

02457 Signal Processing in Non-linear Systems: Lecture 7

The EM algorithm and K-means

Ole Winther

Technical University of Denmark (DTU)

October 6, 2016

- Hour 1
 - Neural networks for classification (signal detection)
 - Your turn! Construct neural network for AND and XOR
 - Exercise 6 walk through
- Hour 2
 - Your turn! Exercise 6 quiz
 - Unsupervised learning
 - Estimating a Gaussian distribution from data revisited
 - Mixture of Gaussians (MoG)
- Hour 3
 - Learning with expectation-maximization (EM)
 - Your turn! Derive EM updates for MoG

Next lecture - lecture 8

- Clustering
 - K -means and family
 - Hierarchical clustering (not part of curriculum but still useful)
- Summary likelihoods
 - What is a likelihood function?
 - Which one to use in a given problem?
- Radial basis networks - from $p(\mathbf{x}, t)$ to $p(t|\mathbf{x})$

- Training data $\mathcal{D} = \{(\mathbf{x}_n, t_n) | n = 1, \dots, N\}$
- Likelihood function for **independent identically distributed (iid)** examples, factorizes

$$p(\mathcal{D}|\mathbf{w}) = \prod_{n=1}^N [p(t_n|\mathbf{x}_n, \mathbf{w})p(\mathbf{x}_n|\mathbf{w})] = \underbrace{p(\mathbf{t}|\mathbf{X}, \mathbf{w})}_{\text{supervised}} \underbrace{p(\mathbf{X}|\mathbf{w})}_{\text{unsupervised}}$$

- For regression, we can use **least squares** learning

$$E(\mathbf{w}) = \sum_{n=1}^N (t_n - y(\mathbf{x}_n, \mathbf{w}))^2$$

- More general learning principle maximum likelihood

- Maximum likelihood, that is maximize

$$\log p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \sum_{n=1}^N \log p(t_n|\mathbf{x}_n, \mathbf{w})$$

- New convenient definition of **cost function**

$$E(\mathbf{w}) = -\log p(\mathbf{t}|\mathbf{X}, \mathbf{w})$$

- The *training error per example*

$$e_{\text{tr}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N -\log p(t_n | \mathbf{x}_n, \mathbf{w})$$

- A *good generalizer* assigns high probability to the true output for a given *new input*.
- We define *the generalization error*.

$$\begin{aligned} e_{\text{gen}} &= \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{m=1}^M -\log p(t_m | \mathbf{x}_m, \mathbf{w}) \\ &= \int \int -\log p(t | \mathbf{x}, \mathbf{w}) p(t | \mathbf{x}) dt p(\mathbf{x}) d\mathbf{x} \end{aligned}$$

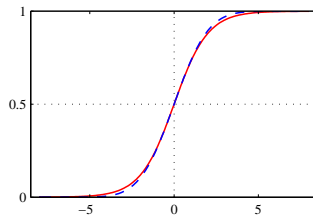
This is the average (expected) error on a test datum (\mathbf{x}, t) .

- Labels two class problem: $t_n = 1$ for class one and $t_n = 0$ for class two
- Logistic regression recap – start with real valued function of inputs:

$$a(\mathbf{x}; \mathbf{w}) = \mathbf{w} \cdot \mathbf{x} + w_0$$

- and apply logistic transformation

$$P(t = 1|\mathbf{x}) = y(\mathbf{x}, \mathbf{w}) = \sigma(a(\mathbf{x}; \mathbf{w})) \quad \text{with} \quad \sigma(a) \equiv \frac{1}{1 + \exp(-a)}$$





Two class problem - cost function

- Labels: $t_n = 1$ for class one and $t_n = 0$ for class two
- Let the network output $y \in [0, 1]$ be the probability of $t = 1$,
- then we can write the likelihood as

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^N p(t_n|\mathbf{x}_n, \mathbf{w}) = \prod_{n=1}^N \left\{ y(\mathbf{x}_n|\mathbf{w})^{t_n} [1 - y(\mathbf{x}_n|\mathbf{w})]^{(1-t_n)} \right\}$$

- and the cost function becomes

$$E(\mathbf{w}) = - \sum_{n=1}^N \{ t_n \log y(\mathbf{x}_n|\mathbf{w}) + (1 - t_n) \log [1 - y(\mathbf{x}_n|\mathbf{w})] \}$$

- This is called the *entropic cost function*

- MLP w linear output: $a(\mathbf{x}; \mathbf{w}) = \mathbf{w}^{(2)} \cdot \mathbf{z}$:

$$y(\mathbf{x}|\mathbf{w}) = \frac{1}{1 + \exp(-a(\mathbf{x}; \mathbf{w}))}$$

- Backprop rule: $\frac{\partial E_n}{\partial w_{jk}} = \delta_{nj} z_{nk}$
- Output unit δ -rule

$$\delta_n = \frac{\partial E_n}{\partial a_n} = \frac{\partial E_n}{\partial y_n} \frac{\partial y_n}{\partial a_n} = \frac{y_n - t_n}{y_n(1 - y_n)} y_n(1 - y_n) = y_n - t_n$$

- Derivative of logistic function:

$$\frac{\partial y_n}{\partial a_n} = \frac{\partial}{\partial a_n} \frac{1}{1 + \exp(-a_n)} = y_n(1 - y_n)$$

- Derivative wrt y :

$$\frac{\partial E}{\partial y_n} = - \left[\frac{t_n}{y_n} - \frac{1 - t_n}{1 - y_n} \right] = \dots = \frac{y_n - t_n}{y_n(1 - y_n)}$$



Multiple classes

- We use $0 \leq y \leq 1$ coding for C classes and we want the outputs to be the posterior probabilities $P(C|\mathbf{x})$, hence they “should sum to one”

$$y_k(\mathbf{x}) = \frac{\exp a_k(\mathbf{x})}{\sum_{k'} \exp a_{k'}(\mathbf{x})}$$

- Targets are represented by ‘1 of K ’-vectors. If **class k** :

$$\mathbf{t} = [0, 0, 0, \dots, \underbrace{1}_k, 0, \dots, 0]$$

- The likelihood function is given by

$$p(\mathbf{t}|\mathbf{x}) = \prod_{k=1}^C y_k(\mathbf{x})^{t_k}$$

Two classes and beyond

- The likelihood and cost function are given by

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}) = \prod_{k=1}^C y_k(\mathbf{x})^{t_k} \quad E = - \sum_n \sum_k t_{nk} \log y_{nk}$$

- The derivatives are relatively simple again

$$\frac{\partial E_n}{\partial \mathbf{a}_k} = \sum_{k'} \frac{\partial E_n}{\partial y_{k'}} \frac{\partial y_{k'}}{\partial \mathbf{a}_k}$$

$$\frac{\partial y_{k'}}{\partial \mathbf{a}_k} = \delta_{kk'} y_k - y_{k'} y_k$$

$$\frac{\partial E_n}{\partial y_{k'}} = -\frac{t_{k'}}{y_{k'}}$$

$$\frac{\partial E_n}{\partial \mathbf{a}_k} = \sum_{k'} -\frac{t_{k'}}{y_{k'}} (\delta_{kk'} y_k - y_{k'} y_k) = -(t_k - y_k \sum_{k'} t_{k'}) = y_k - t_k$$

Your turn! Neural networks for AND and XOR

- Consider 2d inputs $\mathbf{x} = (x_1, x_2)$.
- Represent AND and XOR in truth table & graphically (2d)
- The **decision boundary** is defined as those points in input space with $p(t = 1 | \mathbf{x}, \mathbf{w}) = \frac{1}{2}$
- What is the shape of the decision boundary for **logistic regression**

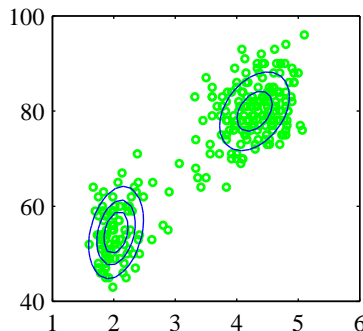
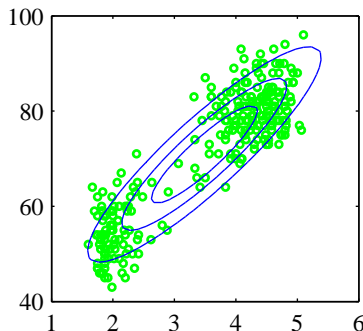
$$P(t = 1 | \mathbf{x}) = y(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w} \cdot \mathbf{x} + w_0)$$

- Try to find \mathbf{w} -values to solve the AND and XOR problems.
- XOR – use hidden layer and two hidden units
- Hint: each hidden unit acts logistic regressor.

- Exercise 6 walk through
- Your turn! Exercise 6 quiz

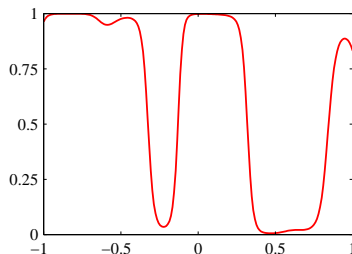
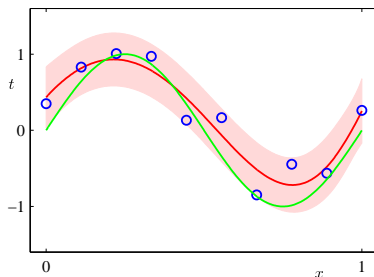
Unsupervised learning

- Learning the distribution of a set of variables $p(\text{input})$.
- Or perhaps just **some important characteristics** of the distribution



Supervised learning

- Learning the **conditional distribution** $p(\text{output}|\text{input})$.
- Regression – output continuous
- Classification – output discrete (e.g. positive diagnosis)



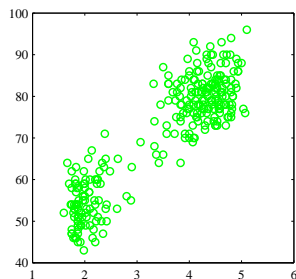


Unsupervised learning task

- **Density estimation**
 - **Compression**, creating compact representation of data
 - **Generative modeling** $P(C_k|\mathbf{x}) \propto p(\mathbf{x}|C_k)p(C_k)$
 - **Outlier detection**, identification
 - In this course only **continuous densities**: Gaussian, mixture of Gaussians and non-parametric (histogram and kernel densities)
- **Clustering**
 - unsupervised classification
 - prototypical summary
- **Feature extraction/visualization** —
 - finding sub-space with most variance (PCA)
 - finding regions with high density (K-means).

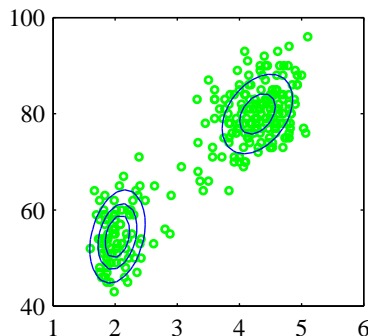
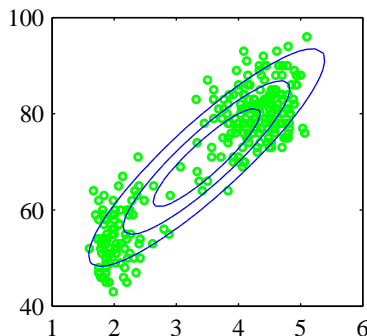
Old Faithful

- Hydrothermal geyser in Yellowstone National Park, Wyoming, USA.

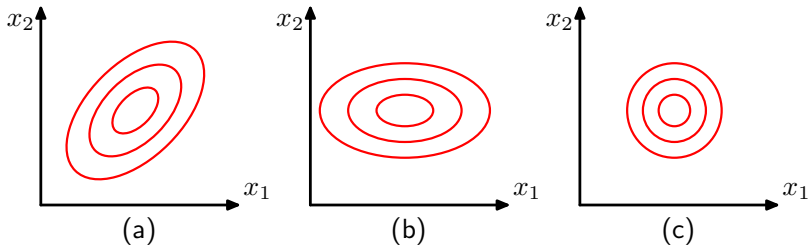


- x-axis duration of eruption in minutes
- y-axis time to next eruption in minutes

Density estimation



Gaussian distribution revisited



$$\Sigma = \begin{pmatrix} \alpha & \gamma \\ \gamma & \beta \end{pmatrix}$$



Maximum likelihood

- Bishop 2.3.4 - show on black board
- Training set: $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_N\}$
- Mean value

$$\mu_{\text{ML}} = \frac{1}{N} \sum_n \mathbf{x}_n$$

- Covariance

$$\Sigma_{\text{ML}} = \frac{1}{N} \sum_n (\mathbf{x}_n - \mu_{\text{ML}})(\mathbf{x}_n - \mu_{\text{ML}})^T$$

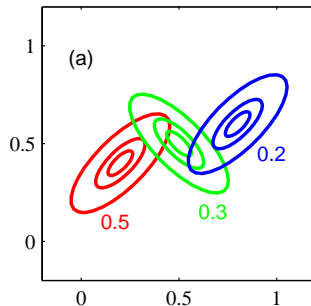
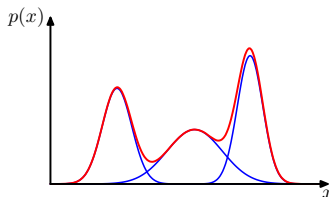
- Overfitting (underestimates covariance):

$$\mathbb{E}_{\mathcal{D}} [\Sigma_{\text{ML}}] = \frac{N-1}{N} \Sigma_{\text{true}}$$

- Mixture modeling – convex combinations of simpler models

$$p(\mathbf{x}) = \sum_{k=1}^K p(k) p(\mathbf{x}|k),$$

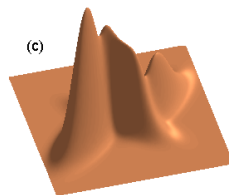
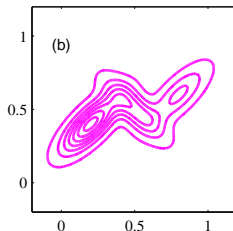
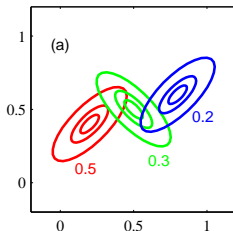
$$\sum_k p(k) = 1$$



Mixture of Gaussians (MoG)

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \mu_k, \Sigma_k),$$

$$\sum_k \pi_k = 1$$



Generative process

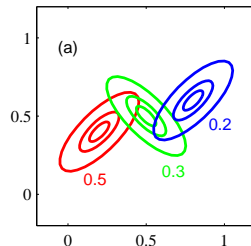
- MoG density

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

$$\sum_k \pi_k = 1$$

- Take a **generative** view of model:

- 1 first draw a component number k with relative probabilities π_k ,
- 2 then draw a random vector \mathbf{x} from the given component with density $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \Sigma_k)$.



Maximum likelihood learning

- The training set is $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_N\}$
- the likelihood function is given by

$$p(\mathbf{X}|\mathbf{w}) = \prod_{n=1}^N p(\mathbf{x}_n|\mathbf{w})$$

- Parameters $\mathbf{w} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$
- The cost function is then (notice sum inside log)

$$\begin{aligned} E(\mathbf{w}) &= -\sum_{n=1}^N \log p(\mathbf{x}_n|\mathbf{w}) = -\sum_{n=1}^N \log \sum_{k=1}^K p(\mathbf{x}_n|\mathbf{w}_k) \pi_k \\ &= -\sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \end{aligned}$$

- We will try to maximize the log likelihood by setting the gradient to zero wrt to the parameters $\{\pi_k, \mu_k, \Sigma_k\}$

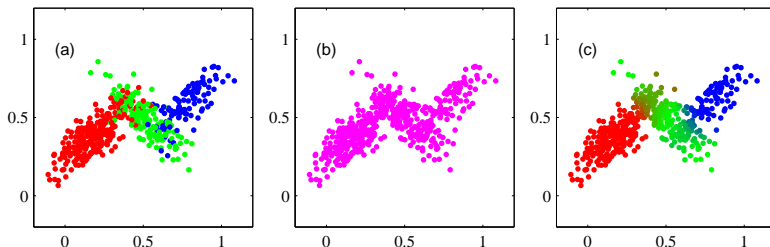
$$E(\mathbf{w}) = - \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \mu_k, \Sigma_k) .$$

- Introduce the so-called **responsibility**

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n | \mu_{k'}, \Sigma_{k'})} \in [0, 1] .$$

Responsibility – soft assignments

$$\begin{aligned}\gamma_{nk} &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})} \\ &= \frac{p(k)p(\mathbf{x}_n | k)}{\sum_{k'} p(k')p(\mathbf{x}_n | k')} = p(k | \mathbf{x}_n) \in [0, 1] .\end{aligned}$$



MoG maximum likelihood - π_k

Derivative wrt π_k of

$$\mathcal{L}(\mathbf{w}, \lambda) = E(\mathbf{w}) + \lambda \left[\sum_{k'=1}^K \pi_{k'} - 1 \right] .$$

Cost function and responsibility

$$E(\mathbf{w}) = - \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})} .$$



MoG maximum likelihood - μ_k

Use (see appendix C)

$$\frac{\partial}{\partial \mu_k} \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k) = \Sigma_k^{-1} (\mathbf{x}_n - \mu_k) \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)$$

Cost function and responsibility

$$E(\mathbf{w}) = - \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)$$

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n | \mu_{k'}, \Sigma_{k'})}.$$

MoG maximum likelihood - Σ_k

Use (see appendix C)

$$\frac{\partial}{\partial \Sigma_k} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k) = -\frac{1}{2} \left[\Sigma_k^{-1} - \Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} \right] \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k)$$

Cost function and responsibility

$$E(\mathbf{w}) = - \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k)$$

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_{k'}, \Sigma_{k'})}$$

E-step - for $n = 1, \dots, N$ and $k = 1, \dots, K$:

$$\gamma_{nk} \leftarrow \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}.$$

M-step - for $k = 1, \dots, K$:

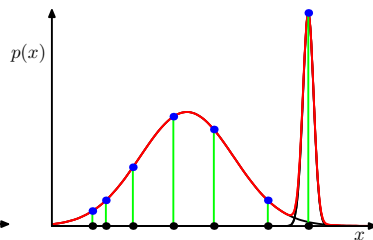
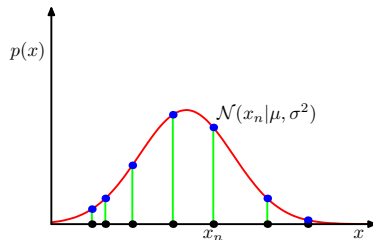
$$N_k \leftarrow \sum_{n=1}^N \gamma_{nk}$$

$$\pi_k \leftarrow \frac{N_k}{N}$$

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

$$\boldsymbol{\Sigma}_k \leftarrow \frac{1}{N_k} \sum_{n=1}^N \gamma_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

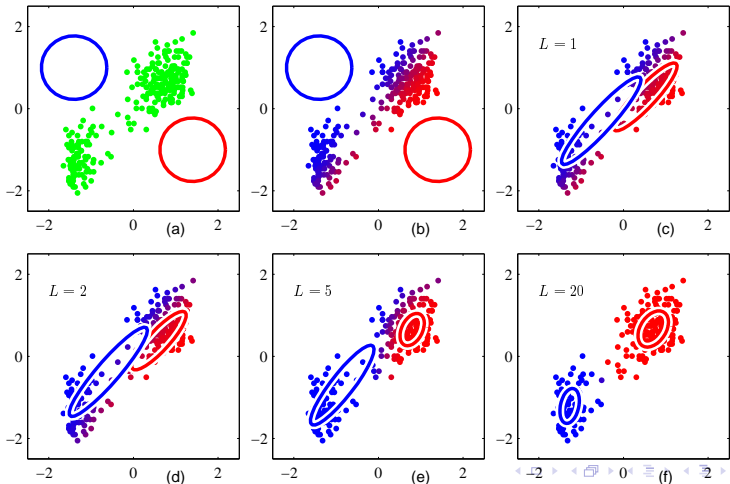
Nature of the maximum likelihood solution



$$E(\mathbf{w}) = - \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

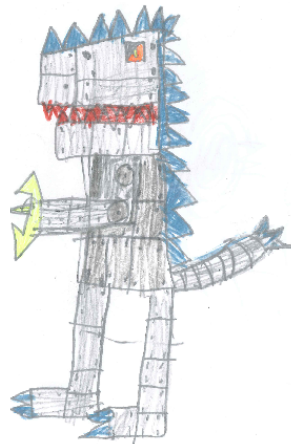
Consider cost when $\mu_k = \mathbf{x}_n$, $\pi_k > 0$ and $\Sigma_k \rightarrow 0$.

MoG for Old Faithful



Summary

- Unsupervised learning task
- Mixture models
- Learning with expectation maximization (EM)



- Gaussian – Bishop 2.3 especially 2.3.4
- Mixture of Gaussians – Bishop 2.3.9
- Mixture models – Bishop 9, 9.2-9.3.1
- Alternative **free** pdf **books**:
- Hastie, Tibshirani and Friedman, The Elements of Statistical Learning, Springer and
- MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge

Your turn! Derive EM update for MoG

Your turn! Derive EM update for MoG

- We will try to maximize the log likelihood by setting the gradient to zero wrt to the parameters $\{\pi_k, \boldsymbol{\mu}_k, \Sigma_k\}$

$$E(\mathbf{w}) = - \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \Sigma_k) .$$

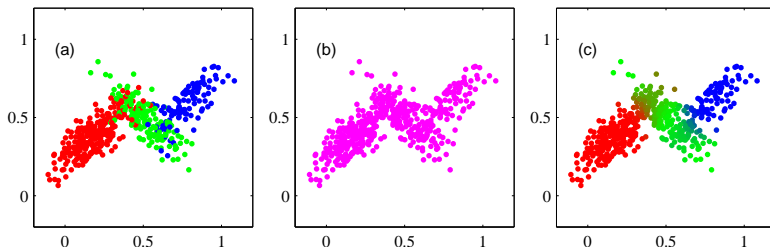
- Introduce the so-called **responsibility**

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_{k'}, \Sigma_{k'})} \in [0, 1] .$$

Your turn! Derive EM update for MoG

Responsibility – soft assignments

$$\begin{aligned}\gamma_{nk} &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})} \\ &= \frac{p(k)p(\mathbf{x}_n | k)}{\sum_{k'} p(k')p(\mathbf{x}_n | k')} = p(k | \mathbf{x}_n) \in [0, 1] .\end{aligned}$$





Your turn! Derive EM update for MoG

MoG maximum likelihood - π_k

Derivative wrt π_k of

$$\mathcal{L}(\mathbf{w}, \lambda) = E(\mathbf{w}) + \lambda \left[\sum_{k'=1}^K \pi_{k'} - 1 \right] .$$

Cost function and responsibility

$$E(\mathbf{w}) = - \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})} .$$



Your turn! Derive EM update for MoG

MoG maximum likelihood - μ_k

Use (see appendix C)

$$\frac{\partial}{\partial \mu_k} \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k) = \Sigma_k^{-1} (\mathbf{x}_n - \mu_k) \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)$$

Cost function and responsibility

$$E(\mathbf{w}) = - \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)$$

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n | \mu_{k'}, \Sigma_{k'})}.$$

Your turn! Derive EM update for MoG

MoG maximum likelihood - Σ_k

Use (see appendix C)

$$\frac{\partial}{\partial \Sigma_k} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k) = -\frac{1}{2} \left[\Sigma_k^{-1} - \Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} \right] \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k)$$

Cost function and responsibility

$$E(\mathbf{w}) = - \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k)$$

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_{k'}, \Sigma_{k'})}$$



E-step - for $n = 1, \dots, N$ and $k = 1, \dots, K$:

$$\gamma_{nk} \leftarrow \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}.$$

M-step - for $k = 1, \dots, K$:

$$N_k \leftarrow \sum_{n=1}^N \gamma_{nk}$$

$$\pi_k \leftarrow \frac{N_k}{N}$$

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

$$\boldsymbol{\Sigma}_k \leftarrow \frac{1}{N_k} \sum_{n=1}^N \gamma_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$