MACHINE LEARNING, SPEECH & LANGUAGE TECHNOLOGY MPHIL

Wednesday 2nd November 2016   10.30 to 12.15

MLSALT1

SOLUTIONS — INTRODUCTION TO MACHINE LEARNING, SPEECH AND LANGUAGE TECHNOLOGY

STATIONERY REQUIREMENTS

SPECIAL REQUIREMENTS TO BE SUPPLIED FOR THIS EXAM

You may not start to read the questions printed on the subsequent pages of this question paper until instructed to do so.

Version RET: final solutions

1 A noisy depth sensor measures the distance to an object an unknown distance d metres away. The depth can be assumed, *a priori*, to be distributed according to a standard Gaussian distribution $p(d) = \mathcal{N}(d; 0, 1)$. The depth sensor returns y a noisy measurement of the depth, that is also assumed to be Gaussian $p(y|d, \sigma_y^2) = \mathcal{N}(y; d, \sigma_y^2)$.

(a) Compute the posterior distribution over the depth given the observation, $p(d|y, \sigma_y^2)$. [80%]

(b) What happens to the posterior distribution as the measurement noise becomes very large $\sigma_y^2 \to \infty$? Comment on this result. [20%]

Q 1a)
$$p(d|y,\sigma_y^2) = \frac{p(y|d,\sigma_y^2)\,p(d)}{p(y|\sigma_y^2)} \propto \exp\left(-\frac{1}{2\sigma_y^2}(y-d)^2 - \frac{1}{2}d^2\right)$$

$$\propto \exp\left(-\frac{1}{2}d^2\left(1+\frac{1}{\sigma_y^2}\right) + \frac{1}{\sigma_y^2}yd\right)$$

$$\propto N(d; \mu_{d|y}, \sigma_{d|y}^2) \propto \exp\left(-\frac{1}{2\sigma_{d|y}^2}(d-\mu_{d|y})^2\right)$$

$$\propto \exp\left(-\frac{1}{2\sigma_{d|y}^2}d^2 + \frac{1}{\sigma_{d|y}^2}\mu_{d|y}d\right)$$

$$\therefore \quad \sigma_{d|y}^2 = \frac{1}{1+1/\sigma_y^2} = \frac{\sigma_y^2}{1+\sigma_y^2} \quad \& \quad \mu_{d|y} = \frac{\cancel{\sigma_y^2}}{1+\sigma_y^2}\frac{1}{\cancel{\sigma_y^2}}y = \frac{y}{1+\sigma_y^2}$$

1b) As $\sigma_y^2 \to \infty$

$$\sigma_{y|d}^2 \to 1 \qquad \mu_{d|y} \to 0$$

As expected the posterior tends to the prior $p(d|y,\sigma_y^2) \to p(d)$ as $\sigma_y^2 \to \infty$ as the observation noise is infinite the data y provide no information about the depth

2 A sequence of coin tosses are observed from a biased coin $x_{1:N} = \{0,1,1,0,1,1,1,1,0\}$ where $x_n = 1$ indicates flip n was a head and $x_n = 0$ indicates that it was tails. An experimenter would like to estimate the coin's probability of landing heads, ρ, from these data.

The experimenter assumes that the coin flips are drawn independently from a Bernoulli distribution $p(x_n|\rho) = \rho^{x_n}(1-\rho)^{1-x_n}$ and uses a prior distribution of the form

$$p(\rho|n_0, N_0) = \frac{1}{Z(n_0, N_0)} \rho^{n_0}(1-\rho)^{N_0-n_0}.$$

Here n_0 and N_0 are parameters set by the experimenter to encapsulate their prior beliefs. $Z(n_0, N_0)$ returns the normalising constant of the distribution as a function of the parameters, n_0 and N_0.

(a) Compute the posterior distribution over the bias $p(\rho|x_{1:N}, n_0, N_0)$. [40%]

(b) Compute the *maximum a posteriori* (MAP) estimate for the bias. [40%]

(c) Provide an intuitive interpretation for the parameters of the prior distribution, n_0 and N_0. [20%]

Q2 a) $N = 9$ $n = \sum_n x_n = 6 = \#$ of 1s

$$p(\rho | x_{1:N}, n_0, N_0) \propto p(\rho | n_0, N_0) \prod_n p(x_n | \rho) = \frac{1}{Z_0(n_0, N_0)} \rho^{n_0}(1-\rho)^{N_0-n_0} \rho^{\sum_n x_n}(1-\rho)^{N-\sum_n x_n}$$

$$\propto \rho^{n_0+n}(1-\rho)^{N_0+N-(n+n_0)}$$

Now we can ensure normalisation by noting that this is of the same form as the prior but instead of having parameters N_0 & n_0 it has parameters N_0+N & n_0+n so the normalisation is $Z(n+n_0, N+N_0)$ $\therefore$

$$p(\rho | x_{1:N}, n_0, N_0) = \frac{1}{Z(n+n_0, N+n_0)} \rho^{n_0+n}(1-\rho)^{N_0+N-(n_0+n)}$$

2b) $$\log p(\rho \mid x_{1:N}, n_0, N_0) = \log Z + (n_0+n)\log \rho + (N_0+N-n_0-n)\log(1-\rho)$$

↑ take logs to make the derivative simpler to compute

MAP: solution

$$\frac{d}{d\rho}\log p(\rho \mid x_{1:N}, n_0, N_0) = \frac{\overbrace{n_0+n}^{n'}}{\rho} - \frac{\overbrace{N_0+N-n_0-n}^{N'-n'}}{1-\rho} = 0$$

$$\Rightarrow \quad (1-\rho)\,n' - \rho(N'-n') = 0$$

$$\Rightarrow \quad \rho = \frac{n'}{N'} = \frac{n_0+n}{N_0+N}$$

2c) n_0 & N_0 play the role of N_0 pseudo coin tosses observed before the data are seen n_0 of which are heads.

[the prior & the posterior have the same form here: this is an example of a "conjugate prior" which always have an interpretation in terms of pseudo data]

3 A data-scientist has computed a complex posterior distribution over a variable of interest, x, given observed data y, that is $p(x|y)$. They would like to return a point estimate of x to their client. The client provides the data-scientist with a reward function $R(\hat{x},x)$ that indicates their satisfaction with a point estimate $\hat{x}$ when the true state of the variable is x.

(a) Explain how to use *Bayesian Decision Theory* to determine the optimal point estimate, $\hat{x}$. [40%]

(b) Compute the optimal point estimate $\hat{x}$ in the case when the reward function is the negative square error between the point estimate and the true value, $R(\hat{x},x) = -(\hat{x}-x)^2$. Comment on your result. [60%]

Q3 a) $\arg\min_{\hat{x}} \int R(\hat{x},x)\, p(x|y)\, dx$ ← averaged over all x [Bayesian Decision Theory c.f. lecture 1]

reward function

weighted by posterior

b) $\frac{d}{d\hat{x}} \int -(\hat{x}-x)^2 p(x|y)\, dx = \int -2(\hat{x}-x)p(x|y)\, dx = 0$

$-2\hat{x} + 2\,\mathbb{E}_{p(x|y)}(x) = 0$

$\Rightarrow \hat{x} = \mathbb{E}_{p(x|y)}(x)$ = mean of the posterior

Mean of posterior is estimate of x that minimises expected square - error

4 A data-scientist has collected a regression dataset comprising N scalar inputs ($\{x_n\}_{n=1}^N$) and N scalar outputs ($\{y_n\}_{n=1}^N$). Their goal is to predict y from x and they have assumed a very simple linear model, $y_n = ax_n + \varepsilon_n$.

The data-scientist also has access to a second set of outputs ($\{z_n\}_{n=1}^N$) that are well described by the model $z_n = x_n + \varepsilon'_n$.

The noise variables ε_n and ε'_n are known to be zero mean correlated Gaussian variables

$$p\left(\left[\begin{array}{c}\varepsilon_n \\ \varepsilon'_n\end{array}\right]\right) = \mathcal{N}\left(\left[\begin{array}{c}\varepsilon_n \\ \varepsilon'_n\end{array}\right]; \mathbf{0}, \Sigma\right) \quad \text{where} \quad \Sigma^{-1} = \left[\begin{array}{cc}1 & 0.5 \\ 0.5 & 1\end{array}\right].$$

(a) Provide an expression for the log-likelihood of the parameter a. [20%]

(b) Compute the maximum likelihood estimate for a. [60%]

(c) Do the additional outputs $\{z_n\}_{n=1}^N$ provide useful additional information for estimating a? Explain your reasoning. [20%]

Q4a) $\mathcal{L}(a) = \log p(\{y_n\}_{n=1}^N, \{z_n\}_{n=1}^N \mid \{x_n\}_{n=1}^N, a) = \sum_n \log p(y_n, z_n \mid x_n, a)$

where $p(y_n, z_n \mid a, x_n) = \mathcal{N}\left(\begin{bmatrix} y_n \\ z_n \end{bmatrix}; \begin{bmatrix} ax_n \\ x_n \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}^{-1}\right)$

$$\therefore \mathcal{L}(a) = \sum_n -\frac{1}{2}\left(\begin{bmatrix} y_n \\ z_n \end{bmatrix} - \begin{bmatrix} ax_n \\ x_n \end{bmatrix}\right)^T \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix} \left(\begin{bmatrix} y_n \\ z_n \end{bmatrix} - \begin{bmatrix} ax_n \\ x_n \end{bmatrix}\right) - \frac{N}{2}\log\det(2\pi\Sigma)$$

$$= -\frac{1}{2}\sum_n (y_n - ax_n)^2 - \frac{1}{2}\sum_n (y_n - ax_n)(z_n - x_n) - \frac{1}{2}\sum_n (z_n - x_n)^2 - \frac{N}{2}\log\det 2\pi\Sigma$$

bit from just observing ys

extra bit from observing zs

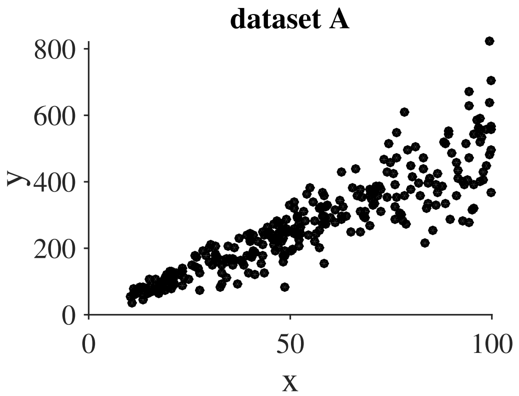
b) $$\frac{d\mathcal{L}(a)}{da} = \sum_n (y_n - ax_n)x_n + \frac{1}{2}\sum_n (z_n - x_n)x_n = 0$$

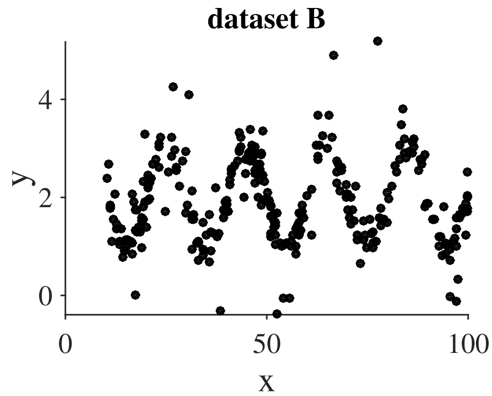
$$= \sum_n y_n x_n + \frac{1}{2}\sum_n z_n x_n - \frac{1}{2}\sum_n x_n^2 - a\sum_n x_n^2$$

$$\therefore \quad a = \left(\sum_n y_n x_n + \underbrace{\frac{1}{2}\sum_n \overbrace{(z_n - x_n)}^{\varepsilon'_n} x_n}_{\text{new contribution from observing zs}}\right) \Big/ \sum_n x_n^2 \quad \text{(max likelihood estimate)}$$

c) The additional outputs change the ML estimate of a. This means that they must provide useful information about a. They do this because the noise in z_n is correlated with the noise in y_n & so observing z_n reveals information about the noise ϵ_n & allows more accurate identification of a.

5 A machine learner observes two separate regression datasets comprising scalar inputs and outputs $\{x_n, y_n\}_{n=1}^N$ shown below.





(a) Suggest a suitable regression model, $p(y_n|x_n)$ for the dataset A. Indicate sensible settings for the parameters in your proposed model where possible. Explain your modelling choices. [50%]

(b) Suggest a suitable regression model, $p(y_n|x_n)$ for the dataset B. Indicate sensible settings for the parameters in your proposed model where possible. Explain your modelling choices. [50%]

Q5 a) $p(y_n|x_n, \theta) = \mathcal{N}(y_n; a x_n, b x_n^2)$

$a = 5 \qquad b = 1$

data lie on straight line w/ gradient 5 & noise variance increases quadratically w/ x

mean of data is 2 + sinusoid of amplitude 2 & period 20

b) $p(y_n|x_n, \theta) = \text{Student-t}\left(y_n; 2 + \sin\left(\frac{2\pi}{20} x_n\right), \frac{1}{4}, 2\right)$

any sparse dist here suitable

variance

hard to estimate by eye

parameter of Student t allowing heavy tails / outliers

[only very rough estimates of numerical values are required important part is to identify features that should be modelled underlined in blue above]

6 A data-scientist would like to summarise high dimensional data points $\mathbf{y}_n$ in terms of a single scalar variable x_n. They use an encoding weight $\mathbf{w}$ to produce the summary, $x_n = \mathbf{w}^\top \mathbf{y}_n$, and a decoding weight $\mathbf{r}$ to reconstruct the data point from the summary, $\hat{\mathbf{y}}_n = \mathbf{r} x_n$. The data-scientist would like to learn the encoding and decoding weights by optimising the squared error of the reconstruction,

$$\mathscr{C} = \sum_n ||\mathbf{y}_n - \hat{\mathbf{y}}_n||^2.$$

(a) Minimise the cost $\mathscr{C}$ with respect to the decoding weights $\mathbf{r}$, returning an expression for them in terms of x_n and $\mathbf{y}_n$. [50%]

(b) Substitute your expression for the optimised decoding weights $\mathbf{r}$ into $\mathscr{C}$ to obtain the cost purely in terms of the encoding weights $\mathbf{w}$. [20%]

(c) Now consider minimising the cost derived in part (b) with respect to the encoding weights. What is the solution? Is it unique? [30%]

It may be useful to know that the solution to the optimisation problem $\mathbf{z}^* = \arg\max_{\mathbf{z}} \frac{\mathbf{z}^\top H \mathbf{z}}{\mathbf{z}^\top \mathbf{z}}$ is the largest eigenvector of matrix H (arbitrarily scaled), $\mathbf{z}^* \propto \mathbf{e}_1$.

Q6 a) $\mathcal{C}(\underline{w}, \underline{r}) = \sum_n \sum_d (y_{dn} - r_d x_n)^2$

$$\frac{d\mathcal{C}(\underline{w}, \underline{r})}{d r_d} = \sum_n 2(y_{dn} - r_d x_n) x_n$$

$$\Rightarrow r_d = \frac{\sum_n y_{dn} x_n}{\sum_n x_n^2}$$ (just like linear regression)

6b) $C(\underline{w}, \underline{r}(\underline{w})) = \sum_n \sum_d \left(y_{nd} - \frac{\sum_{n'} y_{n'd}\, x_{n'}}{\sum_a x_a^2}\, x_n \right)^2$

this more direct using vector notation, but here I use index notation

$= \sum_{nd} y_{nd}^2 - 2 \sum_n \sum_{n'} \sum_d y_{nd}\, y_{n'd} \frac{x_n x_{n'}}{\sum_a x_a^2} + \cancel{\sum_n} \sum_{n'} \sum_{n''} y_{n'd}\, y_{n''d} \frac{x_{n'} x_{n''} \cancel{x_n^2}}{\left(\sum_a x_a^2\right)^{\cancel{2}}}$

does not depend on $\underline{w}$

$= \sum_{nd} y_{nd}^2 - \sum_{nn'd} y_{nd}\, y_{n'd} \frac{x_n x_{n'}}{\sum_a x_a^2}$

$\sum_n \underline{y}_n \underline{y}_n^T$

$\underline{w}^T \Sigma_y \Sigma_y \underline{w}$

$\Rightarrow \underset{\underline{w}}{\arg\max} \sum_{nn'd} y_{nd}\, y_{n'd} \frac{x_n x_{n'}}{\sum_a x_a^2} = \frac{\sum_{n,n'} \underline{y}_n^T \quad \underline{w}^T \underline{y}_n\, \underline{w}^T \underline{y}_{n'}}{\sum_a \underline{w}^T \underline{y}_a \underline{y}_a \underline{w}} = \frac{\underline{w}^T \underline{\underline{\Sigma}}_y \underline{\underline{\Sigma}}_y \underline{w}}{\underline{w}^T \underline{\underline{\Sigma}}_y \underline{w}}$

$\underline{w}^T \Sigma_y \underline{w}$

c) Now use $\underline{v} = \underline{\underline{\Sigma}}_y^{1/2} \underline{w}$ & solve for $\underline{v}$ to find $\underline{w}$ via $\underline{w} = \left(\Sigma_y^{1/2}\right)^{-1} \underline{v}$

$\underset{\underline{v}}{\arg\max} \frac{\underline{v}^T \Sigma_y \underline{v}}{\underline{v}^T \underline{v}} \Rightarrow \underline{v} \propto$ largest eigenvector of $\underline{\Sigma}_y = \underline{e}_1$

$\Rightarrow \underline{w} \propto$ largest eigenvector of Σ_y too $= \underline{e}_1$

matrix of eigenvectors

matrix w/ eigenvalues along diagonal

as $\Sigma_y = E^T \Lambda E$

share eigenvectors

$\Sigma_y^{1/2} = E^T \Lambda^{1/2} E$

share eigenvectors

$\Sigma_y^{-1/2} = E^T \Lambda^{-1/2} E$

$\Rightarrow \underline{w} \propto \Sigma_y^{-1/2} \underline{e}_1 \propto \underline{e}_1$

Not unique as free to scale $\underline{w}$ arbitrarily & rescale $\underline{r}$ accordingly to compensate

This is the hardest question.

7 A set of N scalar data points $\{y_n\}_{n=1}^{N}$ are modelled using a mixture of Gaussians containing two equiprobable components with unknown means (μ_0 and μ_1) and unit variances,

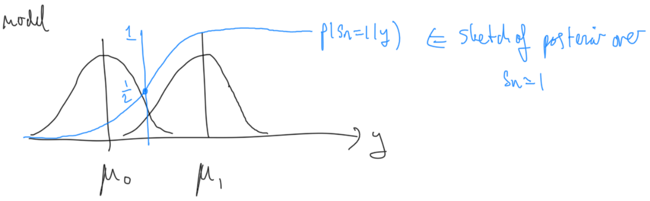
$$p(s_n = 1) = \frac{1}{2}, \quad p(y_n|s_n = 0) = \mathcal{N}(y_n; \mu_0, 1), \quad p(y_n|s_n = 1) = \mathcal{N}(y_n; \mu_1, 1).$$

(a) Compute the posterior distribution over the components, $p(s_n = 1|y_n)$ and sketch how this varies as a function of the observed data y_n. [40%]

(b) Explain how your solution to (a) can be used in the EM algorithm to estimate the component means. Your answer should include a expression for the M-step update. [40%]

(c) Do you expect the EM algorithm to overfit when used to train this model? [20%]

Q7 Sketch of model



a) $$p(s_n=1|y_n) = \frac{p(s_n=1, y_n)}{p(s_n=1,y_n) + p(s_n=0,y_n)} = \frac{1}{1 + \frac{p(s_n=0,y_n)}{p(s_n=1,y_n)}}$$

$$= \frac{1}{1 + \exp\left(-\frac{1}{2}(y_n-\mu_0)^2 + \frac{1}{2}(y_n-\mu_1)^2\right)}$$

$$= \frac{1}{1 + \exp\left(-\underbrace{\left[y_n(\mu_1-\mu_0) + \frac{\mu_1^2}{2} - \frac{\mu_0^2}{2}\right]}_{\alpha}\right)}$$

$$= \frac{1}{1+\exp(-\alpha)} \quad = \text{logistic function}$$

Just like logistic regression with weights $(\mu_1 - \mu_0)$ & bias $+\frac{\mu_1^2}{2} - \frac{\mu_0^2}{2}$

b) E-Step computes $p(s_n = 1 | y_n)$ using expression above

M-Step

$$\mu_1^{(new)} = \frac{\sum_n p(s_n = 1 | y_n) \, y_n}{\sum_n p(s_n = 1 | y_n)}$$

mean of data each weighted by probability they came from class 1

$$\mu_0^{(new)} = \frac{\sum_n \overbrace{(1 - p(s_n = 1 | y_n))}^{p(s_n = 0 | y_n)} \, y_n}{\sum_n (1 - p(s_n = 1 | y_n))}$$

as above, but weighted by prob came from class 0

c) Overfitting in a MoG model can occur when variances shrink to zero. As the variances are fixed to unity in this model over fitting is less likely.

8 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. [40%]

(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. [50%]

(c) What happens to $p(y_{T+\tau} | y_{1:T})$ as $\tau \to \infty$? [10%]

Q8 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)

$$x_{T+\tau} = \lambda x_{T+\tau-1} + \sigma \varepsilon_{T+\tau}$$

$$= \lambda(\lambda x_{T+\tau-2} + \sigma \varepsilon_{T+\tau-1}) + \sigma \varepsilon_{T+\tau}$$

$$= \lambda(\lambda(\lambda x_{T+\tau-3} + \sigma \varepsilon_{T+\tau-2}) + \sigma \varepsilon_{T+\tau-1}) + \sigma \varepsilon_{T+\tau}$$

$$\vdots$$

$$x_{T+\tau} = \lambda^\tau x_T + \sigma \sum_{t'=0}^{\tau-1} \lambda^{t'} \varepsilon_{T+\tau-t'}$$

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

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:

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

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

9 (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*. [20%]

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

AABBBACABBBACAAAAAAAAABBBACAAAAABACAAAAAABBBBACAAAAAAAA
AAAABACABACAABBACAAABBBBACAAABACAAAABACAABACAAABBACAAAA
BBBBACABBACAAAAAABACABACAAABACAABBBACAAAABACABBACA

Carefully analyse the string paying close attention to repeated patterns. 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. [80%]

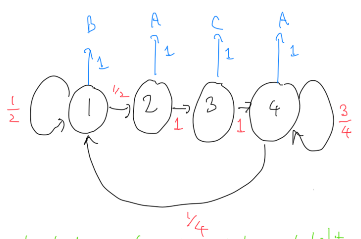
Q 9a) bookwork

$p(x_1 = k) = \pi_k$ 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

AA BB[BACA]BB[BACA]BB[BACA]AAAAAAAA BB
[BACA] AAAA [BACA] AAAAA BBB [BACA] AAAAAAA
AAAA [BACA] [BACA] AB [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]

Lots of choices for initial state probability

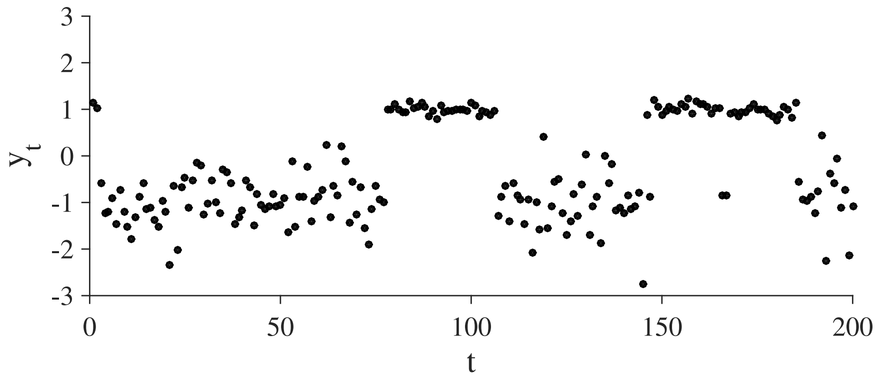
eg $p(4) = 1$ $p(k) \propto n_k$ ← number of times in that state

[There are a number of other possible solutions]

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

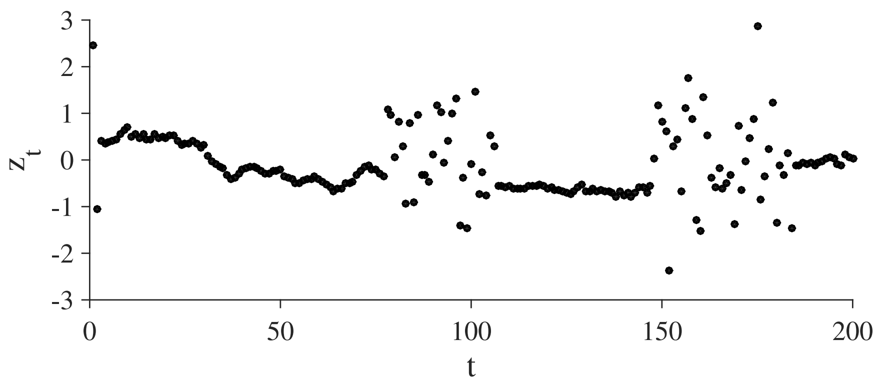
from	to ①	to ②	to ④	total
①	23	22		45
④	22		54	76

10 (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. [50%]

(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. [50%]

Q10a) 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

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

$z_t = s_t z_t^{(1)} + (1 - s_t) z_t^{(2)}$ ie $z_t = z_t^{(1)}$ 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

END OF PAPER