Engineering Tripos Part IIB FOURTH YEAR

Module 4F10: DEEP LEARNING AND STRUCTURED DATA

Examples Paper 1

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

Bayes Risk and Decision Boundaries

1. In many pattern classification problems, there is the option either to assign the pattern to one of the c classes, or to reject it as being unrecognizable. If the cost to reject is not too high, rejection may be a desirable action. For particular pattern classificatiokn task the cost of classification be defined as

$$\lambda(\omega_i|\omega_j) = \begin{array}{lll} 0 & \omega_i = \omega_j & (i.e.\ \text{(Correct classification)} \\ \lambda_r & \omega_i = \omega_0 & (i.e.\ \text{Rejection}) \\ \lambda_s & \text{Otherwise} & (i.e.\ \text{Substitution Error}) \end{array}$$

   Show that for minimum risk classification, the decision rule should associate a test vector $\boldsymbol{x}^\star$ with class ω_i, if $P(\omega_i|\boldsymbol{x}^\star) \geq P(\omega_j|\boldsymbol{x}^\star)$ for all j **and** $P(\omega_i|\boldsymbol{x}^\star) \geq 1 - \lambda_r/\lambda_s$, and reject otherwise.

2. A logistic regression discriminative classifier is to be used for a two-class, ω_1 and ω_2, classification problem. Given a test vector $\boldsymbol{x}^\star$ the posterior probability of class one, ω_1, is given by

$$P(\omega_1|\boldsymbol{x}^\star; \mathbf{a}, b) = \phi(\mathbf{a}^\mathsf{T}\boldsymbol{x}^\star + b) = \frac{1}{1 + \exp(-(\mathbf{a}^\mathsf{T}\boldsymbol{x}^\star + b))}$$

   where $\mathbf{a}$ and b are the model parameters. To train these parameters there is supervised training data, $\mathcal{D} = \{\{\boldsymbol{x}_1, y_1\}, \ldots, \{\boldsymbol{x}_N, y_N\}\}$, where $y_i \in \{\omega_1, \omega_2\}$.

   (a) Give the equation for a point $\boldsymbol{x}$ that lies on the decision boundary between classes ω_1 and ω_2.

   (b) Give an expression that can be used to train the model parameters, $\mathbf{a}$ and b, using the training data $\mathcal{D}$. Brielfy discuss how the parameters can be found.

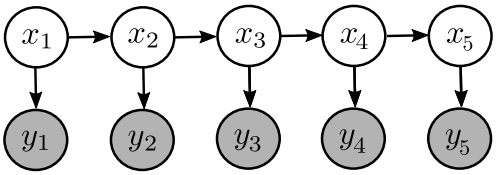
   (c) An alternative generative model is proposed using multivariate Gaussian class-conditional PDFs. Here for class ω_i $(i = 1, 2)$

$$p(\boldsymbol{x}|\omega_i; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

   Discuss how the model parameters for this generative model can be trained. What constraints on the model parameters are required to yield the same form of decision boundary as the logistic regression classifier?

Graphical Models

3. Let $p(x_1, \ldots, x_5, y_1, \ldots, y_5)$ be specified by the following Bayesian network:



   (a) What is the factorization of $p(x_1, \ldots, x_5, y_1, \ldots, y_5)$ implied by the graph?

   (b) Write the set of conditional independencies (CIs) implied by the graph.

   (c) Let $x_1, \ldots, x_5$ be discrete random variables taking ten possible values each. What is the cost of computing $p(x_1, \ldots, x_5|y_1, \ldots, y_5)$ in terms of sum operations if we ignore the CIs present in $p(x_1, \ldots, x_5, y_1, \ldots, y_5)$, that is, if we assume full generality for this distribution?

   (d) What is the cost of computing $p(x_1, \ldots, x_5|y_1, \ldots, y_5)$ in terms of sum operations if we do make use of the factorization implied by the graph?

Latent Variable Models

4. A hidden Markov model (HMM) is to be used to model a sequence of T, d-dimensional, vectors, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$. Each element of the vector $\boldsymbol{x}_t$ has a binary value, $x_{ti} \in \{0, 1\}$. The HMM comprises 2 non-emitting states, $\mathtt{s}_1$ and $\mathtt{s}_N$, and $N-2$ emitting states, $\mathtt{s}_2, \ldots, \mathtt{s}_{N-2}$. The transition probabilities (and model structure) are assumed known, but the state output distribution are to be trained using EM. The required auxiliary function (considering only a single sequence) can be expressed as

$$\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = C + \sum_{t=1}^{T} \sum_{j=2}^{N-1} P(q_t = \mathtt{s}_j | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T; \boldsymbol{\theta}) \log \left(P(\boldsymbol{x}_t | \mathtt{s}_j, \hat{\boldsymbol{\theta}}) \right)$$

   where C is independent of the values of the parameters to be estimated.

   (a) Briefly describe the meaning of each of the terms in the auxiliary function above.

   (b) Show that the forward, $\alpha_j(t)$, and backward, $\beta_j(t)$, probabilities described in the lectures can be used to find $P(q_t = \mathtt{s}_j | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T; \boldsymbol{\theta})$.

   (c) If the output probability distribution has the form

$$P(\boldsymbol{x}_t | \mathtt{s}_j; \boldsymbol{\theta}) = \prod_{i=1}^{d} \lambda_{ji}^{x_{ti}} (1 - \lambda_{ji})^{1 - x_{ti}}$$

   where state $\mathtt{s}_j$ PMF has parameters $\lambda_{j1}, \ldots, \lambda_{jd}$. Show that the estimate for the "new" parameters that maximise the auxiliary function $\hat{\lambda}_{ji}$, are given by

$$\hat{\lambda}_{ji} = \frac{\sum_{t=1}^{T} P(q_t = \mathtt{s}_j | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T; \boldsymbol{\theta}) x_{ti}}{\sum_{t=1}^{T} P(q_t = \mathtt{s}_j | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T; \boldsymbol{\theta})}$$

5. $*$ Expectation maximisation can be used to train a factor analysis model. In factor analysis the d-dimensional observations $\boldsymbol{x}$ is assumed to be generated by a process of the form

$$\boldsymbol{x} = \boldsymbol{C}\boldsymbol{z} + \boldsymbol{v}$$

where $\boldsymbol{z}$ is p-dimensional and Gaussian distributed (zero-mean identity covariance matrix), $\boldsymbol{C}$ is a $d \times p$ matrix ($d > p$), and $\boldsymbol{v}$ is d-dimensional. The parameters to estimate are the loading matrix $\boldsymbol{C}$ and the diagonal covariance matrix , $\boldsymbol{\Sigma}_{\texttt{diag}}$, where

$$p(\boldsymbol{v}) = \mathcal{N}(\boldsymbol{v}; \boldsymbol{0}, \boldsymbol{\Sigma}_{\texttt{diag}})$$

A set of n independent training samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ are available. The mean of the data is known to be zero, $\boldsymbol{0}$.

(a) The joint distribution of $\boldsymbol{x}$ and $\boldsymbol{z}$ is multivariate Gaussian distributed. Show that this distribution can be expressed as

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{C}\boldsymbol{C}^{\mathsf{T}} + \boldsymbol{\Sigma}_{\texttt{diag}} & \boldsymbol{C} \\ \boldsymbol{C}^{\mathsf{T}} & \boldsymbol{I} \end{bmatrix} \right)$$

(b) Using the previous result, show that the posterior probability of the hidden variables given the current model parameters, $\boldsymbol{C}$ and $\boldsymbol{\Sigma}_{\texttt{diag}}$ is

$$p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{C}, \boldsymbol{\Sigma}_{\texttt{diag}}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\beta}\boldsymbol{x}, \boldsymbol{I} - \boldsymbol{\beta}\boldsymbol{C})$$

where

$$\boldsymbol{\beta} = \boldsymbol{C}^{\mathsf{T}}(\boldsymbol{\Sigma}_{\texttt{diag}} + \boldsymbol{C}\boldsymbol{C}^{\mathsf{T}})^{-1}$$

You can make use of the following standard equalities

$$\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \right)$$

$$\boldsymbol{x}_1|\boldsymbol{x}_2 \sim \mathcal{N}(\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21})$$

(c) Derive the re-estimation formulae for the new model parameters $\hat{\boldsymbol{\theta}} = \{\hat{\boldsymbol{C}}, \hat{\boldsymbol{\Sigma}}_{\texttt{diag}}\}$ based on the current model parameters, $\boldsymbol{\theta} = \{\boldsymbol{C}, \boldsymbol{\Sigma}_{\texttt{diag}}\}$, using the following auxiliary function

$$\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \sum_{i=1}^{N} \int p(\boldsymbol{z}|\boldsymbol{x}_i; \boldsymbol{\theta}) \log\left(p(\boldsymbol{x}_i|\boldsymbol{z}; \hat{\boldsymbol{\theta}})\right) d\boldsymbol{z}$$

(d) Describe why factor analysis is a useful tool for covariance modelling and data visualisation.

Deep Learning

6. † A multi-layer perceptron (feed-forward, fully connected, neural network) consists of d inputs, L hidden layers with M hidden units in each hidden layer, and K output nodes. Write down an expression for the total number of weights (including biases) in the network. Describe the factors that influence the number of hidden layers, the activation functions on the output layer, and the number of hidden units.

7. † For the logistic regression function, $\phi(z)$, show that

$$\frac{\partial}{\partial z}\phi(z) = \phi(z)(1 - \phi(z))$$

   How does the nature of the activation function affect the computational cost of the error-back propagation algorithm?

8. A *leaky ReLU* activation function is to be used in a multi-layer perceptron. This activation function has the form

$$\phi(z_i) = \begin{cases} z_i; & z_i \geq 0; \\ \alpha z_i & z_i < 0 \end{cases}$$

   A large number of samples, generated from a Gaussian distribution with zero mean and a variance of σ^2, are passed through this activation function. What is the variance of the data at the output of the activation function?

   How could this information be used when initialising the network with N nodes per layer?

9. $*$ The Hessian is a useful matrix for use in the optimisation of the weights of multi-layer perceptrons.

   (a) Describe how the Hessian may be used for optimising the weights of a multi-layer perceptron. Discuss the limitations for the practical implementation of such schemes.

   (b) For the least squares error function

$$E = \sum_{p=1}^{n} E^{(p)} = \frac{1}{2} \sum_{p=1}^{n} (y(x_p) - t(x_p))^2$$

   show that the elements of the Hessian matrix can be expressed as

$$\frac{\partial^2 E}{\partial w_{ij} \partial w_{lk}} = \sum_{p=1}^{n} \frac{\partial y(x_p)}{\partial w_{ij}} \frac{\partial y(x_p)}{\partial w_{lk}} + \sum_{p=1}^{n} (y(x_p) - t(x_p)) \frac{\partial^2 y(x_p)}{\partial w_{ij} \partial w_{lk}}$$

   For the case of well trained, sufficiently powerful, network, with an infinitely large training set, show that at the minimum the second term may be ignored. This is called the *outer-product* approximation.

   (c) The Hessian after the N^{th} data point is approximated by

$$\boldsymbol{H}_N = \sum_{p=1}^{N} \boldsymbol{g}^{(p)} (\boldsymbol{g}^{(p)})^{\mathsf{T}}$$

   where

$$\boldsymbol{g}^{(p)} = \boldsymbol{\nabla} y(x_p)|_{\boldsymbol{w}_{[\tau]}}$$

   By using the equality

$$(\boldsymbol{A} + \boldsymbol{BC})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{I} + \boldsymbol{CA}^{-1}\boldsymbol{B})^{-1}\boldsymbol{CA}^{-1}$$

   where $\boldsymbol{I}$ is the identity matrix, show that

$$\boldsymbol{H}_{N+1}^{-1} = \boldsymbol{H}_N^{-1} - \frac{\boldsymbol{H}_N^{-1} \boldsymbol{g}^{(N+1)} (\boldsymbol{g}^{(N+1)})^{\mathsf{T}} \boldsymbol{H}_N^{-1}}{1 + (\boldsymbol{g}^{(N+1)})^{\mathsf{T}} \boldsymbol{H}_N^{-1} \boldsymbol{g}^{(N+1)}}$$

   Why is this a useful approximation to estimate the inverse Hessian during multi-layer perceptron training.

M.J.F. Gales & J. M. Hernandez-Lobato
Oct 2017