MLMI MPhil

Module MLMI1: Introduction to Machine Learning

Solutions to Example Sheet 2: Clustering, the Expectation Maximisation Algorithm, Markov Models, and Hidden Markov Models

Clustering

1. K-means clustering*

   Consider the K-means algorithm that seeks to minimise the cost function

$$C = \sum_{n=1}^{N} \sum_{k=1}^{K} s_{nk} ||x_n - m_k||^2$$

   where m_k is the mean (centre) of cluster k, x_n is the data point n, $s_{nk} = 1$ indicates that the nth data point has been assigned to cluster k, and $s_{nk} = 0$ indicates that it has not been assigned to that cluster. There are N data points and K clusters.

   (a) Given all the assignments $\{s_{nk}\}$ determine the value of m_k that minimises the cost C and give an interpretation in terms of the K-means algorithm.

   (b) You would like to automatically learn the number of clusters K from data. One possibility is to minimise the cost C as a function of K. Explain whether this is a good idea or not, and what the solution to this minimisation is.

   (c) In many real-world applications, data points arrive sequentially and one wants to cluster them as they come in. Devise a sequential variant of the k-means algorithm which takes in one data point at a time and updates the means $\{m_1, ..., m_K\}$ sequentially without revisiting previous data points. Describe your sequential algorithm.

5 $C(\{s\},\{m\}) = \sum_{n=1}^{N}\sum_{k=1}^{K} s_{nk}\,\|x_n - m_k\|^2 = \sum_n\sum_k\sum_d s_{nk}\,(x_{nd} - m_{kd})^2$

a) $\frac{dC(\{s\},\{m\})}{dm_{le}} = -2\sum_{n=1}^{N} s_{nl}\,(x_{ne} - m_{le}) = 0$

$\Rightarrow \sum_{n=1}^{N} s_{nl}\,x_{ne} = m_{le}\sum_{n=1}^{N} s_{nl}$

$\Rightarrow m_{le} = \dfrac{\sum_{n=1}^{N} s_{nl}\,x_{ne}}{\sum_{n=1}^{N} s_{nl}}$

Or in vector form $\underline{m}_k = \dfrac{\sum_{n=1}^{N} s_{nk}\,\underline{x}_n}{\sum_{n=1}^{N} s_{nk}}$

Sum of all of the data values assigned to k^{th} cluster

number of data points assigned to k^{th} cluster

b) Bad idea: if $K = N$ & $\underline{m}_n = \underline{x}_n$ cost is zero which is lowest possible

i.e. each datapoint is a "cluster"

c) Several possible options for this one but a logical soln would use:

since $$\underline{M}_k = \frac{\sum_{n=1}^{N} \underline{x}_n s_{nk}}{\sum_{n=1}^{N} s_{nk}} = \frac{\sum_{n=1}^{N-1} \underline{x}_n s_{nk} + \underline{x}_N s_{Nk}}{\sum_{n=1}^{N-1} s_{nk} + s_{Nk}}$$

where $\sum_{n=1}^{N-1} \underline{x}_n s_{nk} = \underline{\beta}_k^{(N-1)}$ and $\sum_{n=1}^{N-1} s_{nk} = \alpha_k^{(N-1)}$

New datum arrives

- assign to nearest cluster and set assignment vector $\{s_{Nk}\}_{k=1}^{K}$
- update statistics $\underline{\beta}_k^{(N)} = \underline{\beta}_k^{(N-1)} + \underline{x}_N s_{Nk}$ [running sum of data vectors associated w/ k^{th} cluster]

  $\alpha_k^{(N)} = \alpha_k^{(N-1)} + s_{Nk}$ [running count of # of data points associated w/ k^{th} cluster]
- recompute means $\underline{M}_k = \frac{\underline{\beta}_k^{(N)}}{\alpha_k^{(N)}}$
- repeat

2. The KL Divergence

The KL Divergence between two discrete distributions $p(x = k) = p_k$ and $p(x = k) = q_k$ is defined as $\mathcal{KL}(q, p) = \sum_{k=1}^{K} q_k \log \frac{q_k}{p_k}$.

(a) Prove that the KL Divergence is non-negative and that it attains its unique minimum when $q_k = p_k$.

(b) A machine learner has a target distribution $\mathbf{p} = [p_1, p_2, p_3, p_4, p_5, p_6] = [1, 1, 0, 0, 1, 1]/4$ which they want to fit with approximating distribution q. The choices for q available to the machine learner are: $q_1 = [1, 1, 0, 0, 0, 0]/2$, $q_2 = [0, 1, 1, 0, 0, 0]/2$, $q_3 = [0, 0, 1, 1, 0, 0]/2$, $q_4 = [0, 0, 0, 1, 1, 0]/2$, $q_5 = [0, 0, 0, 0, 1, 1]/2$, and $q_6 = [1, 1, 1, 1, 1, 1]/6$.

i. Determine the distribution(s) q_i that minimises $\mathcal{KL}(q_i, p)$. Comment on your result.

ii. Determine the distribution(s) q_i that minimises $\mathcal{KL}(p, q_i)$. Comment on your result.

Note that, by convention, $0 \times \log 0 = 0$ since $\lim_{\Delta \to 0} \Delta \log \Delta = 0$.

b

a) $KL(q \| p) = \sum_k q_k \log \frac{q_k}{p_k}$

constraint that probabilities sum to 1

$0 = \frac{\partial}{\partial q_l} \left[\sum_k q_k \log \frac{q_k}{p_k} + \lambda \left(\sum_k q_k - 1 \right) \right]$

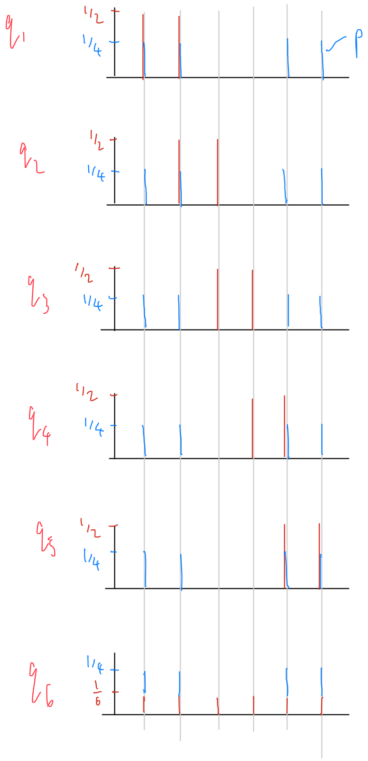
Lagrange multiplier

$0 = \log \frac{q_l}{p_l} + \frac{q_l}{q_l} + \lambda$

$\Rightarrow q_l = p_l$ @which $KL(q^{opt} \| p) = \sum_k p_k \log \frac{p_k}{p_k} = 0$

$\frac{\partial^2}{\partial q_l^2} KL(q \| p) = \frac{1}{q_l} > 0 \Rightarrow$ maximum

b)



	$KL(q \| p)$	$KL(p \| q)$
q_1	$\log 2$	∞
q_2	∞	∞
q_3	∞	∞
q_4	∞	∞
q_5	$\log 2$	∞
q_6	∞	$\log \frac{3}{2}$

$KL(q||p)$ is zero avoiding & tends to fit single modes of p with q

$KL(p||q)$ is not zero avoiding & tends to cover multiple modes of p with q

The EM Algorithm

3. Factor Analysis and EM*

   The noisy depth sensor from question 2 in example sheet 1 is used to collect measurements of the distances to a set of N objects that are unknown distances d_n metres away. The object depths can be assumed, *a priori*, to be distributed according to independent standard Gaussian distributions $p(d_n) = \mathcal{N}(d_n; 0, 1)$. As before, the depth sensor returns y_n a noisy measurement of the depth, that is also assumed to be Gaussian $p(y_n|d_n, \sigma_y^2) = \mathcal{N}(y_n; d_n, \sigma_y^2)$.

   The variance of the σ_y^2 sensor noise is unknown and must be estimated from the measured depths $\{y_n\}_{n=1}^N$ using maximum likelihood.

   (a) Derive the steps of the EM algorithm for performing this. Your answer should include the E-Step (you may find your solution to question 2 in example sheet 1 useful here) and the M-Step.

   (b) An alternative approach to maximum-likelihood learning would optimise the log-likelihood $p(\{y_n\}_{n=1}^N|\sigma_y^2)$ directly. Compute the objective function for this procedure. How do you think this approach will perform compared to EM?

8 $p(d_n) = \mathcal{N}(d_n; 0, 1)$

$p(y_n | d_n, \sigma_y^2) = \mathcal{N}(y_n; d_n, \sigma_y^2)$

in E-Step set $q_n(d_n) = p(d_n | y_n, \sigma_y^2) = \mathcal{N}(d_n; \mu_{d_n|y_n}, \sigma^2_{d_n|y_n})$

where $\mu_{d_n|y_n} = \frac{y_n}{1+\sigma_y^2}$ $\quad \sigma^2_{d_n|y_n} = \frac{\sigma_y^2}{1+\sigma_y^2}$ (See Q2 Examples Sheet 1)

in M-Step

$$\sigma_y^2 = \underset{\sigma_y^2}{\operatorname{argmax}} \underbrace{\mathbb{E}_q\left[\log p(\{d_n\}_{n=1}^N, \{y_n\}_{n=1}^N | \sigma_y^2) \right]}_{G_t(\sigma_y^2)}$$

↑ "average log-joint"

$$G_t(\sigma_y^2) = \mathbb{E}_q\left[\sum_n \log p(d_n) + \sum_n \log p(y_n | d_n, \sigma_y^2) \right]$$

$$= \underbrace{c}_{\text{constant wrt } \sigma_y^2} - \frac{1}{2\sigma_y^2} \sum_n \mathbb{E}_{q(d_n)}\left[(y_n - d_n)^2 \right] - \frac{N}{2} \log 2\pi\sigma_y^2 \quad ★$$

$$= c - \frac{1}{2\sigma_y^2} \sum_n \left(y_n^2 + \mathbb{E}_{q(d_n)}[d_n^2] - 2y_n \mathbb{E}_{q(d_n)}[d_n] \right) - \frac{N}{2} \log 2\pi\sigma_y^2$$

$$\frac{dG(\sigma_y^2)}{d\sigma_y^2} = \frac{1}{2\sigma_y^4} \sum_n \left(y_n^2 + \mathbb{E}_{q(d_n)}[d_n^2] - 2y_n \mathbb{E}_{q(d_n)}[d_n] \right) - \frac{N}{2\sigma_y^2} = 0$$

$$\therefore \sigma_y^2 = \frac{1}{N}\sum_n y_n^2 + \frac{1}{N}\sum_n \left(\mu_{d_n|y_n}^2 + \sigma_{d_n|y_n}^2\right) - \frac{2}{N}\sum_n y_n \mu_{d_n|y_n}$$

Algorithmically this is what you need, although for intuition ✡ yields:

$$\sigma_y^2 = \frac{1}{N}\sum_n \mathbb{E}_{q(d_n)}\left[(y_n - d_n)^2\right]$$

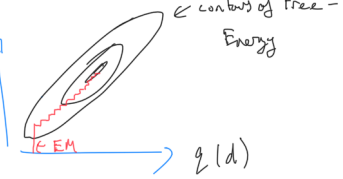
i.e. the noise variance is the average (over data points n & the posterior) of the squared error between y_n & d_n.

b) $$p(y_n|\sigma_y^2) = \mathcal{N}(y_n; 0, 1+\sigma_y^2)$$

$$\therefore \log p(\{y_n\}_{n=1}^N|\sigma_y^2) = -\frac{N}{2}\log[2\pi(1+\sigma_y^2)] - \frac{1}{2(1+\sigma_y^2)}\sum_n y_n^2$$

Optimising this function for σ_y^2 (eg by gradient ascent) would be MUCH faster than EM. EM performs co-ordinate ascent of the free-energy in q & θ which tends to be slow.

It turns out that if EM is analytic, so too is direct optimisation of the log likelihood which is almost always to be preferred.



4. Gaussian Mixture Models and EM** (beyond tripos standard; optional)

   A Gaussian Mixture Model for D-dimensional data $\{\boldsymbol{x}_n\}_{n=1}^N$ comprises a categorical distribution over the class membership variables $p(s_n = k|\theta) = \pi_k$ and a general multivariate Gaussian distribution over the observed data given the class membership variables, $p(\boldsymbol{x}_n|s_n = k, \theta) = \mathcal{N}(\boldsymbol{x}_n; \boldsymbol{m}_k, \Sigma_k)$, (i.e. Σ_k is not isotopic). The posterior distribution over the class membership variables has been computed $p(s_n = k|\boldsymbol{x}_n, \theta) = q_{n,k}$.

   (a) Derive the M-Step update of the EM algorithm, i.e. the formulae for updating the parameters $\{\pi_k, \boldsymbol{m}_k, \Sigma_k\}_{k=1}^K$ using the posterior probabilities, $q_{n,k}$.

   (b) A friend suggests that it might be possible to speed up the EM algorithm by updating the posterior distribution of only a subset of $K < N$ data points during the E-Step that are selected uniformly at random each time, and then updating the parameters using the same expressions derived in part (a).

      i. Will this procedure converge to a (local) optimum of the likelihood?

      ii. Will this partial E-Step update procedure be computationally more efficient?

   You may find the following identities useful,

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} \log\det(\Sigma) = \mathrm{trace}\left(\Sigma^{-1}\frac{\mathrm{d}\Sigma}{\mathrm{d}\alpha}\right), \quad \frac{\mathrm{d}}{\mathrm{d}\alpha}\Sigma^{-1} = -\Sigma^{-1}\frac{\mathrm{d}\Sigma}{\mathrm{d}\alpha}\Sigma^{-1}.$$

9. $$\mathcal{F}(q(\underline{s}), \theta) = \sum_n q(s_n) \log \frac{p(\underline{x}_n | s_n, \theta) p(s_n|\theta)}{q(s_n)}$$

$$q(s_n = k) = q_{nk}$$

$$\therefore \quad \mathcal{F}(\underline{q}(\underline{s}), \theta) = \sum_n \sum_k q_{nk} \log \frac{p(\underline{x}_n | s_n = k, \theta) p(s_n = k|\theta)}{q(s_n = k)}$$

where

$$p(\underline{x}_n | s_n = k, \theta) = N(\underline{x}_n; \underline{\mu}_k, \underline{\underline{\Sigma}}_k) \qquad p(s_n = k|\theta) = \pi_k$$

$$\therefore \quad \mathcal{F}(\underline{q}(\underline{s}), \theta) = \sum_n \sum_k q_{nk} \left[-\frac{1}{2} \log \det(2\pi \underline{\underline{\Sigma}}_k) - \frac{1}{2} (\underline{x}_n - \underline{\mu}_k)^T \underline{\underline{\Sigma}}_k^{-1} (\underline{x}_n - \underline{\mu}_k) + \log \pi_k \right] + c$$

(c: does not depend on parameters)

$$\frac{\partial \mathcal{F}}{\partial \mu_{k'd'}} = \frac{\partial}{\partial \mu_{k'd'}} \left[-\frac{1}{2} \sum_n \sum_k q_{nk} \sum_{ij} (x_{in} - \mu_{ki}) \Sigma^{-1}_{kij} (x_{jn} - \mu_{kj}) \right] = 0$$

$$= \frac{1}{2} \sum_n q_{nk'} \left[\sum_j \Sigma^{-1}_{k'd'j} (x_{jn} - \mu_{k'j}) + \sum_i (x_{in} - \mu_{ki}) \Sigma^{-1}_{kid'} \right]$$

$$= \sum_n q_{nk'} \sum_j \Sigma^{-1}_{k'd'j} (x_{jn} - \mu_{k'j}) = \left[\Sigma^{-1}_{k'} \sum_n q_{nk'} (\underline{x}_n - \underline{\mu}_{k'}) \right]_{d'}$$

$$\therefore \quad \underline{\mu}_k = \frac{\sum_n q_{kn}\,\underline{x}_n}{\sum_n q_{kn}}$$

alternatively, you could differentiate wrt $\Sigma^{-1}_{k'i'j'}$ & use $\det(\Sigma) = \frac{1}{\det(\Sigma^{-1})}$

$$\frac{dq}{d\Sigma_{k'i'j'}} = \frac{d}{d\Sigma_{k'i'j'}}\left[\sum_{n,k} q_{nk}\left[-\tfrac{1}{2}\log\det\left(2\pi\underline{\underline{\Sigma}}_k\right) - \tfrac{1}{2}(\underline{x}_n-\underline{\mu}_k)^T\underline{\underline{\Sigma}}^{-1}(\underline{x}_n-\underline{\mu}_k)\right]\right]$$

$$\frac{d}{d\alpha}\log\left(\det\underline{\underline{\Sigma}}_k\right) = \mathrm{trace}\left(\underline{\underline{\Sigma}}_k^{-1}\frac{d\underline{\underline{\Sigma}}_k}{d\alpha}\right) \qquad \frac{d}{d\alpha}\underline{\underline{\Sigma}}_k^{-1} = -\underline{\underline{\Sigma}}^{-1}\frac{d\underline{\underline{\Sigma}}}{d\alpha}\underline{\underline{\Sigma}}^{-1}$$

$$\frac{dq}{d\Sigma_{k'i'j'}} = \sum_{nk} q_{nk}\left[-\tfrac{1}{2}\sum_{ij}\Sigma^{-1}_{kij}\frac{d\Sigma_{kji}}{d\Sigma_{k'i'j'}} + \tfrac{1}{2}\sum_{ijlm}(x_{ni}-\mu_{ki})\Sigma^{-1}_{kil}\frac{d\Sigma_{klm}}{d\Sigma_{k'i'j'}}\Sigma^{-1}_{kmj}(x_{nj}-\mu_{kj})\right]$$

$$= \sum_{nk} q_{nk}\left[-\frac{1}{2}\sum_{ij}\Sigma^{-1}_{kij}\,\delta_{kk'}\delta_{i'j}\delta_{j'i} + \frac{1}{2}\sum_{ijlm}(x_{ni}-\mu_{ki})\Sigma^{-1}_{kil}\,\delta_{k'k}\delta_{i'l}\delta_{mj'}\Sigma^{-1}_{kmj}(x_{nj}-\mu_{kj})\right]$$

$$= \frac{1}{2}\sum_n q_{nk'}\left(-\Sigma^{-1}_{k'j'i'} + \sum_{ij}(x_{ni}-\mu_{k'i})\Sigma^{-1}_{k'i'i}\Sigma^{-1}_{k'j'j}(x_{nj}-\mu_{k'j})\right)$$

$\Rightarrow \sum_n q_{nk'}\left(-\underline{\underline{\Sigma}}^{-1}_{k'} + \underline{\underline{\Sigma}}^{-1}_{k'}(\underline{x}_n-\underline{\mu}_{k'})(\underline{x}_n-\underline{\mu}_{k'})^T\underline{\underline{\Sigma}}^{-1}_{k'}\right) = 0$

$$0 = \sum_n q_{nk'}\left(-\underline{\underline{\Sigma}}^{-1}_{k'} + \underline{\underline{\Sigma}}^{-1}_{k'}(\underline{x}_n-\underline{\mu}_{k'})(\underline{x}_n-\underline{\mu}_{k'})^T\underline{\underline{\Sigma}}^{-1}_{k'}\right)$$

$$\Rightarrow \underline{\underline{\Sigma}}_{k'} = \sum_n q_{nk'}(\underline{x}_n-\mu_{k'})(\underline{x}_n-\mu_{k'})^T \Big/ \sum_n q_{nk'}$$

pre & post multiply by $\underline{\underline{\Sigma}}_{k'}$

To find π we need to use Lagrange multipliers:

$$\frac{d}{d\pi_{k'}}\left(\mathcal{F}(q,\theta) - \lambda\left(\sum_k \pi_k - 1\right)\right) = \frac{d}{d\pi_{k'}}\left[\sum_{nk}\left(q_{nk}\log\pi_k - \lambda\pi_k\right) - 1\right]$$

$$= \sum_n q_{nk'}\left(\frac{1}{\pi_{k'}} - \lambda\right) = 0$$

$$\Rightarrow \pi_{k'} = \frac{\sum_n q_{nk'}}{\sum_{nk} q_{nk'}} \leftarrow \text{from constraint that } \sum_k \pi_k = 1$$

$$= \frac{1}{N}\sum_n q_{nk'}$$

b i) The free-energy is guaranteed to improve (or stay the same) if only a fraction of the datapoint posteriors are updated.
∴ the procedure will converge to a local optimum of the likelihood as for normal EM

ii) It's not completely clear that the procedure will help in all cases - for although the cost of the E-Step is reduced (from $O(N)$ to $O(K)$) convergence will take more steps in general. However, when there are lots of data, you don't need to visit all of them to figure out a good update for the parameters so this means the procedure is likely to be substantially more efficient in these cases.

5. Markov Models: fitting bi-gram models

   A data scientist observes part of a long sequence that contains $K = 3$ characters: $ABAAABBABCCCBC$. She would like to use a bi-gram model to fit the data with parameters $p(y_1 = k|\theta) = \pi^0_k$ and $p(y_t = k|y_{t-1} = l, \theta) = T_{k,l}$.

   (a) Write down the log-likelihood for the model and optimise it with respect to π^0 and T to find the maximum likelihood parameter estimates.

   (b) Is the maximum-likelihood estimate sensible? How might you improve the estimate?

a) $$p(y_{1:T}|\underline{\pi}^0, \underline{\underline{T}}) = \left(\prod_k \pi_k^{0\,\mathbb{1}(y_1=k)}\right)\left(\prod_{t=2}^{T}\prod_{k=1}^{K}\prod_{l=1}^{K} T_{kl}^{\mathbb{1}(y_t=k,\,y_{t-1}=l)}\right)$$

where

$$\mathbb{1}(x) = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{if } x \text{ is false} \end{cases}$$

called the indicator function

$$\therefore \log p(y_{1:T}|\underline{\pi}^0, \underline{\underline{T}}) = \sum_k \mathbb{1}(y_1=k)\log \pi_k^0 + \sum_{t=2}^{T}\sum_{k=1}^{K}\sum_{l=1}^{K}\mathbb{1}(y_t=k, y_{t-1}=l)\log T_{kl}$$

$$= \sum_k \mathbb{1}(y_1=k)\log\pi_k^0 + \sum_{kl} N_{kl}\log T_{kl} = \mathcal{L}(\theta)$$

(N_{kl}: number of transitions from state l to state k)

Maximum likelihood:

$$\frac{d}{d\pi_a^0}\left[\mathcal{L}(\theta) + \lambda\left(\sum_k \pi_k^0 - 1\right)\right] = \frac{\mathbb{1}(y_1=a)}{\pi_a^0} + \lambda = 0$$

$$\Rightarrow \pi_a^0 = \mathbb{1}(y_1=a)$$

$$\frac{d}{dT_{ab}}\left[\mathcal{L}(\theta) + \sum_{l=1}^{K}\lambda_l\left(\sum_{k=1}^{K} T_{kl} - 1\right)\right] = \frac{N_{ab}}{T_{ab}} + \lambda_b \Rightarrow T_{ab} = \frac{N_{ab}}{\sum_a N_{ab}}$$

(MLE = fraction of transitions from b that end up at a)

In the specific case given:

$$\underline{\pi}^0 = [1, 0, 0] \qquad \underline{\underline{T}} = \begin{bmatrix} 2/5 & 2/5 & 0 \\ 3/5 & 1/5 & 1/3 \\ 0 & 2/5 & 2/3 \end{bmatrix}$$

$$\underline{\underline{N}} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{bmatrix} 2 & 2 & 0 \\ 3 & 1 & 1 \\ 0 & 2 & 2 \end{bmatrix} \end{array} \quad \begin{array}{l} \leftarrow \text{from} \\ \\ \\ \\ \text{to} \end{array}$$

$$\begin{array}{ccc} 5 & 5 & 4 \\ \uparrow & \nearrow & \nearrow \\ \sum_a N_{ab} & & \end{array}$$

$$\mathcal{L}(\theta) = \log \pi_1^{(0)} + 2\log T_{11} + 2\log T_{12} + 3\log T_{21} + \log T_{22} + \log T_{23} + 2\log T_{32} + 2\log T_{33}$$

$$= \log \pi_1^{(0)} + 2\left[\log T_{11} + \log T_{12} + \log T_{32} + \log T_{33}\right] + 3\log T_{21} + \log T_{22} + \log T_{23}$$

b) The MLE is prone to over fitting (eg the estimate for $\underline{\pi}^0$ is deterministic even though we have only seen one data point from $\underline{\pi}^0$)

To avoid such pathologies in estimating $\underline{\pi}^0$ & $\underline{\underline{T}}$ we can use priors to get a MAP or Bayesian estimate.

6. Markov Models: Gaussian AR(1) models

A data scientist observes a sequence of scalar variables $y_{1:T} = \{y_t\}_{t=1}^T$ generated from a Gaussian AR(1) process $y_t = \lambda y_{t-1} + \epsilon_t$ where $\epsilon_t \sim \mathcal{N}(\mu, \sigma^2)$.

She knows that the invariant distribution of the process has the following properties

$$\lim_{t\to\infty} \mathbb{E}(y_t) = \mu_\infty, \quad \lim_{t\to\infty} \mathbb{E}(y_t^2) = \sigma_\infty^2 + \mu_\infty^2, \quad \lim_{t\to\infty} \mathbb{E}(y_t y_{t-1}) = \alpha_\infty.$$

(a) Derive the parameters of the Gaussian AR(1) process $\{\lambda, \mu, \sigma^2\}$ in terms of the properties of the invariant distribution $\{\mu_\infty, \sigma_\infty^2, \alpha_\infty\}$.

(b) The data scientist reinterprets the original Markov model in terms of a new random variable z_t such that $y_t = z_t + \mu_\infty$. State the form of the distribution $p(z_{1:T})$ required for this model to be equivalent to the original one.

Q2 a) First the long way!

$$y_t = \lambda y_{t-1} + \varepsilon_t \qquad \varepsilon_t \sim N(\mu, \sigma^2)$$

① Taking expectations of both sides wrt to all sources of randomness

$$\langle y_t \rangle = \langle \lambda y_{t-1} \rangle + \langle \varepsilon_t \rangle = \lambda \langle y_{t-1} \rangle + \mu$$

If invariant distribution exists then

$$\mu_\infty = \lambda \mu_\infty + \mu \qquad \Rightarrow \mu_\infty = \frac{\mu}{1-\lambda}$$

② Similarly

$$\langle y_t^2 \rangle = \langle (\lambda y_{t-1} + \varepsilon_t)^2 \rangle = \lambda^2 \langle y_{t-1}^2 \rangle + \langle \varepsilon_t^2 \rangle + 2\lambda \langle \varepsilon_t y_{t-1} \rangle$$

$$= \lambda^2 \langle y_{t-1}^2 \rangle + \mu^2 + \sigma^2 + 2\mu\lambda \langle y_{t-1} \rangle$$

if invariant distribution exists then

$$\sigma_\infty^2 + \mu_\infty^2 = \lambda^2 \left[\sigma_\infty^2 + \mu_\infty^2 \right] + \mu^2 + \sigma^2 + 2\mu\lambda\mu_\infty$$

$$\sigma_\infty^2 + \mu_\infty^2 = \frac{\mu^2 + \sigma^2 + 2\mu\lambda\mu_\infty}{1-\lambda^2}$$

③ $\langle y_t y_{t-1} \rangle = \langle (\lambda y_{t-1} + \varepsilon_t) y_{t-1} \rangle = \lambda \langle y_{t-1}^2 \rangle + \mu\mu_\infty$

If invariant distribution exists then;

$$\alpha_\infty = \lambda\left[\sigma_\infty^2 + \mu_\infty^2\right] + \mu\mu_\infty$$

rearrange: $\lambda = \dfrac{\alpha_\infty - \mu_\infty \mu}{\sigma_\infty^2 + \mu_\infty^2}$ & $\mu = \mu_\infty(1-\lambda)$

eliminate λ

So, a procedure for finding λ, μ & σ^2 is

① $\mu = \left(\sigma_\infty^2 + \mu_\infty^2 - \alpha_\infty\right)\dfrac{\mu_\infty}{\sigma_\infty^2}$

② $\lambda = \dfrac{\alpha_\infty - \mu_\infty^2}{\sigma_\infty^2}$

Sub ① into $\lambda = \dfrac{\alpha_\infty - \mu_\infty \mu}{\sigma_\infty^2 + \mu_\infty^2}$

③ $\sigma^2 = \left(\sigma_\infty^2 + \mu_\infty^2\right)(1-\lambda^2) - \mu^2 - 2\mu\lambda\mu_\infty$

$= \sigma_\infty^2(1-\lambda^2) + \mu_\infty^2\left[(1-\lambda^2) - (1-\lambda)^2 - 2\lambda(1-\lambda)\right]$

$= \sigma_\infty^2(1-\lambda^2) + \mu_\infty^2\left[1 - \lambda^2 - 1 + 2\lambda - \lambda^2 - 2\lambda + 2\lambda^2\right]$

$= \sigma_\infty^2(1-\lambda^2)$

b) $y_t = z_t + \mu_\infty = \lambda y_{t-1} + \varepsilon_t \qquad \varepsilon_t \sim N(\mu, \sigma^2)$

$= \lambda(z_{t-1} + \mu_\infty) + \varepsilon_t$

$\Rightarrow \quad z_t = \lambda(z_{t-1} + \mu_\infty) - \mu_\infty + \varepsilon_t$

$= \lambda z_{t-1} + \underbrace{\lambda\mu_\infty - \mu_\infty}_{-\mu \text{ (see expression for } \mu_\infty)} + \mu + n_t \qquad n_t \sim N(0, \sigma^2)$

$z_t = \lambda z_t + n_t$

$\Rightarrow p(z_{1:T}) = \prod_t N(z_t; \lambda z_{t-1}, \sigma^2)$ [standard AR(1) model]

This leads to a much faster way of doing part (a) & a useful way of checking this result:

$\sigma_\infty^2 = \langle z_t^2 \rangle$ [standard AR(1) process]

For standard AR(1):

$\sigma_\infty^2 = \dfrac{\sigma^2}{1-\lambda^2}$ (see lectures)

$\therefore \quad \sigma^2 = \sigma_\infty^2(1-\lambda^2)$ much faster!

Hidden Markov Models

7. Discrete Valued Hidden Markov Models*

   (a) Provide the probabilistic equations that define a Hidden Markov Model (HMM) for observed data that takes discrete values. Indicate what aspects of the model the following terms refer to: *initial state probabilities*, *transition matrix* and *emission matrix*.

   (b) Consider a dataset consisting of the following string of 160 symbols from the alphabet $\{A, B, C\}$:

   $$AABBBACABBBACAAAAAAAAAABBBACAAAAABACAAAAAA$$
   $$BBBBACAAAAAAAAAAAAABACABACAABBACAAABBBBACA$$
   $$AABACAAAABACAABACAAABBACAAAABBBBACABBACAA$$
   $$AAAABACABACAAABACAABBBACAAAABACABBACA$$

   Carefully analyse the string. Describe an HMM model for the string. Your description should include the number of states in the HMM, the transition matrix including the values of the elements of the matrix, the emission matrix including the values of its elements, and the initial state probabilities. Explain your reasoning.

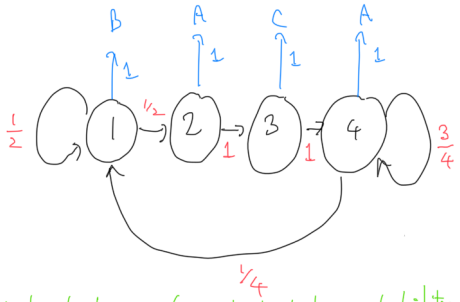
a) bookwork

$p(x_1 = k) = \pi_k$ initial state probabilities

$p(x_t = k \mid x_{t-1} = l) = T_{kl}$ transition probabilities

$p(y_t = m \mid x_t = k) = S_{mk}$ emission probabilities

b) BACA common motif & only way 'C' is emitted & only way A follows B



- only one symbol deterministically emitted from each latent state
- calculate state transition probabilities as follows

Lots of choices for initial state probability

eg $p(4) = 1$ $p(k) \propto n_k$ ↑ number of times in that state

[There are a number of other possible solutions]

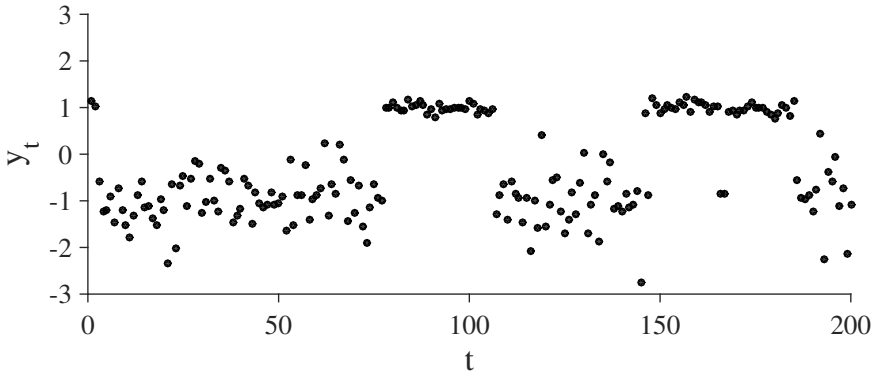
AA BB[BACA] BB[BACA] BB[BACA] AAAA AAAA BB [BACA] AAAA [BACA] AAAAA BBB [BACA] AAAAAAA AAAA[BACA] [BACA] A B [BACA] AA BBB[BACA] AA[BACA] AAA[BACA] A [BACA] AAB[BACA] AAA BBB[BACA] B[BACA] AAAAA[BACA][BACA] AA[BACA] ABB[BACA] AAA [BACA] B [BACA]

[rough counts - rounded to fractions a human might choose]

from	to		total
(1)	(1) 23	(2) 22	45
(4)	(1) 22	(4) 54	76

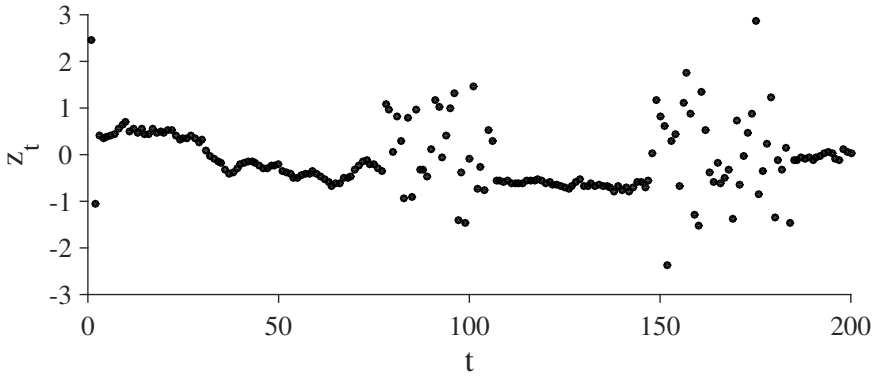
8. Probabilistic Modelling using HMMs for continuous valued observations

(a) A machine learner observes the time-series, y_t, shown below:



Suggest a suitable Hidden Markov Model (HMM) for this sequence and state the model's probabilistic equations. Indicate plausible numerical values for the parameters where possible.

(b) The machine learner is provided with a second set of observations z_t that were measured simultaneously with y_t, shown below:



Extend the HMM you proposed for part (a) so that it can jointly model the first and second set of observations.

a) Binary hidden state $S \in \{0, 1\}$

$$p(s_1 = 1) = \frac{1}{2} \qquad p(s_t \mid s_{t-1}) = \begin{bmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{bmatrix}$$

$$p(y_t \mid s_t = 1) = \mathcal{N}(y_t;\ \underbrace{1}_{\text{mean}},\ \underbrace{0.1^2}_{\text{variance}})$$

$$p(y_t \mid s_t = 0) = \mathcal{N}(y_t;\ -1,\ (1/2)^2)$$

b) $$p(z_t^{(1)}) = \mathcal{N}(z_t^{(1)};\ 0,\ 1)$$

{ This is known as a switching state-space model

$$p(z_t^{(2)} \mid z_{t-1}^{(2)}) = \mathcal{N}(z_t^{(2)};\ \lambda z_{t-1}^{(2)},\ \left(\frac{1}{2}\right)^2 (1 - \lambda^2)) \qquad \lambda = 0.99$$

$z_t = s_t z_t^{(1)} + (1 - s_t) z_t^{(2)}$ i.e. $z_t = z_t^{(1)}$ if $s_t = 1$ & $z_t = z_t^{(2)}$ if $s_t = 0$

Again rough estimates for parameter values is fine, the general structure is the main thing to convey

9. Inference in HMMs with Discrete Hidden States†

A Hidden Markov Model contains a discrete hidden state variable x_t that takes one of two values and a discrete observed state y_t that also takes one of two values. The hidden state has a transition probability,

$$\begin{bmatrix} P(x_t = 1|x_{t-1} = 1) & P(x_t = 1|x_{t-1} = 2) \\ P(x_t = 2|x_{t-1} = 1) & P(x_t = 2|x_{t-1} = 2) \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} = \begin{bmatrix} 2/3 & 1/3 \\ 1/3 & 2/3 \end{bmatrix}.$$

The filtering distribution at time $t-1$ is

$$P(x_{t-1}|y_{1:t-1}) = \begin{bmatrix} P(x_{t-1} = 1|y_{1:t-1}) \\ P(x_{t-1} = 2|y_{1:t-1}) \end{bmatrix} = \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix}.$$

(a) Compute the predictive distribution for the next hidden state variable, $P(x_t|y_{1:t-1})$.

(b) Explain how your solution to part (a) can be used to compute the filtering distribution $P(x_t|y_{1:t})$. What additional piece of information would you require to carry out this computation?

Q5 a)

$$\begin{bmatrix} p(x_t=1 \mid x_{t-1}=1) & p(x_t=1 \mid x_{t-1}=2) \\ p(x_t=2 \mid x_{t-1}=1) & p(x_t=2 \mid x_{t-1}=2) \end{bmatrix} \begin{bmatrix} p(x_{t-1}=1 \mid y_{1:t-1}) \\ p(x_{t-1}=2 \mid y_{1:t-1}) \end{bmatrix}$$

(T ≡ the transition matrix; $p(x_{t-1} \mid y_{1:t-1})$)

$$= \begin{bmatrix} p(x_t=1 \mid x_{t-1}=1)\, p(x_{t-1}=1 \mid y_{1:t-1}) + p(x_t=1 \mid x_{t-1}=2)\, p(x_{t-1}=2 \mid y_{1:t-1}) \\ p(x_t=2 \mid x_{t-1}=1)\, p(x_{t-1}=1 \mid y_{1:t-1}) + p(x_t=2 \mid x_{t-1}=2)\, p(x_{t-1}=2 \mid y_{1:t-1}) \end{bmatrix}$$

$$= \begin{bmatrix} p(x_t=1 \mid y_{1:t-1}) \\ p(x_t=2 \mid y_{1:t-1}) \end{bmatrix}$$

∴ Predictive is

$$\begin{bmatrix} 2/3 & 1/3 \\ 1/3 & 2/3 \end{bmatrix} \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix} = \begin{bmatrix} 2/12 + 3/12 \\ 1/12 + 6/12 \end{bmatrix} = \frac{1}{12} \begin{bmatrix} 5 \\ 7 \end{bmatrix}$$

b) $p(x_t \mid y_{1:t}) \propto p(y_t \mid x_t)\, p(x_t \mid y_{1:t-1})$

↑ need the likelihood to get the posterior

10. Forecasting in Linear Gaussian State Space Models*

A simple linear Gaussian state space model with scalar hidden state variables x_t has been used to model scalar observations y_t,

$$p(x_t|x_{t-1}, \lambda, \sigma^2) = \mathcal{N}(x_t; \lambda x_{t-1}, \sigma^2), \quad p(y_t|x_t, \sigma_y^2) = \mathcal{N}(y_t; x_t, \sigma_y^2).$$

The Kalman filter recursions have been used to process T observations, $y_{1:T}$, in order to return the posterior distribution over the Tth latent state, $p(x_T|y_{1:T}) = \mathcal{N}(x_T; \mu_T, \sigma_T^2)$.

(a) Explain how to transform the posterior distribution over the Tth latent state into a forecast for the observations one time step into the future, i.e. express $p(y_{T+1}|y_{1:T})$ in terms of μ_T and σ_T^2.

(b) Now provide a forecast for the observations τ time steps into the future by expressing $p(y_{T+\tau}|y_{1:T})$ in terms of μ_T and σ_T^2.

(c) What happens to $p(y_{T+\tau}|y_{1:T})$ as $\tau \to \infty$?

a)

$$p(y_{T+1} \mid y_{1:T}) = \mathcal{N}(y_{T+1};\ \lambda\mu_T,\ \lambda^2\sigma_T^2 + \sigma^2 + \sigma_y^2)$$

Calculated by passing $\mathcal{N}(x_T; \mu_T, \sigma_T^2)$ through $x_{T+1} = \lambda x_T + \sigma\varepsilon_T$ & then noting $y_{T+1} = x_{T+1} + \sigma_y n_T$ where $\varepsilon_T, n_T \sim \mathcal{N}(0,1)$

b)

$$\begin{aligned} x_{T+\tau} &= \lambda x_{T+\tau-1} + \sigma\varepsilon_{T+\tau} \\ &= \lambda(\lambda x_{T+\tau-2} + \sigma\varepsilon_{T+\tau-1}) + \sigma\varepsilon_{T+\tau} \\ &= \lambda(\lambda(\lambda x_{T+\tau-3} + \sigma\varepsilon_{T+\tau-2}) + \sigma\varepsilon_{T+\tau-1}) + \sigma\varepsilon_{T+\tau} \\ &\vdots \\ x_{T+\tau} &= \lambda^\tau x_T + \sigma\sum_{t'=0}^{\tau-1}\lambda^{t'}\varepsilon_{T+\tau-t'} \end{aligned}$$

$$\therefore p(y_{T+\tau} \mid y_{1:T}) = \mathcal{N}\left(y_{T+\tau};\ \lambda^\tau\mu_T,\ \sigma_y^2 + \sigma^2\sum_{t'=0}^{\tau-1}\lambda^{2t'} + \lambda^{2\tau}\sigma_T^2\right)$$

geometric series: $S_\tau = \sum_{t'=0}^{\tau-1}\lambda^{2t'}$ $\quad S_{\tau+1} = \lambda^2 S_\tau + 1$

c) As $\tau \to \infty$ the forecast will tend to the stationary distribution of the chain:

$$p(y_\infty \mid y_{1:T}) = \mathcal{N}\left(y_\infty;\ 0,\ \frac{\sigma^2}{1-\lambda^2} + \sigma_y^2\right)$$

$S_\infty = \lambda^2 S_\infty + 1$