorithm (SEM)

- *Randomly initialize* $(z_1, \ldots, z_K)$
- *Repeat until CV:*
  (a) *Compute*

  $$\hat{\theta} = \arg\max_{\theta} l_{obs}((x_1, z_1), \ldots, (x_n, z_n); \theta)$$

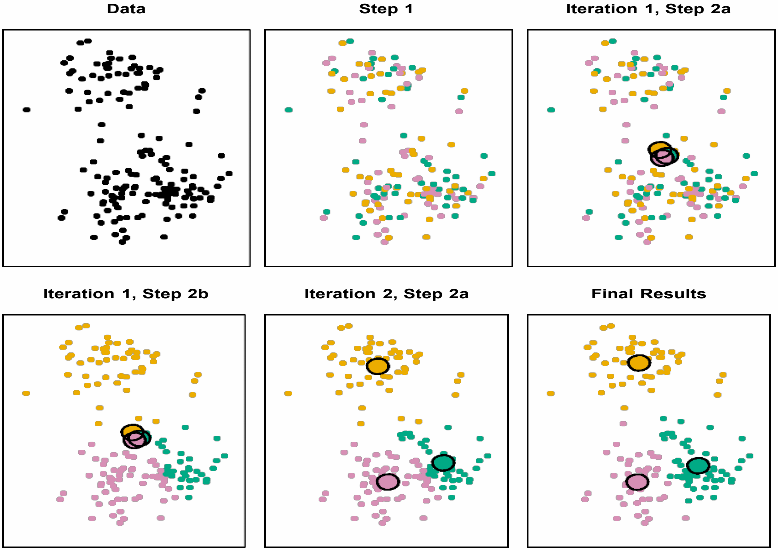
  thanks to Theorem (MLE)

  (b) *for* $i \in \{1, \ldots, n\}$, *randomly choose* z_i *according to*

  $$P_{\hat{\theta}}(Z = .|X_i = x_i)$$

  given by Eq. (4).

Stochastic EM



Stochastic EM - N trials

Algorithm (N-SEM (1))

- *Replicate* N *times, the observations* $(x_1, \ldots, x_n) \to \left(x_i^{(j)}\right)_{1\leq i\leq n, 1\leq j\leq N}$
- *Apply SEM algo to this dataset.*

Algorithm (N-SEM (2))

- *Randomly initialize* N *classes* $z_i^1, \ldots, z_i^N \in \{1, \ldots, K\}, \forall i$
- *Repeat until CV*
  (a) *Compute*
  $$\hat{\theta} = \arg\max_{\theta} l_{obs}\left((x_i, z_i^1)_{i=1,\ldots,n} \cup \ldots \cup (x_i, z_i^N)_{i=1,\ldots,n}; \theta\right)$$
  thanks to Theorem (MLE)
  (b) *for* $i \in \{1, \ldots, n\}$, *randomly choose* $z_i^1, \ldots, z_i^N$ *(independently!) according to*
  $$P_{\hat{\theta}}(Z = .|X_i = x_i)$$
  given by Eq. (4).

Expectation-Maximization algorithm

General idea: N-SEM with $N \to +\infty$ ...

Lemma

Given $(x_i)_{1\le i\le n}$ and associated classes for N trials $(z_i^k)_{1\le i\le n, 1\le k\le K}$, one has

$$\forall\theta, l_{obs}\left(\left(x_i, z_i^1\right)_{i=1,\dots,n} \cup \dots \cup \left(x_i, z_i^N\right)_{i=1,\dots,n}; \theta\right) = \sum_{j=1}^{N} l_{obs}\left(\left(x_i, z_i^j\right)_{i=1,\dots,n}; \theta\right)$$

Theorem (First part)

Given the observations $(x_i)_{1\le i\le n}$ and $\theta_{old} \in \Theta$.

(a) *Let $Z_1, \dots, Z_n$ independent r.v. such that $Z_i \sim \mathcal{L}_{\theta_{old}}(Z|X = x_i)$. One has $\forall\theta = (p_k, \mu_k, \sigma_k)_{1\le k\le K} \in \Theta$,*

$$E[l\left((x_i, z_i)_{i=1,\dots,n}; \theta\right)] = \sum_{i=1}^{n}\sum_{k=1}^{K} P_{\theta_{old}}(Z = k|X = x_i) \log\left(p_k \times f_k(x_i)\right)$$

where $P_{\theta_{old}}(Z = .|X = x_i)$ given by Eq. (4).

Expectation-Maximization algorithm

Theorem (Second part)

Given the observations $(x_i)_{1 \le i \le n}$ *and* $\theta_{old} \in \Theta$,

(b) *One has that* $\underset{\theta}{\operatorname{argmax}} E[l((x_i, z_i)_{i=1,\dots,n}; \theta)]$ *is given by:*

- ***Classes probabilities:*** $\forall k = 1, \dots, K$,

$$p_k^{argmax} = \frac{1}{n} \sum_{i=1}^{n} P_{\theta_{old}}(Z = k | X = x_i)$$

- ***Classes means:*** $\forall k = 1, \dots, K$,

$$\mu_k^{argmax} = \frac{1}{n\, p_k^{argmax}} \sum_{i=1}^{n} P_{\theta_{old}}(Z = k | X = x_i)\, x_i$$

- ***Classes variances:*** $\forall k = 1, \dots, K$,

$$(\sigma_k^{argmax})^2 = \frac{1}{n\, p_k^{argmax}} \sum_{i=1}^{n} P_{\theta_{old}}(Z = k | X = x_i)\, (x_i - \mu_k^{argmax})^2$$

Expectation-Maximization algorithm

Following previous theorem, one has the following theoretical algorithm:

Algorithm (Theory)

- *Randomly initialization of* θ_0
- *Repeat until CV for* $t = 0, 1, \ldots$
  (a) ***E-Step:*** *Compute*

$$L_t(\theta) = E\left[l\left(\left(X_i, Z_i^t\right)_{i=1,\ldots,n}; \theta \right) \right] \left(\Longleftrightarrow Q(\theta, \theta_t) = E\left(l(\theta; \mathbf{t}) | \mathbf{x}, \theta_t\right) \right)$$

  where $Z_1^t, \ldots, Z_n^t$ *are i.i.d. with* $Z_i^t \sim \mathscr{L}_{\theta_t}(Z|X = x_i)$

  (b) ***M-Step:*** *Maximize* $L_t(\theta)$ *to obtain* $\theta_{t+1} = \text{arg}\max_\theta L_t(\theta)$

- **E** for *Expectation*
- **M** for *Maximization*

Outline of the proof...

Expectation-Maximization algorithm

In practice, one has to implement the following algorithm...

Algorithm (Practice)

- *Randomly initialization of* θ_0
- *Repeat until CV for* $t = 0, 1, \ldots$
  (a) ***E-Step:*** *Compute the matrix*
  $$\left[P_{\theta_t}(Z = k | X = x_i)\right]_{1 \le i \le n, 1 \le k \le K} = \left[\frac{p_k^t \times f_{k,t}(x_i)}{\sum_{l=1}^{K} p_l^t \times f_{l,t}(x_i)}\right]_{1 \le i \le n, 1 \le k \le K}$$
  (b) ***M-Step:*** *Compute* θ_{t+1}*, for all* $k = 1, \ldots, K$,
  $$\hat{p}_k^{t+1} = \frac{1}{n}\sum_{i=1}^{n} P_{\theta_t}(Z = k | X = x_i), \tag{5}$$
  $$\hat{\mu}_k^{t+1} = \frac{1}{n\hat{p}_k^{t+1}}\sum_{i=1}^{n} x_i P_{\theta_t}(Z = k | X = x_i) \tag{6}$$
  $$\left(\hat{\sigma}_k^{t+1}\right)^2 = \frac{1}{n\hat{p}_k^{t+1}}\sum_{i=1}^{n} P_{\theta_t}(Z = k | X = x_i)\left(x_i - \hat{\mu}_k^{t+1}\right)^2 \tag{7}$$

A different view - *Maximization-Maximization* procedure

- Consider the function $F(\theta, \mathbf{P}) = E_{\mathbf{P}}[l_0(\theta; \mathbf{t})] - E_{\mathbf{P}}[\log(\mathbf{P}(\mathbf{z}))]$
- $\mathbf{P}$ can be any distribution for the *latent* variables $\mathbf{z}$.
- Note that F evaluated at $\mathbf{P}(\mathbf{z}) = P(\mathbf{z}|\mathbf{x}, \theta)$ is the log-likelihood of the observed data.
- EM algo can be viewed as a joint maximization method for F over θ and $\mathbf{P}(\mathbf{z})$. Maximizer over $\mathbf{P}(\mathbf{z})$ for fixed θ can be shown to be $\mathbf{P}(\mathbf{z}) = P(\mathbf{z}|\mathbf{x}, \theta)$. (dist. computed at the E-step).
- M-step: Maximize $F(\theta, \mathbf{P})$ over θ for fixed $\mathbf{P}(\mathbf{z}), \Longleftrightarrow$ maximizing $E_{\mathbf{P}}[l_0(\theta; \mathbf{t})|\mathbf{x}, \theta^*]$ (2nd term do not depend on θ).

Since $F(\theta, \mathbf{P})$ and the obs. data log-likelihood agree when $\mathbf{P}(\mathbf{z}) = P(\mathbf{z}|\mathbf{x}, \theta)$, maximization of the former accomplishes maximization of the latter.

Course 6.2

Model Order Selection

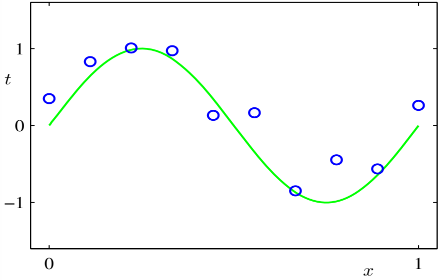
What it is useful for?

- Data-to-knowledge
  - Statistical models fitting $\Rightarrow$ models learning
  - Features extraction for data, e.g. behavior, shapes...
  - Data characterisation $\Rightarrow$ Complex modelling
- Complex estimation problems, e.g. many parameters, non parametric estimation...
- Clustering / Classification: Modes $\simeq$ clusters / classes
- Dealing with missing (latent) data: unknown labels can be generalized to unobserved data...

Introduction / Motivations

Make high-level decisions about the model we want to use:

- Number of components in a mixture model
- Network architecture of (deep) neural networks
- Type of kernel in a support vector machine
- Degree of a polynomial in a regression problem
- Others examples...



True data generated from a sinusoid ($\sin(2\pi x)$) + (small) Gaussian noise (Bishop, 2006)

Introduction / Motivations

Goal: predict the value of t for some new value of x, without knowledge of the green curve → Model selection

Simple / natural approach: curve fitting

$$y(x, \mathbf{w}) = \sum_{i=1}^{M} w_i x^i$$

where M is the polynomial order (unknown) and $\mathbf{w} = (w_0, \ldots, w_M)$ are the polynomial coefficients (unknown). For $\mathbf{w}$, let's minimize an *error function*, e.g.,

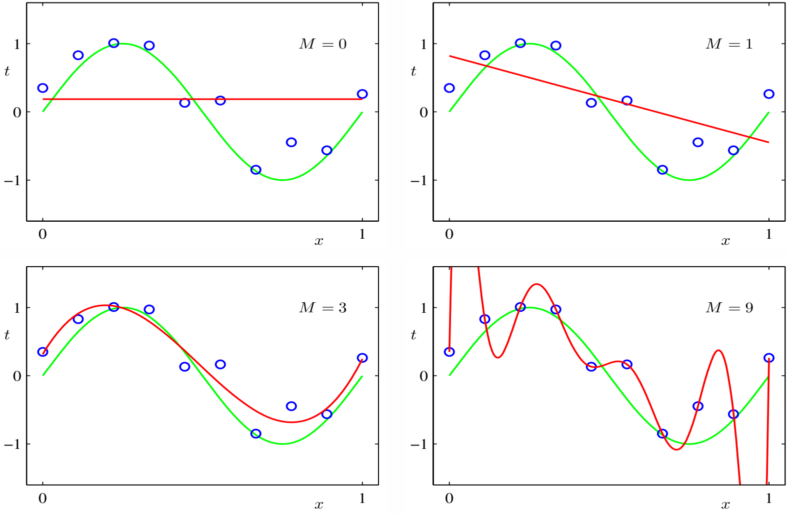
$$e(\mathbf{w}) = \sum_{n=1}^{N} \left(\rho(y(x_n, \mathbf{w}) - t_n)\right)^2 \text{ or } e_{RMS} = \sqrt{e(\mathbf{w}^*)/N}$$

where N stands for the number of observed data. $e(\mathbf{w})$ is a quadratic function w.r.t $\mathbf{w}$ ⇒ unique solution $\mathbf{w}^*$

Problem: choose M!!!

Discussion with mixture models / EM algo

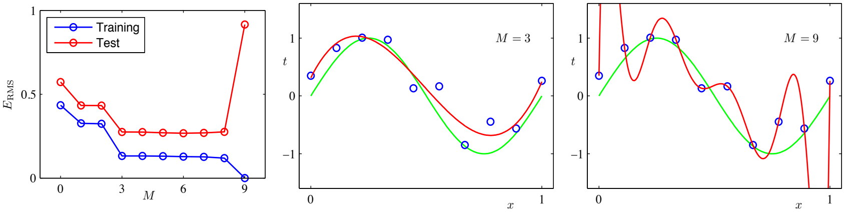
Introduction / Motivations



Polynomial models fitting (Bishop, 2006)

How to evaluate the "best model"?

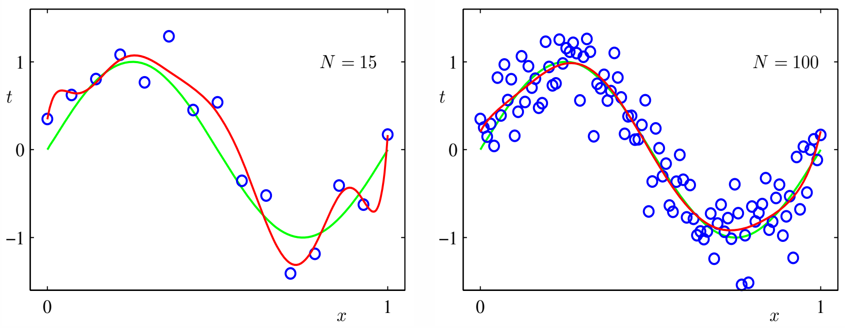
Test vs training data



Different errors behavior between training and test datasets (Bishop, 2006)

- Model fits training data perfectly, but may not do well on test data: Overfitting ($M = 9 \Rightarrow e_{RMS} = 0$, but poor estimation of $\sin(2\pi x)$)
- Training performance ≠ test performance, but we are largely interested in test performance
- Need mechanisms for assessing how a model generalizes to unseen test data: Model selection
- Computational costs ...

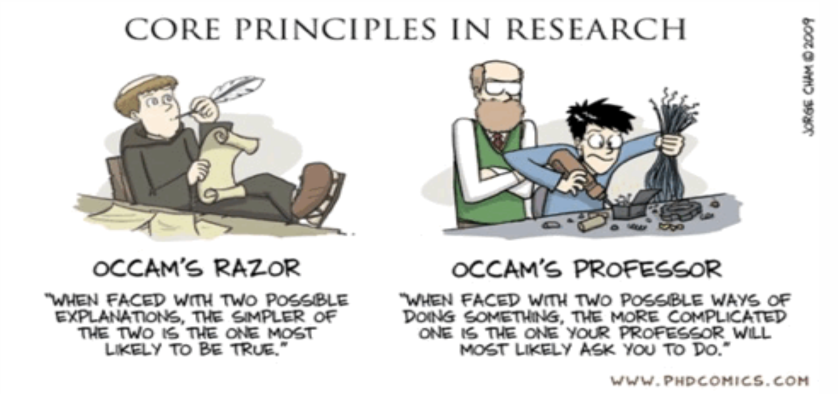
Overfitting / Number of data / Computational cost



Plots of the solutions obtained by minimizing the sum-of-squares error function using the $M = 9$ polynomial for $N = 15$ data points (left plot) and $N = 100$ data points (right plot).

Increasing the size of the data set reduces the over-fitting problem

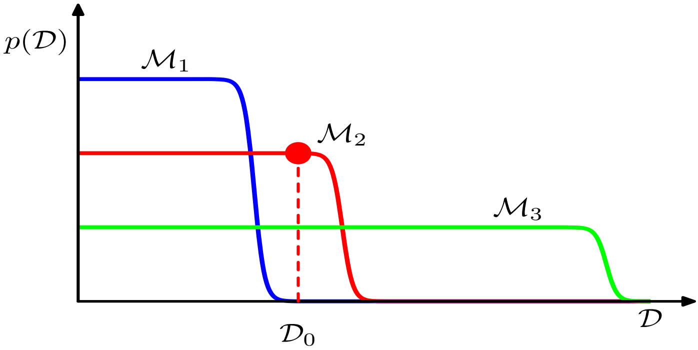
Model choice - Occam's (Ockham) Razor



(PhD comics)

William of Ockham: "More things should not be used than are necessary" (Wikipedia)

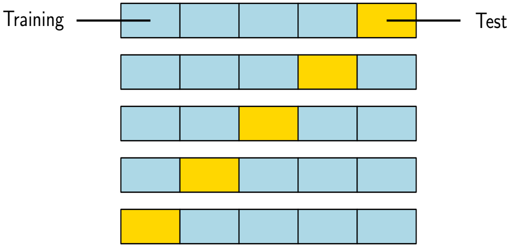
Model choice - Occam's (Ockham) Razor



(Normalized) dist. of data sets for three models of different complexity, in which $\mathscr{M}_1$ is the simplest and $\mathscr{M}_3$ is the most complex - $\mathscr{D}_0$: observed dataset - $\mathscr{M}_2$ with intermediate complexity has the largest evidence (Bishop, 2006)

Idea: choose the simplest model that explains "reasonably" well the data

Cross-validation



- Partition your training data into L subsets
- Train the model on $L-1$ subsets
- Evaluate the model on the remaining subset
- To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.
- Train many models, compare test error

Number of training runs increases with the number of partitions

Information criteria (to be maximized)

- Correct for the bias of MLE by addition of a penalty term to compensate for the overfitting of more complex models (with lots of parameters)
- Akaike Information Criterion (AIC)[1]:

$$AIC(j) = \ln(p(\mathbf{x}|\hat{\theta}_{ML})) - M_j$$

  where M_j is the number of unknown parameters for model $\mathscr{M}_j$.
- Bayesian Information Criterion (BIC) / Minimum Description Length (MDL)[2]:

$$BIC(j) = \ln(p(\mathbf{x}|\hat{\theta}_{ML})) - \frac{1}{2} M_j N$$

  where M_j is the number of unknown parameters for model $\mathscr{M}_j$ and N the number of data.
- BIC penalizes model complexity more heavily than AIC.

[1] H. Akaike. A New Look at the Statistical Model Identification. IEEE Transactions on Automatic Control, 19(6) : 716-723, 1974.

[2] G. E. Schwarz. Estimating the Dimension of a Model. Annals of Statistics, 6(2) : 461-464, 1978.

Bayesian Model Comparison

- Place a prior $p(\mathscr{M})$ on the class of models
- Given a training set $\mathscr{D}$, we compute the posterior distribution over models as

$$p(\mathscr{M}_i|\mathscr{D}) \propto p(\mathscr{M}_i)\, p(\mathscr{D}|\mathscr{M}_i)$$

  which allows us to express a preference for different models
- Model evidence (marginal likelihood):

$$p(\mathscr{D}|\mathscr{M}_i) = \int p(\mathscr{D}|\theta_i)\, p(\theta_i|\mathscr{M}_i)\, d\theta_i$$

- Bayes factor for comparing two models: $p(\mathscr{D}|\mathscr{M}_1)\,/\,p(\mathscr{D}|\mathscr{M}_2)$
- Integral often intractable...

Bayesian Model Averaging

For predicting new observations...

- Place a prior $p(\mathscr{M})$ on the class of models
- Instead of selecting the "best" model, integrate out the corresponding model parameters $\theta_{\mathscr{M}}$ and average over all models $\mathscr{M}_i, i = 1, \ldots, L$

$$
\begin{aligned}
p(\mathscr{D}) &= \sum_{i=1}^{L} p(\mathscr{M}_i)\, p(\mathscr{D}|\theta_i)\, p(\theta_i|\mathscr{M}_i)\, d\theta_i \\
&= \sum_{i=1}^{L} p(\mathscr{M}_i)\, p(\mathscr{D}|\mathscr{M}_i)
\end{aligned}
$$

- Computationally expensive
- Integral often intractable

On Model (Order) Selection ...

- Many others techniques:
  - Minimum Message Length (see applications - Bayesian criterion)
  - Modified AIC accounting for small sample size:

$$mAIC(j) = \ln(p(\mathbf{x}|\hat{\theta}_{ML})) - M_j - \frac{M_j(M_j+1)}{N-M_j-1}$$

  - Hypothesis testing vs Bayesian model comparison
  - ...
- For estimating models mixture:
  - All previous techniques
  - Split and merge[3](see applications + TP)
  - Reversible jump[4](outside of the scope of this course)

Simple example...

[1] Zhang, Z., Chen, C., Sun, J., Chan, K. L. (2003). EM algorithms for Gaussian mixtures with split-and-merge operation. Pattern recognition, 36(9), 1973-1983.
[2] Zhang, Z., Chan, K. L., Wu, Y., Chen, C. (2004). Learning a multivariate Gaussian mixture model with the reversible jump MCMC algorithm. Statistics and Computing, 14(4), 343-355.

Applications to image processing with Mixtures of Asymmetric Generalized Gaussian distributions

`Course 5`

New slides