Engineering Tripos Part IIA THIRD YEAR

Paper 3F8: Inference

Example Sheet 2: Bayesian Linear Regression, Classification, Dimensionality Reduction and Clustering

Straightforward questions are marked †
*Tripos standard (but not necessarily Tripos length) questions are marked **

Bayesian linear regression

1. Bayesian Linear Regression

A single data point $\{x, y\}$ is fit using Bayesian linear regression. The output y is assumed to be generated from the input x according to a linear relationship that is corrupted by Gaussian noise $y = mx + c + \epsilon$. The noise ϵ is mean 0 and variance 1 so $p(y|m, c, x) = \mathcal{N}(y; mx + c, 1)$. Gaussian priors are placed on the slope m and intercept c with zero mean and unit variance, that is $p(m) = \mathcal{N}(m; 0, 1)$ and $p(c) = \mathcal{N}(c; 0, 1)$.

(a) Compute the posterior probability of the slope and intercept given the data point, that is $p(m, c|x, y)$.

(b) Show that the posterior derived in part a is consistent with the expressions for Bayesian linear regression given in lectures.

(c) Compute the posterior in the following three cases and provide explanations for why the posteriors take the form that they do.

i. $x = 0$ and $y = 0$

ii. $x = 1$ and $y = 0$

iii. $x = 100$ and $y = 100$

You might like to compute the predictive mean for these three cases i.e. the mean of $p(y^*|x^*, x, y)$ where x^* is the location of a new test input and y^* is the corresponding output.

Classification

2. Probit Classification

   Consider classification as described in lectures, but with the following model

   $$y^{(n)} = H(\mathbf{w}^\top \mathbf{x}^{(n)} + \epsilon_n)$$

   where ϵ_n is Gaussian with mean 0 and variance σ^2 and $H(\cdot)$ is the Heaviside step function.

   (a) Compute the probability $P(y^{(n)} = 1|\mathbf{x}_n, \mathbf{w}, \sigma^2)$ in terms of the Gaussian cumulative distribution. Sketch $P(y^{(n)} = 1|\mathbf{x}_n, \mathbf{w}, \sigma^2)$ as a function of the inputs $\mathbf{x}_n$ in the case where they are one dimensional.

   (b) What happens as the noise variance tends to infinity $\sigma^2 \to \infty$?

3. Multi-class Classification

   Consider a multi-class classification problem with K classes. The training labels are represented by K dimensional vectors $\mathbf{t}_n$ which has a single element set to 1, indicating the class membership, and all other values are set to 0. The inputs are multi-dimensional vectors $\mathbf{x}_n$. The goal is to use a training set of input vectors and output labels $\{\mathbf{x}_n, \mathbf{t}_n\}_{n=1}^N$ to enable prediction at unseen input locations.

   A friend suggests using a a soft-max function for this purpose which is parameterised by weights $\mathbf{W} = \{\mathbf{w}_k\}_{k=1}^K$. The output of the function is a vector, $\mathbf{y}$, with elements given by

   $$y_i(\mathbf{x}; \mathbf{W}) = \frac{\exp(\mathbf{w}_i^\top \mathbf{x})}{\sum_{k=1}^K \exp(\mathbf{w}_k^\top \mathbf{x})}.$$

   (a) What happens to the softmax function as the magnitude of the weights tends to infinity?

   (b) Interpretting the output of the softmax as $y_i = p(\mathrm{t}_i = 1|\mathbf{W}, \mathbf{x})$ write down a cost-function for training this model based on the log-probability of the training data given the weights $\mathbf{W}$ and inputs $\{\mathbf{x}_n\}_{n=1}^N$.

   (c) What is the relationship between this network and logistic regression?

Dimensionality Reduction

4. Principal Component Analysis[†]

A dataset comprises of N data points $\{x_n\}_{n=1}^N$. The data are zero-mean, $\mu = \frac{1}{N}\sum_{n=1}^N x_n = 0$, and the data-covariance is given by, $\Sigma = \frac{1}{N}\sum_{n=1}^N x_n x_n^\top$.

(a) Describe the purpose of the principal component analysis algorithm and outline one procedure for computing the first principal component of a dataset.

(b) Consider a data-covariance $\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$. Show that the two eigenvectors for this matrix are $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ and state the corresponding eigenvalues. What is the first principal component when $-1 < \rho < 0$?

5. Principal Component Analysis and Auto-Encoders[**]

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, $\hat{\mathbf{y}}_n = \mathbf{r}x_n$. This architecture is called an auto-encoder. The data-scientist would like to learn the encoding and decoding weights by optimising the squared error of the reconstruction,

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

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

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

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

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), $\mathbf{z}^* \propto \mathbf{e}_1$.

Clustering

6. 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.

7. 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 $\lim_{\Delta \to 0} \Delta \log \Delta = 0$.

8. Mixtures of Gaussians and EM*

   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. Briefly discuss how this relates to logistic classification.

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

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

The EM Algorithm

9. 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?

10. 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,

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

Selected solutions and hints

1. a) $p(m,c|y,x) = \mathcal{N}\left(\begin{bmatrix} m \\ c \end{bmatrix}; \mu^{\text{post}}, \Sigma^{\text{post}}\right)$ where $\mu^{\text{post}} = \frac{1}{x^2+2}\begin{bmatrix} yx \\ y \end{bmatrix}$ and $\Sigma^{\text{post}} = \frac{1}{x^2+2}\begin{bmatrix} 2 & -x \\ -x & x^2+1 \end{bmatrix}$

2. a) Let $\text{CDF}(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2) \mathrm{d}x$ then $P(y^{(n)} = 1|\mathbf{x}_n, \mathbf{w}, \sigma^2) = 1 - \text{CDF}(-\mathbf{w}^\top \mathbf{x}_n / \sigma^2)$

3. b) $\sum_{n=1}^{N} \sum_{k=1}^{K} t_k^{(n)} \log y_k(\mathbf{x}^{(n)}, W)$
   c) show that logistic regression (classification) is recovered

4. A simple exercise in verifying eigenvectors and interpreting eigenvalues

5. a) $r_d = \sum_n y_{d,n} x_n / \left(\sum_n x_n^2\right)$,
   b) $\mathcal{C} = \frac{\mathbf{w}^\top \Sigma_y \Sigma_y \mathbf{w}}{\mathbf{w}^\top \Sigma_y \mathbf{w}}$ where $\Sigma_y = \frac{1}{N}\sum_n \mathbf{y}_n \mathbf{y}_n^\top$,
   c) consider reparameterising using $\mathbf{v} = \Sigma_y^{1/2} \mathbf{w}$, solving for $\mathbf{v}$, and using this to find $\mathbf{w}$

6. a) $\mathbf{m}_k = \sum_{n=1}^{N} s_{n,k} \mathbf{x}_n / \left(\sum_{n=1}^{N} s_{n,k}\right)$

7. b) $\{\mathcal{KL}(q_i, p)\}_{i=1}^{6} = \{\log 2, \infty, \infty, \infty, \log 2, \infty\}$,
   c) $\{\mathcal{KL}(q_i, p)\}_{i=1}^{6} = \{\infty, \infty, \infty, \infty, \infty, \log 3/2\}$

8. a) $p(s_n = 1|y_n) = \left(1 + \exp(-(y_n(\mu_1 - \mu_0) + \mu_1^2/2 - \mu_0^2/2))\right)^{-1}$
   b) E-Step: Use (a),
   M-Step: $\mu_k^{(new)} = \sum_n p(s_n = k|y_n) y_n / \sum_n p(s_n = k|y_n)$ for $k = 0, 1$

9. a) E-Step: $q_n(d_n) = \mathcal{N}(\mu_{d_n|y_n}, \sigma^2_{d_n|y_n})$ where $\mu_{d_n|y_n} = y_n/(1+\sigma_y^2)$, $\sigma^2_{d_n|y_n} = \sigma_y^2/(1+\sigma_y^2)$
   M-Step: $\sigma_y^2 = \frac{1}{N}\sum_{n=1}^{N} y_n^2 + \frac{1}{N}\sum_n(\mu^2_{d_n|y_n} + \sigma^2_{d_n|y_n}) - \frac{2}{N}\sum_n y_n \mu_{d_n|y_n}$

10. a) $\boldsymbol{m}_k = \frac{\sum_n q_{n,k} \boldsymbol{x}_n}{\sum_n q_{n,k}}$ $\Sigma_k = \frac{\sum_n q_{n,k}(\boldsymbol{x}_n - \boldsymbol{m}_k)(\boldsymbol{x}_n - \boldsymbol{m}_k)^\top}{\sum_n q_{n,k}}$ $\pi_k = \frac{1}{N}\sum_n q_{n,k}$
    b) i) consider the free-energy before and after the partial update, ii) consider what happens in the limit $N \to \infty$ and whether it is necessary to see all of the data before you make the first M-Step.

Richard E. Turner