COURSE 02457

Signal Processing in Non-linear Systems:

Lecture 8

- Neural networks, weight decay, pruning
- Probabilities and densities, Bayes' theorem
- The normal distribution
- Gaussian mixtures
- Maximum likelihood learning
- Expectation maximization algorithm
- Exercise 8

The probability density function $p(x)$

- In one dimension, the probability density function $p(x)$ is characterized by

$$P(x \in [a,b]) = \int_a^b p(x)dx$$

and expectations are computed by

$$\mathcal{E}(f(x)) = \int_{\text{Domain of x}} f(x)p(x)dx$$

the density function is normalized

$$P(x \in \text{Domain of x}) = \int_{\text{Domain of x}} p(x)dx = 1$$

The 'average value of x' (the mean of x)

$$\mathcal{E}(x) \equiv \mu = \int_{\text{Domain of x}} xp(x)dx$$

The spread of x around it's mean (the standard deviation)

$$\sigma = \sqrt{\int_{\text{Domain of x}} (x-\mu)^2 p(x)dx}$$

The probability density function $p(x_1, x_2)$

- In two dimensions we need to worry about the density functions of the variables x_1, x_2. If the $joint\ density$ $p(x_1, x_2) = p(x_1)p(x_2)$, the two variables are independent. This means that expectations factorize, e.g.

$$\mathcal{E}(x_1 x_2) = \mathcal{E}(x_1)\mathcal{E}(x_2) \equiv \mu_1 \mu_2$$

- If the two variables are not independent we can, e.g., investigate the covariance between them,

$$\mathcal{E}((x_1 - \mu_1)(x_2 - \mu_2)) = \mathcal{E}(x_1 x_2) - \mu_1 \mu_2.$$

- If we divide the covariance by the standard deviations we get the correlation coefficient

$$\rho = \frac{\mathcal{E}((x_1 - \mu_1)(x_2 - \mu_2))}{\sigma_1 \sigma_2}$$

  The correlation coefficient is limited as $-1 < \rho < +1$

The probability density function $p(\mathbf{x})$

- In the multivariate case the probability density function $p(\mathbf{x})$ is characterized by

$$P(x_j \in [a_j, b_j] | j = 1, .., d) = \int_{a_1}^{b_1} ... \int_{a_d}^{b_d} p(\mathbf{x}) d\mathbf{x}$$

and expectations are computed by

$$\mathcal{E}(f(\mathbf{x})) = \int_{\text{Domain of } \mathbf{x}} f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

the density function is normalized $\mathcal{E}(1) = 1$.

The 'average value of $\mathbf{x}$' (the mean of $\mathbf{x}$)

$$\mathcal{E}(\mathbf{x}) \equiv \boldsymbol{\mu} = \int_{\text{Domain of } \mathbf{x}} \mathbf{x} p(\mathbf{x}) d\mathbf{x}$$

The spread of $\mathbf{x}$ around it's mean (the standard deviation) needs to be characterized by a matrix!

$$\boldsymbol{\Sigma} = \int_{\text{Domain of } \mathbf{x}} (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top p(\mathbf{x}) d\mathbf{x}$$

Bayes' theorem – multivariate version

$$
\begin{aligned}
P(\mathcal{C}_k, \mathbf{x}) &= p(\mathbf{x}|\mathcal{C}_k)P(\mathcal{C}_k) \\
P(\mathcal{C}_k, \mathbf{x}) &= P(\mathcal{C}_k|\mathbf{x})p(\mathbf{x})
\end{aligned}
$$

$$
\begin{aligned}
P(\mathcal{C}_k|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_k)P(\mathcal{C}_k)}{p(\mathbf{x})} \\
p(\mathbf{x}|\mathcal{C}_k) &= \frac{P(\mathcal{C}_k|X^l)p(\mathbf{x})}{P(\mathcal{C}_k)}
\end{aligned}
$$

$$
\sum_{k=1}^{c} P(\mathcal{C}_k|\mathbf{x}) = 1
$$

$$
\sum_{k=1}^{c} p(\mathbf{x}|\mathcal{C}_k)P(\mathcal{C}_k) = p(\mathbf{x})
$$

The uni-variate normal distribution

- In one dimension, the normal distribution's probability density function is given by

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

where the mean value parameter is

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

and the variance,

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

Multivariate normal distribution

- In d dimensions, the multivariate normal probability density function is given by

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

  where $\boldsymbol{\mu}$ is a d-dimensional vector, and $\boldsymbol{\Sigma}$ is a $d \times d$ covariance matrix.

- The covariance matrix has a set of eigenvectors

$$\boldsymbol{\Sigma}\mathbf{u}_j = \lambda_j \mathbf{u}_j, \quad j = 1, ..., d$$

  or in matrix notation

$$\boldsymbol{\Sigma}\mathbf{U} = \boldsymbol{\Lambda}\mathbf{U}$$

  If we define a vector $\mathbf{z} = \mathbf{U}^\top \mathbf{x}$, then

$$\mathcal{E}(\mathbf{z}\mathbf{z}^\top) = \mathbf{U}^\top \mathbf{x}\mathbf{x}^\top \mathbf{U} = \mathbf{U}^\top \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}\mathbf{U}^\top = \boldsymbol{\Lambda}$$

  The covariances are zero!, hence, the $\mathbf{z}$ vector has uncorrelated components.

Density estimation

- We want to model the density of a stochastic signal source

$$p(\mathbf{x}) \sim p(\mathbf{x}|\mathbf{w})$$

- where the family $p(\mathbf{x}|\mathbf{w})$ is a given parametric density.
- A density model can e.g. be used for outlier detection: How likely is a data point?

Maximum likelihood learning

- The training set is $D = (\chi)$, with $\chi = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_N\}$
- the likelihood function is given by

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

- The costfunction is then

$$E(\mathbf{w}) = -\log\left(\prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{w})\right)$$

$$E(\mathbf{w}) = \sum_{n=1}^{N} -\log p(\mathbf{x}_n|\mathbf{w})$$

Gaussian mixtures

- The gaussian mixture model is defined

$$p(\mathbf{x}|\mathbf{w}) = \sum_{j=1}^{M} P(j)p(\mathbf{x}|j, \mathbf{w}_j)$$

- Where each component density is a normal distribution with parameters: $\mathbf{w}_j = \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}$
- Think of the stochastic process as a $two\text{-}step$ process: first draw a component number j with relative probabilities $P(j)$, then draw a random vector from the given component. This is the way to simulate data from the this source.

Maximum likelihood learning for GM

- The costfunction is

$$\begin{aligned} E(\mathbf{w}) &= \sum_{n=1}^{N} -\log p(\mathbf{x}_n|\mathbf{w}) \\ &= \sum_{n=1}^{N} -\log \sum_{j=1}^{M} p(\mathbf{x}_n|j)P(j) \end{aligned}$$

- We will simplify the family to isotropic Gaussians, al expressions generalize easily to full covariance matrices,

$$p(\mathbf{x}|\boldsymbol{\mu}_j, \sigma_j^2) = \frac{1}{(2\pi\sigma_j^2)^{d/2}} \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_j)^2}{2\sigma_j^2}\right)$$

- The derivative w.r.t. the mean value vector is

$$\begin{aligned} \frac{\partial E}{\partial \boldsymbol{\mu}_j} &= -\sum_{n=1}^{N} \frac{\partial/\partial \boldsymbol{\mu}_j \sum_{j'=1}^{M} p(\mathbf{x}_n|j')P(j')}{p(\mathbf{x}_n|\mathbf{w})} \\ &= \sum_{n=1}^{N} P(j|\mathbf{x}_n)\frac{(\mathbf{x}_n - \boldsymbol{\mu}_j)}{\sigma_j^2} \end{aligned}$$

Maximum likelihood learning for GM

- The derivative w.r.t. the mean value vector is

$$\frac{\partial E}{\partial \boldsymbol{\mu}_j} = \sum_{n=1}^{N} P(j|\mathbf{x}_n) \frac{(\mathbf{x}_n - \boldsymbol{\mu}_j)}{\sigma_j^2}$$

- the derivative w.r.t. the widths is given by

$$\frac{\partial E}{\partial \sigma_j} = \sum_{n=1}^{N} P(j|\mathbf{x}_n) \left[\frac{d}{\sigma_j} - \frac{(\boldsymbol{\mu}_j - \mathbf{x}_n)^2}{\sigma_j^3} \right]$$

- We can understand these rules, let us try to solve them by equating the derivative to zero

$$\widehat{\boldsymbol{\mu}_j} = \frac{\sum_{n=1}^{N} P(j|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} P(j|\mathbf{x}_n)}$$

and

$$\widehat{\sigma_j^2} = \frac{1}{d} \frac{\sum_{n=1}^{N} P(j|\mathbf{x}_n)(\mathbf{x}_n - \boldsymbol{\mu}_j)^2}{\sum_{n=1}^{N} P(j|\mathbf{x}_n)}$$

Maximum likelihood learning for GM

- Next we want to estimate $P(j)$. Note that the prior probabilities sum to unity:

$$\sum_{j=1}^{M} P(j) \;=\; 1$$

- Use the softmax trick

$$P(j) \;=\; \frac{\exp(\gamma_j)}{\sum_{j'=1}^{M} \exp(\gamma_{j'})}$$

- The derivative of the cost function is

$$\frac{\partial E}{\partial \gamma_j} \;=\; \sum_{k=1}^{M} \frac{\partial E}{\partial P(k)} \frac{\partial P(k)}{\partial \gamma_j}$$

Maximum likelihood learning for GM

- The derivative of the cost function is

$$\frac{\partial E}{\partial \gamma_j} = \sum_{k=1}^{M} \frac{\partial E}{\partial P(k)} \frac{\partial P(k)}{\partial \gamma_j}$$

$$\frac{\partial E}{\partial P(k)} = -\sum_{n=1}^{N} \frac{1}{p(\mathbf{x}_n)} p(\mathbf{x}_n|k) = -\sum_{n=1}^{N} \frac{P(k|\mathbf{x}_n)}{P(k)}$$

$$\frac{\partial P(k)}{\partial \gamma_j} = \delta_{k,j} P(k) - P(k)P(j)$$

- hence,

$$\frac{\partial E}{\partial \gamma_j} = -\sum_{n=1}^{N} [P(j|\mathbf{x}_n) - P(j)] = 0$$

- the solution is

$$\widehat{P(j)} = \frac{1}{N} \sum_{n=1}^{N} P(j|\mathbf{x}_n)$$

The EM algorithm

- The Expectation-maximization algorithm is a general scheme for maximum likelihood estimation. Note that the change in costfunction that occurs when we iterate the estimates

$$\begin{aligned} E^{\text{new}} - E^{\text{old}} &= -\sum_{n=1}^{N} \log \frac{p^{\text{new}}(\mathbf{x}_n)}{p^{\text{old}}(\mathbf{x}_n)} \\ &= -\sum_{n=1}^{N} \log \frac{\sum_{j=1}^{M} p^{\text{new}}(\mathbf{x}_n|j) P^{\text{new}}(j)}{p^{\text{old}}(\mathbf{x}_n)} \frac{P^{\text{old}}(j|\mathbf{x}_n)}{P^{\text{old}}(j|\mathbf{x}_n)} \\ &\leq -\sum_{n=1}^{N} \sum_{j} P^{\text{old}}(j|\mathbf{x}_n) \log \frac{p^{\text{new}}(\mathbf{x}_n|j) P^{\text{new}}(j)}{p^{\text{old}}(\mathbf{x}_n) P^{\text{old}}(j|\mathbf{x}_n)} \end{aligned}$$

- The inequality is based on Jensen inequality:

$$\log \left(\sum_j \lambda_j x_j \right) \geq \sum_j \lambda_j \log(x_j)$$

- This is an upper bound so that it can be minimized and this give us similar results as for the maximum likelihood, now in iterative form.

The EM algorithm cont'd

- This is an upper bound so that it can be minimized. This gives us similar results as for the maximum likelihood, now in iterative form: The M-step

$$\widehat{\boldsymbol{\mu}_j^{\text{new}}} = \frac{\sum_{n=1}^{N} P^{\text{old}}(j|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} P^{\text{old}}(j|\mathbf{x}_n)}$$

and

$$\widehat{(\sigma_j^{\text{new}})^2} = \frac{1}{d}\frac{\sum_{n=1}^{N} P^{\text{old}}(j|\mathbf{x}_n)(\mathbf{x}_n - \boldsymbol{\mu}_j^{\text{new}})^2}{\sum_{n=1}^{N} P^{\text{old}}(j|\mathbf{x}_n)}$$

$$P_j^{\text{new}} = \frac{1}{N}\sum_{n=1}^{N} P^{\text{old}}(j|\mathbf{x}_n)$$

- The E-step is to re-calculate using Bayes theorem

$$P^{\text{old}}(j|\mathbf{x}_n) = \frac{P_j^{\text{new}} p^{\text{new}}(\mathbf{x}_n|j)}{p^{\text{new}}(\mathbf{x}_n)} = \frac{P_j^{\text{new}} p^{\text{new}}(\mathbf{x}_n|j)}{\sum_j P_j^{\text{new}} p^{\text{new}}(\mathbf{x}_n|j)}$$

The K-means algorithm

- The K-means algorithm is a simple $clustering$ algorithm aimed at minimizing the cost function for K clusters,

$$E = \sum_{j=1}^{K} \sum_{n \in S_j} (\mathbf{x}_n - \boldsymbol{\mu}_j)^2$$

- ...where $\boldsymbol{\mu}_j$ is the mean of the data points associated most with the $j'th$ component S_j (i.e. closest to)

$$\boldsymbol{\mu}_j = \frac{\sum_{n \in S_j} \mathbf{x}_n}{\sum_{n \in S_j} 1}$$

- Initialization is rather important, e.g., a cluster component which is never assigned any points will not be updated

The K-means algorithm cont'

- If we calculate the variance associated with the $j'th$ cluster

$$\sigma_j^2 = \frac{1}{d}\frac{\sum_{n \in S_j}(\mathbf{x}_n - \boldsymbol{\mu}_j)^2}{\sum_{n \in S_j} 1}$$

- and let the assignments be

  $P(j) = 1/K$ ...or even

  $P(j) = \sum_{n \in S_j} 1/N$

  we can actually use the parameters to define a density estimate as for the Gaussian mixture.