Applied Machine Learning Boogie

1st draft

25/03/2017

Shahid Bizzle

This random fucking document is a half-arsed attempt to bunch together questions from similar topics from past exams in Applied Machine Learning taught by David The Barbaric (years 2011 to 2016 inclusive).

(PCA comes up a fucking shitload…ain't that quaint)

The below grouping of questions is based upon the main topic of interest, although naturally there is some overlap. For example, a question may be about logistic regression but then goes on to ask how to optimise the loss function etc and so is also an optimisation q.

Also, even if a question is repeated word for word in multiple years (this does happen), I've still pasted the question from each year, so you can explicitly see just what has been repeated.

Note: The 2013 past paper has an errata page at the end of it, check this if any of the 2013 Qs seem like bollocks.

In rough order of descending frequency:

Dimensionality reduction, PCA and Autoencoders **Page 2**

Optimisation features heavily in:

- Linear Regression **Page 13**
- Logistic Regression **Page 19**
- Neural Nets and AutoDiff (dash of LSTM in 2016) **Page 23**
- Gradients **Page 26**

Nearest Neighbour methods **Page 27**

Gaussian Mixture models **Page 31**

Where the fuck is Markov? **Page 35**

Time Series and Auto-Regression **Page 37**

Sampling, MCMC and Belief Networks **Page 38**

Data scaling/transformation **Page 39**

Random shit **Page 40**

Dimensionality reduction, PCA and Autoencoders (appears in 2011, 2012, 2013, 2014, 2015 and 2016 exams)

2011

3. a. Principal Components Analysis (PCA) is a method to form a lower dimensional representation of data. For datapoints $\mathbf{x}^n, n = 1, \ldots, N$, define the matrix

$$\mathbf{X} = \left[\mathbf{x}^1, \mathbf{x}^2 \ldots, \mathbf{x}^N\right]$$

That is, for datapoints $\mathbf{x}$ with dimension D, then $\mathbf{X}$ is $D \times N$ dimensional. The data is such that the mean is zero, that is

$$\sum_{n=1}^{N} \mathbf{x}^n = \mathbf{0}$$

The covariance matrix of the data, $\mathbf{S}$, has elements

$$S_{ij} = \frac{1}{N} \sum_{n=1}^{N} x_i^n x_j^n$$

i. Explain how to write $\mathbf{S}$ in terms of matrix multiplication of $\mathbf{X}$.

[3 marks]

ii. PCA is based on a linear model of the data

$$\mathbf{x}^n \approx \mathbf{M}\mathbf{y}^n$$

where $\mathbf{M}$ is a $D \times H$ dimensional matrix, and each $\mathbf{y}^n$ is a H dimensional vector, with $H < D$. With reference to an orthogonal matrix

$$\mathbf{R}^\mathsf{T}\mathbf{R} = \mathbf{I}$$

explain why there is no unique setting in general for $\mathbf{M}$ and $\mathbf{y}^n$. Use also a diagram to explain the geometric meaning of this result.

[4 marks]

iii. PCA is typically described in terms of the eigen-decomposition of $\mathbf{S}$. Explain how this procedure works and also discuss the computational complexity of performing PCA based on directly computing the eigen-decomposition of $\mathbf{S}$.

[4 marks]

iv. Whilst there no unique setting to the best lower dimensional linear representation, PCA nevertheless provides a unique setting for the lower dimensional representation. Explain why this is, and what additional criterion PCA uses over least squared reconstruction error.

[4 marks]

v. An alternative way to perform PCA is based on the singular value decomposition (SVD) of the matrix $\mathbf{X}$. Explain why this is related to the eigen-decomposition of $\mathbf{S}$ and explain the computational complexity of this approach to performing PCA compared to directly computing the eigen-decomposition of $\mathbf{S}$.

[5 marks]

vi. The matrix $\mathbf{X}$, with elements $X_{i,n}$ represents the real-valued rating for a user n given to film i. Only a small fraction of the elements of the matrix $\mathbf{X}$ are known since each user has only seem a small number of films. Explain how PCA with missing data can be used to 'complete' the missing entries of $\mathbf{X}$, thereby forming a prediction for the rating of a user to the missing entries in $\mathbf{X}$. Explain also a method to implement PCA in this missing data case.

[5 marks]

b. PCA can be consider a form of matrix factorisation. An alternative matrix factorisation method is probabilistic latent semantic analysis (PLSA) (also called non-negative matrix factorisation). This takes a positive matrix $\mathbf{X}$ whose entries all sum to 1:

$$\sum_{ij} X_{ij} = 1, \qquad 0 \leq X_{ij} \leq 1$$

and forms an approximation based on

$$X_{ij} \approx \sum_{k=1}^{H} U_{ik} V_{kj}$$

for positive U and V with $\sum_i U_{ik} = 1$ and $\sum_k V_{kj} = 1$.

i. In the lectures, we compared the application of PCA and PLSA on a set of face images. Explain what are the typical characteristics of the 'eigenfaces' compared with the 'plsa' faces.

[4 marks]

ii. Describe a way to train PLSA based on an interpretation of X, U and V in terms of probability distributions.

[4 marks]

[Total 33 marks]

3. Principal Components Analysis (PCA) is a method to form a lower dimensional representation of data. For datapoints $\mathbf{x}^n, n = 1,\ldots,N$, define the matrix

$$\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N]$$

That is, for datapoints $\mathbf{x}$ with dimension D, then $\mathbf{X}$ is $D \times N$ dimensional. The data is such that the mean is zero, that is

$$\sum_{n=1}^{N} \mathbf{x}^n = \mathbf{0}$$

The covariance matrix of the data, $\mathbf{S}$, has elements

$$S_{ij} = \frac{1}{N} \sum_{n=1}^{N} x_i^n x_j^n$$

a. Explain why PCA is often used as a pre-processing step in machine learning and explain what the geometric meaning of PCA is.

[4 marks]

b. Explain how to write $\mathbf{S}$ in terms of matrix multiplication of $\mathbf{X}$.

[3 marks]

c. PCA is typically described in terms of the eigen-decomposition of $\mathbf{S}$. Explain how this procedure works and also discuss the computational complexity of performing PCA based on directly computing the eigen-decomposition of $\mathbf{S}$.

[4 marks]

d. Consider the situation in which the datapoints $\mathbf{x}^n$ are very sparse – that is, only a few elements of each vector $\mathbf{x}^n$ are non-zero, resulting also in a sparse matrix $\mathbf{S}$. Describe a computationally efficient procedure to estimate the principal direction (the largest eigenvector of $\mathbf{S}$) and explain why this is efficient.

[4 marks]

e. Continuing with the sparse datapoint scenario, can you conceive a technique that would enable one to perform full PCA (not just the principal eigenvector) efficiently?

[4 marks]

f. An alternative way to perform PCA is based on the singular value decomposition (SVD) of the matrix $\mathbf{X}$. Explain why this is related to the eigen-decomposition of $\mathbf{S}$ and explain the computational complexity of this approach to performing PCA compared to directly computing the eigen-decomposition of $\mathbf{S}$.

[5 marks]

g. PCA can be considered as representing the i^{th} component of the n^{th} datapoint using

$$x_i^n \approx \sum_j y_j^n b_{ji}$$

where b_{ji} are the elements of the basis vectors for the j^{th} basis vector, and y_j^n is the corresponding coefficient. In the case that some of the components of x_i^n are missing, we cannot find the optimal PCA solution by the standard eigen-approach. In this case, define the least squares objective

$$E(\mathbf{B}, \mathbf{Y}) = \sum_{n=1}^{N} \sum_{i=1}^{D} \gamma_i^n \left[x_i^n - \sum_j y_j^n b_{ji} \right]^2$$

where

$$\gamma_i^n = \begin{cases} 1 & \text{if } x_i^n \text{ exists} \\ 0 & \text{if } x_i^n \text{ is missing} \end{cases}$$

Derive a procedure for minimising $E(\mathbf{B}, \mathbf{Y})$ that is guaranteed to decrease the objective function at each stage of the iteration, and for which each iteration corresponds to the solution of a set of linear equations.

[9 marks]

[Total 33 marks]

2013

3. Principal Components Analysis (PCA) is a method to form a lower dimensional representation of data. For datapoints $\mathbf{x}^n, n = 1, \ldots, N$, define the matrix

$$\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N]$$

That is, for datapoints $\mathbf{x}$ with dimension D, then $\mathbf{X}$ is $D \times N$ dimensional. The data is such that the mean is zero, that is

$$\sum_{n=1}^{N} \mathbf{x}^n = \mathbf{0}$$

The covariance matrix of the data, $\mathbf{S}$, has elements

$$S_{ij} = \frac{1}{N} \sum_{n=1}^{N} x_i^n x_j^n$$

a. Explain why PCA is often used as a pre-processing step in machine learning and explain what the geometric meaning of PCA is.

[4 marks]

b. Explain how to write $\mathbf{S}$ in terms of matrix multiplication of $\mathbf{X}$.

[3 marks]

c. PCA is typically described in terms of the eigen-decomposition of $\mathbf{S}$. Explain how this procedure works and also discuss the computational complexity of performing PCA based on directly computing the eigen-decomposition of $\mathbf{S}$.

[4 marks]

d. Consider the situation in which the datapoints $\mathbf{x}^n$ are very sparse – that is, only a few elements of each vector $\mathbf{x}^n$ are non-zero, resulting also in a sparse matrix $\mathbf{S}$. Describe a computationally efficient procedure to estimate the principal direction (the largest eigenvector of $\mathbf{S}$) and explain why this is efficient.

[4 marks]

e. Continuing with the sparse datapoint scenario, can you conceive a technique that would enable one to perform full PCA (not just the principal eigenvector) efficiently?

[4 marks]

f. An alternative way to perform PCA is based on the singular value decomposition (SVD) of the matrix $\mathbf{X}$. Explain why this is related to the eigen-decomposition of $\mathbf{S}$ and explain the computational complexity of this approach to performing PCA compared to directly computing the eigen-decomposition of $\mathbf{S}$.

[5 marks]

g. PCA can be considered as representing the i^{th} component of the n^{th} datapoint using

$$x_i^n \approx \sum_j y_j^n b_{ji}$$

where b_{ji} are the elements of the basis vectors for the j^{th} basis vector, and y_j^n is the corresponding coefficient. In the case that some of the components of x_i^n are missing, we cannot find the optimal PCA solution by the standard eigen-approach. In this case, define the least squares objective

$$E(\mathbf{B}, \mathbf{Y}) = \sum_{n=1}^{N} \sum_{i=1}^{D} \gamma_i^n \left[x_i^n - \sum_j y_j^n b_{ji} \right]^2$$

where

$$\gamma_i^n = \begin{cases} 1 & \text{if } x_i^n \text{ exists} \\ 0 & \text{if } x_i^n \text{ is missing} \end{cases}$$

Derive a procedure for minimising $E(\mathbf{B}, \mathbf{Y})$ that is guaranteed to decrease the objective function at each stage of the iteration, and for which each iteration corresponds to the solution of a set of linear equations.

[9 marks]

[Total 33 marks]

b. Explain what is meant by an autoencoder neural network.

[3 marks]

c. Consider an autoencoder with a single hidden layer ($L = 2$). When the transfer function at the output layer is the identity, $\sigma^L(x) = x$, derive an expression for the optimal weight matrices $\mathbf{W}^2, \mathbf{W}^1$ and relate this to Principal Components Analysis. What would be the optimal weights for an autoencoder with a larger number of layers, $L > 2$ but with the identity transfer function on the output layer?

[10 marks]

3. Principal Components Analysis (PCA) is a method to form a lower dimensional representation of data. For datapoints $\mathbf{x}^n, n = 1, \ldots, N$, define the matrix

$$\mathbf{X} = \left[\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\right]$$

That is, for datapoints $\mathbf{x}$ with dimension D, then $\mathbf{X}$ is $D \times N$ dimensional. The data is such that the mean is zero, that is

$$\sum_{n=1}^{N} \mathbf{x}^n = \mathbf{0}$$

The covariance matrix of the data, $\mathbf{S}$, has elements

$$S_{ij} = \frac{1}{N} \sum_{n=1}^{N} x_i^n x_j^n$$

a. Explain why PCA is often used as a pre-processing step in machine learning and explain what the geometric meaning of PCA is.

[4 marks]

b. Explain how to write $\mathbf{S}$ in terms of matrix multiplication of $\mathbf{X}$.

[3 marks]

c. PCA is typically described in terms of the eigen-decomposition of $\mathbf{S}$. Explain how this procedure works and also discuss the computational complexity of performing PCA based on directly computing the eigen-decomposition of $\mathbf{S}$.

[4 marks]

d. Consider the situation in which the datapoints $\mathbf{x}^n$ are very sparse – that is, only a few elements of each vector $\mathbf{x}^n$ are non-zero, resulting also in a sparse matrix $\mathbf{S}$. Describe a computationally efficient procedure to estimate the principal direction (the largest eigenvector of $\mathbf{S}$) and explain why this is efficient.

[4 marks]

e. Continuing with the sparse datapoint scenario, can you conceive a technique that would enable one to perform full PCA (not just the principal eigenvector) efficiently?

[4 marks]

f. An alternative way to perform PCA is based on the singular value decomposition (SVD) of the matrix $\mathbf{X}$. Explain why this is related to the eigen-decomposition of $\mathbf{S}$ and explain the computational complexity of this approach to performing PCA compared to directly computing the eigen-decomposition of $\mathbf{S}$.

[5 marks]

g. PCA can be considered as representing the i^{th} component of the n^{th} datapoint using

$$x_i^n \approx \sum_j y_j^n b_{ji}$$

where b_{ji} are the elements of the basis vectors for the j^{th} basis vector, and y_j^n is the corresponding coefficient. In the case that some of the components of x_i^n are missing, we cannot find the optimal PCA solution by the standard eigen-approach. In this case, define the least squares objective

$$E(\mathbf{B}, \mathbf{Y}) = \sum_{n=1}^{N} \sum_{i=1}^{D} \gamma_i^n \left[x_i^n - \sum_j y_j^n b_{ji} \right]^2$$

where

$$\gamma_i^n = \begin{cases} 1 & \text{if } x_i^n \text{ exists} \\ 0 & \text{if } x_i^n \text{ is missing} \end{cases}$$

Derive a procedure for minimising $E(\mathbf{B}, \mathbf{Y})$ that is guaranteed to decrease the objective function at each stage of the iteration, and for which each iteration corresponds to the solution of a set of linear equations.

[9 marks]

h. Explain the method of Fisher's Linear Discriminants and derive an explicit formula for the projection vector.

[6 marks]

[Total 39 marks]

2. Principal Components Analysis (PCA) is a method to form a lower dimensional representation of data. For datapoints $\mathbf{x}^n, n = 1,\ldots,N$, define the matrix

$$\mathbf{X} = \left[\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\right]$$

That is, for datapoints $\mathbf{x}$ with dimension D, then $\mathbf{X}$ is $D \times N$ dimensional. The data is such that the mean is zero, that is

$$\sum_{n=1}^{N} \mathbf{x}^n = \mathbf{0}$$

K-dimensional PCA aims to find a representation

$$\mathbf{x}^n \approx \sum_{k=1}^{K} y_k^n \mathbf{b}^k$$

where $\mathbf{b}^1, \ldots, \mathbf{b}^K$ are 'basis' vectors and y_k^n are the coefficients.

a. Explain how to efficiently compute the basis vectors and coefficients in order to minimise the squared loss between the approximation and each $\mathbf{x}^n$, namely

$$\sum_{n=1}^{N} \left(\mathbf{x}^n - \sum_{k=1}^{K} y_k^n \mathbf{b}^k \right)^2$$

[8 marks]

b. Explain how Autoencoders can also be used to find low dimensional representations of data and explain how PCA relates to an Autoencoder.

[5 marks]

c. Consider an Autoencoder with structure $\mathbf{x} \rightarrow \mathbf{h} \rightarrow \tilde{\mathbf{x}}$, trained to minimise the squared loss

$$\sum_{n=1}^{N} (\tilde{\mathbf{x}}^n - \mathbf{x}^n)^2$$

with $\mathbf{h}^n = f(\mathbf{A}\mathbf{x}^n)$ and $\tilde{\mathbf{x}}^n = \mathbf{B}\mathbf{h}^n$ for matrices $\mathbf{A}$, $\mathbf{B}$ and a non-linear function f.

For K-dimensional $\mathbf{h}$, is this non-linear procedure in principle more powerful than K-dimensional PCA, in the sense that it has a lower squared loss? Explain fully your answer.

[6 marks]

d. For N datapoints $\mathbf{x}^1, \ldots, \mathbf{x}^N$, explain how it is possible to obtain essentially perfect reconstructions of these datapoints using an Autoencoder with N units in the bottleneck layer.

[3 marks]

e. When training a deep Autoencoder (say more than 8 layers) explain why it is important to initialise the parameters of Autoencoders carefully. Suggest a criterion to initialise the parameters and explain the motivation behind this approach.

[5 marks]

f. PCA can be considered a form of matrix factorisation. An alternative matrix factorisation method is probabilistic latent semantic analysis (PLSA) (also called non-negative matrix factorisation). This takes a positive matrix $\mathbf{X}$ whose entries all sum to 1:

$$\sum_{ij} X_{ij} = 1, \qquad 0 \leq X_{ij} \leq 1$$

and forms an approximation based on

$$X_{ij} \approx \sum_{k=1}^{H} U_{ik} V_{kj}$$

for matrices U and V non-negative entries and $\sum_i U_{ik} = 1$ and $\sum_k V_{kj} = 1$.

i. In the lectures, we compared the application of PCA and PLSA on a set of face images. Explain what are the typical characteristics of the 'eigenfaces' compared with the 'plsa' faces.

[4 marks]

ii. Derive an algorithm to find U and V based on an interpretation of X, U and V in terms of probability distributions.

[8 marks]

[Total 39 marks]

Linear regression (appears in 2011, 2012, 2014, and 2016 exams)

2011

2. a. Explain why, in many real-world machine learning applications, relatively simple methods are often considered.

[3 marks]

b. Linear regression is a popular method. For a dataset of inputs $\mathbf{x}$ and scalar outputs y, $\{(\mathbf{x}^n, y^n), n = 1, \ldots, N\}$, linear regression is the model

$$y = \mathbf{w}^\mathsf{T}\mathbf{x}$$

i. The least squares criterion sets $\mathbf{w}$ based on minimising

$$\sum_{n=1}^{N} (y^n - \mathbf{w}^\mathsf{T}\mathbf{x}^n)^2$$

Show that the optimal solution is given by

$$\mathbf{w} = \left(\sum_{n=1}^{N} \mathbf{x}^n (\mathbf{x}^n)^\mathsf{T} \right)^{-1} \left(\sum_{n=1}^{N} y^n \mathbf{x}^n \right)$$

[5 marks]

ii. Explain why, in practice, it is not recommend to find $\mathbf{w}$ by using matrix inversion as above, and explain an alternative procedure.

[3 marks]

iii. Explain the issue of overfitting and describe a modification to the above method that may prevent overfitting. Give the corresponding optimal solution for $\mathbf{w}$.

[3 marks]

c. i. Show that the least squares error criterion from part b(i) above can be written in the form

$$E(\mathbf{w}) = \mathbf{w}^\mathsf{T}\mathbf{A}\mathbf{w} - 2\mathbf{w}^\mathsf{T}\mathbf{b} + c$$

for suitably defined $\mathbf{A}$, $\mathbf{b}$, c.

[4 marks]

ii. Explain the concept of gradient descent optimisation for minimising $E(\mathbf{w})$ and explain why this is, in general, an inefficient procedure for this problem.

[2 marks]

iii. Explain the concept of line-search optimisation and show that for a line going through the point $\mathbf{w}_k$ and direction $\mathbf{p}_k$,

$$\mathbf{w} = \mathbf{w}_k + \lambda \mathbf{p}_k$$

the optimal point on the line to minimise the squared error $E(\mathbf{w})$ is given when

$$\lambda = \frac{(\mathbf{b} - \mathbf{A}\mathbf{p}_k)^\top \mathbf{p}_k}{\mathbf{p}_k^\top \mathbf{A}\mathbf{p}_k}$$

[5 marks]

iv. Explain the meaning of 'conjugate direction' and why this means that the optimum of $E(\mathbf{w})$ can be found by optimising along each conjugate direction independently.

[4 marks]

v. One setting to find conjugate vectors 'on the fly' is to set at iteration k

$$\mathbf{g}_k = \mathbf{A}\mathbf{x}_k - \mathbf{b}, \qquad \beta = \frac{\mathbf{g}_k^\top \mathbf{g}_k}{\mathbf{g}_{k-1}^\top \mathbf{g}_{k-1}}$$

and then define the new conjugate direction

$$\mathbf{p}_k = -\mathbf{g}_k + \beta \mathbf{p}_{k-1}$$

Explain why using conjugate gradients can therefore be fast to implement when the matrix $\mathbf{A}$ is sparse.

[4 marks]

[Total 33 marks]

2012

2. a. For a dataset of inputs $\mathbf{x}$ and scalar outputs y, $\{(\mathbf{x}^n, y^n), n = 1, \ldots, N\}$, linear regression is the model

$$y = \mathbf{w}^\mathsf{T}\mathbf{x}$$

i. The least squares criterion sets $\mathbf{w}$ based on minimising

$$E(\mathbf{w}) \equiv \sum_{n=1}^{N} (y^n - \mathbf{w}^\mathsf{T}\mathbf{x}^n)^2$$

Show that the optimal solution is given by

$$\mathbf{w} = \left(\sum_{n=1}^{N} \mathbf{x}^n (\mathbf{x}^n)^\mathsf{T} \right)^{-1} \left(\sum_{n=1}^{N} y^n \mathbf{x}^n \right)$$

[5 marks]

ii. Explain why, in practice, it is not recommend to find $\mathbf{w}$ by using matrix inversion as above, and explain an alternative procedure.

[3 marks]

iii. Consider a simple gradient descent procedure to find $\mathbf{w}$. For iteration $k+1$, component i of the new weight vector $\mathbf{w}^{k+1}$ is given by

$$w_i^{k+1} = w_i^k - \eta \frac{\partial E(\mathbf{w})}{\partial w_i}$$

where $\eta > 0$. Show that we may write the update in vector form

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \eta \sum_n (y^n - (\mathbf{w}^k)^\mathsf{T}\mathbf{x}^n)\mathbf{x}^n$$

[4 marks]

iv. Show that, starting from $\mathbf{w}^1 = \mathbf{0}$, after a single gradient update, for an input $\mathbf{x}$, the output is given by

$$-\eta \sum_n y^n \mathbf{x}^\mathsf{T}\mathbf{x}^n$$

[2 marks]

v. The components of the vector $\mathbf{x}$ represent physical distances measured in metric units. Show that representing $\mathbf{x}$ in terms of imperial units of distance can be expressed by

$$\mathbf{x} = \Lambda\tilde{\mathbf{x}}$$

where Λ is a diagonal matrix and $\tilde{\mathbf{x}}$ is the imperial-units representation of $\mathbf{x}$.

[2 marks]

vi. Based on a single update starting from the zero vector, explain why for gradient descent the output of the regression function depends on whether imperial or metric units are used to represent the input vector.

[4 marks]

b. i. Explain the concept of line-search optimisation and show that for a line going through the point $\mathbf{w}_k$ and direction $\mathbf{p}_k$,

$$\mathbf{w} = \mathbf{w}_k + \lambda\mathbf{p}_k$$

the optimal point on the line to minimise the squared error $E(\mathbf{w})$ is given when

$$\lambda = \frac{(\mathbf{b} - \mathbf{A}\mathbf{p}_k)^{\mathsf{T}}\mathbf{p}_k}{\mathbf{p}_k^{\mathsf{T}}\mathbf{A}\mathbf{p}_k}$$

[5 marks]

ii. Explain the concept of Newton methods for optimisation and derive an explicit update formula for the case of linear regression.

[3 marks]

iii. Discuss also the computational issues involved with using Newton methods in optimising more general objective functions.

[3 marks]

iv. For the case of linear regression, the Newton update is invariant with respect to linear transformations of the inputs. Given a mathematical explanation why this.

[2 marks]

[Total 33 marks]

2014

1. Consider the linear regression problem for training data with vector input $\mathbf{x}^n$ and scalar output y^n, $n = 1, \ldots, N$:

$$\mathbf{w}_{opt} = \arg\min_{\mathbf{w}} E(\mathbf{w})$$

where we define the regularised total square loss

$$E(\mathbf{w}) = \sum_{n=1}^{N} \left(y^n - \mathbf{w}^\top \mathbf{x}^n \right)^2 + \lambda \mathbf{w}^\top \mathbf{w}$$

   a. Derive an explicit expression for the optimal weight vector $\mathbf{w}_{opt}$ in terms of the training data and regularisation constant λ. Give also an estimation for the computational complexity required to find $\mathbf{w}_{opt}$ using this expression.

   [10 marks]

   b. Describe the gradient descent procedure for finding the minimum of $E(\mathbf{w})$, explaining also any potential practical advantages or disadvantages of this approach.

   [3 marks]

   c. Describe the Newton procedure for finding the minimum of $E(\mathbf{w})$, explaining also any potential practical advantages or disadvantages of this approach. You should give an explicit update formula for the new weight vector in terms of the old weight vector so that this could be implemented directly by a computer programmer.

   [5 marks]

   d. Describe the conjugate gradients procedure for finding the minimum of $E(\mathbf{w})$, explaining also any potential practical advantages or disadvantages of this approach. You should give an explicit update formula for the new weight vector in terms of the old weight vector so that this could be implemented directly by a computer programmer.

   [10 marks]

   e. Consider the case that each training vector $\mathbf{x}$ is sparse, with only $0 \leq s \leq 1$ of the elements of the vector being non-zero. Give estimates for the computational complexity of finding $\mathbf{w}_{opt}$ based on the above procedures, namely (batch) gradient descent, Newton's method and conjugate gradients.

   [7 marks]

3. a. For input-output training points $(\mathbf{x}^n, y^n)$, $n = 1, \ldots, N$, where each input $\mathbf{x}^n$ is a vector and each output y^n is a scalar, the squared loss of a linear regression model is

$$E(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \left(y^n - \mathbf{w}^\top \mathbf{x}^n \right)^2$$

i. Compute the gradient and Hessian of this objective function and show that $E(\mathbf{w})$ is convex.

[4 marks]

ii. Explain what Stochastic Gradient Descent is and how it could be used to find the $\mathbf{w}$ that minimises $E(\mathbf{w})$.

[4 marks]

iii. In the case that the input vectors are sparse (only a fraction f of the elements of each $\mathbf{x}^n$ are non-zero), explain what computational savings this has when implementing gradient descent.

[4 marks]

iv. Explain how Conjugate Gradients could be used to find the $\mathbf{w}$ that minimises $E(\mathbf{w})$ and what computational savings can be made when the input vectors $\mathbf{x}^n$ are sparse.

[4 marks]

b. Consider a multi-class classification problem with input vector $\mathbf{x}^n$ and corresponding class label $c^n \in \{1, \ldots, C\}$. The softmax log likelihood objective is to maximise

$$L(\mathbf{w}_1, \ldots, \mathbf{w}_C) \equiv \sum_{n=1,\ldots,N} \log p(c^n | \mathbf{x}^n)$$

where

$$p(c^n | \mathbf{x}^n) = \frac{e^{\mathbf{w}_{c^n}^\top \mathbf{x}^n}}{\sum_{c=1}^{C} e^{\mathbf{w}_c^\top \mathbf{x}^n}}$$

i. Calculate the gradient vectors

$$\frac{\partial}{\partial \mathbf{w}_c} L$$

[4 marks]

ii. Show that $L(\mathbf{w}_1, \ldots, \mathbf{w}_C)$ is jointly concave.

[4 marks]

[Total 24 marks]

Logistic regression (appears in 2012 and 2013 exams)

2012

4. Consider binary classification problems with class $c \in \{0,1\}$. Logistic Regression models the probability of input vector $\mathbf{x}$ being in class $c = 1$ as

$$p(c=1|\mathbf{x},\mathbf{w}) = \sigma\left(\mathbf{w}^{\mathsf{T}}\mathbf{x}\right)$$

for input vector $\mathbf{x}$ and weight (parameter) vector $\mathbf{w}$, where

$$\sigma(x) = \frac{e^x}{1+e^x}$$

The dataset is $\mathcal{D} = \{(\mathbf{x}^n, c^n), n = 1,\ldots,N\}$.

a. Assuming that the data is independently and identically distributed, show that the gradient of the log likelihood is given by

$$\sum_{n=1}^{N}(2c^n - 1)\sigma(1-2c^n)\mathbf{x}^n$$

[5 marks]

b. Consider training logistic regression in which the components of the input vector $\mathbf{x}$ can be on very different scales and also potentially highly correlated. Discuss how you might adapt the gradient ascent based approach to training logistic regression.

[5 marks]

c. Missing data is a common problem in practice. Describe how you might adapt logistic regression when there are missing elements in the input data vectors. Discuss the advantages and disadvantages of your approaches.

[6 marks]

d. Often in practice data is 'imbalanced'. For example, for a classification problem there may be many datapoints corresponding to 'healthy' people with only a few datapoints corresponding to people with a rare disease. Describe approaches to deal with such imbalanced data in training a classifier. Discuss carefully how to assess the performance of your classifier in the case of highly imbalanced data.

[4 marks]

e. In classification we often have an associated 'loss' function for each class. For example it might be that it is important in a medical situation to detect cancer, even if some of the detections are actually false. We can use a loss function $L(c_{true}, c_{pred})$ to measure our loss when our predicted class is c_{pred} whereas the truth is c_{true}.

i. Explain how to adapt logistic regression to minimize expected loss $\mathcal{L}(\mathbf{w})$

$$\mathcal{L}(\mathbf{w}) \equiv \sum_{n=1}^{N} \left\langle L(c_{true}^n, c_{pred}^n) \right\rangle_{p(c_{pred}^n | \mathbf{x}^n, \mathbf{w})}$$

on $\mathcal{D}$ and derive a gradient based training scheme.

[8 marks]

ii. Comment on the geometric structure of the objective function $\mathcal{L}(\mathbf{w})$ and discuss any potential computational issues involved in training this model.

[5 marks]

[Total 33 marks]

2013

4. Consider binary classification problems with class $c \in \{0, 1\}$. Logistic Regression models the probability of the D-dimensional input vector $\mathbf{x}$ being in class $c = 1$ as

$$p(c = 1|\mathbf{x}, \mathbf{w}) = \sigma\left(\mathbf{w}^\top \mathbf{x}\right)$$

for input vector $\mathbf{x}$ and weight (parameter) vector $\mathbf{w}$, where

$$\sigma(x) = \frac{e^x}{1 + e^x}$$

The dataset is $\mathcal{D} = \{(\mathbf{x}^n, c^n), n = 1, \ldots, N\}$.

a. Assuming that the data is independently and identically distributed, show that the log likelihood is given by

$$L(\mathbf{w}) \equiv \sum_{n=1}^{N} c^n \log \sigma\left(\mathbf{w}^\top \mathbf{x}^n\right) + (1 - c^n) \log \sigma\left(-\mathbf{w}^\top \mathbf{x}^n\right)$$

[3 marks]

b. Show that the gradient of the log likelihood is given by

$$g_i \equiv \frac{\partial}{\partial w_i} L = \sum_{n=1}^{N} \left(c^n - \sigma\left(\mathbf{w}^\top \mathbf{x}^n\right)\right) x_i^n$$

[3 marks]

c. Show that the Hessian is given by

$$H_{ij} \equiv \frac{\partial^2}{\partial w_i \partial w_j} L = -\sum_{n=1}^{N} \sigma\left(\mathbf{w}^\top \mathbf{x}^n\right) \sigma\left(-\mathbf{w}^\top \mathbf{x}^n\right) x_i^n x_j^n$$

[3 marks]

d. Show that the Hessian is negative (semi) definite. That is, for any non-zero vector $\mathbf{y}$

$$\mathbf{y}^\top \mathbf{H} \mathbf{y} \leq 0$$

and explain what this means in terms of the geometry of the log likelihood.

[3 marks]

e. Explain how Newton's method (using the Hessian) can be used to find the maximum likelihood parameter vector $\mathbf{w}$.

[2 marks]

f. Describe the computational complexity of a single Newton update, both in terms of time and storage cost. For D-dimensional inputs $\mathbf{x}$, comment on the practical usefulness of the Newton method for high dimensional inputs $D \gg 1$.

[3 marks]

g. Conjugate gradients is an alternative way to find the optimal maximum likelihood $\mathbf{w}$. Given a search direction $\mathbf{p}_k$ at the k^{th} iteration of the algorithm, conjugate gradients needs to maximise

$$L(\mathbf{w}_k + \alpha \mathbf{p}_k)$$

with respect to α Explain whether the optimal α can be found in closed form and, if not, how the optimal α can be obtained.

[3 marks]

h. For sparse data in which only on average S components of an input vector $\mathbf{x}^n$ will be non-zero.

i. Explain the computational complexity of computing the gradient vector $\mathbf{g}$.

[3 marks]

ii. Give an estimate of the number of operations required to find the optimal weight vector using conjugate gradients.

[3 marks]

i. Describe what is meant by 'batch' versus 'online' gradient methods.

[3 marks]

j Consider maximum likelihood learning for $\mathbf{w}$ in logistic regression using gradient ascent.

i. Explain if 'online' gradient ascent will converge for linearly separable training data.

[2 marks]

ii. Explain if 'online' gradient ascent will converge for training data that is not linearly separable.

[2 marks]

[Total 33 marks]

Neural Nets and AutoDiff (appears in 2014, 2015 and 2016 exams)

2014

2. Consider a set of training data with vector input $\mathbf{x}^n$ and vector output $\mathbf{y}^n$. For input vector $\mathbf{x}$, a neural network produces output vector

$$\mathbf{f}(\mathbf{x}|\mathcal{W}) = \sigma^L(\mathbf{W}^L\mathbf{h}^{L-1}),$$

where the hidden layer values are recursively defined by

$$\mathbf{h}^l = \sigma^l\left(\mathbf{W}^l\mathbf{h}^{l-1}\right), \quad l = 2,\ldots,L-1, \qquad \mathbf{h}^1 = \sigma^1\left(\mathbf{W}^1\mathbf{x}\right)$$

The total set of parameters is given by $\mathcal{W} = \left\{\mathbf{W}^1,\ldots,\mathbf{W}^L\right\}$ and the dimension of layer l is defined by d_l where $dim(\mathbf{W}^l) = d_l \times d_{l-1}$. You may assume that the transfer functions $\sigma^1,\ldots,\sigma^L$ are known.

a. For the squared loss objective function

$$E(\mathcal{W}) = \sum_{n=1}^{N} \left(\mathbf{y}^n - \mathbf{f}(\mathbf{x}^n|\mathcal{W})\right)^2$$

derive an efficient recursive procedure to compute the gradient of $E(\mathcal{W})$ with respect to all the parameters $\mathcal{W}$.

[12 marks]

b. Explain what is meant by an autoencoder neural network.

[3 marks]

c. Consider an autoencoder with a single hidden layer ($L = 2$). When the transfer function at the output layer is the identity, $\sigma^L(x) = x$, derive an expression for the optimal weight matrices $\mathbf{W}^2, \mathbf{W}^1$ and relate this to Principal Components Analysis. What would be the optimal weights for an autoencoder with a larger number of layers, $L > 2$ but with the identity transfer function on the output layer?

[10 marks]

[Total 25 marks]

2015

1. a. Explain what is meant by Forward Automatic Differentiation (AutoDiff) and give two procedures (one exact and the other an approximation) that compute the gradient of a subroutine $\mathbf{x}$ with respect to its arguments $\mathbf{x}$, giving time complexities of the approaches.

[5 marks]

b. Explain what is meant by Reverse AutoDiff. Describe the algorithm and its time complexity. Give an example to illustrate how the algorithm works.

[7 marks]

c. Consider a set of training data with vector input $\mathbf{x}^n$ and vector output $\mathbf{y}^n$. For input vector $\mathbf{x}$, a neural network produces output vector

$$\mathbf{f}(\mathbf{x}|\mathcal{W}) = \sigma^L(\mathbf{W}^L\mathbf{h}^{L-1}),$$

where the hidden layer values are recursively defined by

$$\mathbf{h}^l = \sigma^l\left(\mathbf{W}^l\mathbf{h}^{l-1}\right), \quad l = 2,\ldots,L-1, \qquad \mathbf{h}^1 = \sigma^1\left(\mathbf{W}^1\mathbf{x}\right)$$

The total set of parameters is given by $\mathcal{W} = \left\{\mathbf{W}^1,\ldots,\mathbf{W}^L\right\}$ and the dimension of layer l is defined by d_l where $dim(\mathbf{W}^l) = d_l \times d_{l-1}$. You may assume that the transfer functions $\sigma^1,\ldots,\sigma^L$ are known.

i. For the squared loss objective function

$$E(\mathcal{W}) = \sum_{n=1}^{N}\left(\mathbf{y}^n - \mathbf{f}(\mathbf{x}^n|\mathcal{W})\right)^2$$

derive an efficient recursive procedure to compute the gradient of $E(\mathcal{W})$ with respect to all the parameters $\mathcal{W}$ and relate this to reverse mode AutoDiff.

[5 marks]

ii. Explain how to compute the gradient of $E(\mathcal{W})$ when parameters of the network are tied.

[3 marks]

[Total 20 marks]

2016

1. a. Describe Forward Automatic Differentiation (AutoDiff) and give two procedures (one exact and the other an approximation) that compute the gradient of a subroutine $f(\mathbf{x})$ with respect to its arguments $\mathbf{x}$, giving time complexities of the approaches.

[5 marks]

b. Describe Reverse AutoDiff and explain its time complexity.

[7 marks]

c. Explain how to use Reverse AutoDiff to efficiently calculate the gradient with respect to θ_1, θ_2 of

$$\sum_{n=1}^{N} (y^n - sin(\theta_1 + \theta_2 x^n))^2$$

where (x^n, y^n) are the input-output values for the n^{th} datapoint. Your computation graph should have nodes representing elementary functions. Annotate your graph suitably and define the forward and backward passes explicitly.

[8 marks]

d. Consider a time series prediction problem in which, given a sequence of inputs $x_1, x_2, \ldots, x_t$, we make a prediction $\tilde{y}_t$ for the output at time t. To do this we define:

$$\begin{aligned} h_1 &= x_1 \\ h_t &= f(x_t, h_{t-1}, A) \qquad t > 1 \\ \tilde{y}_t &= g(h_t, B) \end{aligned}$$

where A and B are parameters and f and g are some (unspecified) functions. The objective is to find parameters A and B that minimise the loss

$$\sum_{t=1}^{T} (y_t - \tilde{y}_t)^2$$

Explain how to use Reverse AutoDiff to efficiently calculate the gradient of this loss function with respect to A and B.

[7 marks]

e. An input-output time-series (x_t, y_t), $t = 1, \ldots, T$ can be modelled by a recurrent LSTM (Long Short Term Memory) network. Explain the essential components of an LSTM network and what difficulties it tries to overcome (compared to standard recurrent networks).

[10 marks]

[Total 37 marks]

Gradients (a question about gradients/optimisation appears independent of an associated model in 2015)

2015

2. a. Explain why the gradient of a function points along the direction of maximal change.
[2 marks]

b. Prove that for a convex function $f(\mathbf{x})$ which has a Hessian with eigenvalues less than L, then under the gradient descent procedure

$$f(\mathbf{x}_T) - f(\mathbf{x}^*) \leq \frac{1}{2\varepsilon T}(\mathbf{x}_1 - \mathbf{x}^*)^2$$

where $\mathbf{x}_1$ is the starting point, $\mathbf{x}_T$ is the value after $T-1$ gradient descent steps and the learning rate $\varepsilon \leq 1/L$.

[20 marks]

c. Explain what is meant by Nesterov's Accelerated Gradient procedure and compare its theoretical convergence rate with that of standard gradient descent for a convex function.

[4 marks]

d. Explain what is meant by Newton's method for optimisation of a function $f(\mathbf{x})$ and show that the position $\mathbf{x}_T$ obtained after T updates of the algorithm is invariant with respect to a linear coordinate transformation $\mathbf{x} = \mathbf{M}\mathbf{y}$.

[10 marks]

e. Explain what is meant by the Gauss-Newton optimisation method and what advantages this has over the standard Newton method.

[5 marks]

[Total 41 marks]

Nearest Neighbours methods (appears in the 2013 and 2014 exams)

2013

1. This question concerns nearest neighbour methods.

   a. i. Explain what is meant by nearest neighbour classification for a dataset of N examples, $\mathcal{D} = \{(\mathbf{x}^n, c^n), n = 1, \ldots, N\}$, for D-dimensional inputs $\mathbf{x}$ and discrete class labels c. Explain how to classify a new input $\mathbf{x}$.

   [2 marks]

   ii. For the Euclidian distance

   $$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{D} (x_i - y_i)^2}$$

   and the dataset $\mathcal{D}$ above, describe the computational complexity (both time and storage) of computing the nearest neighbour classifier for a novel input $\mathbf{x}$.

   [2 marks]

   iii. Explain what is meant by a metric distance and show that the Euclidean distance is a metric.

   [4 marks]

   b. Orchard's algorithm is a way to speed up the calculation of the nearest neighbour (for a metric distance) of a query $\mathbf{q}$ to a set of D-dimensional training vectors $\{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$.

   i. For a metric distance show that if $d(\mathbf{q}, \mathbf{x}^i) \leq \frac{1}{2} d(\mathbf{x}^i, \mathbf{x}^j)$ then $d(\mathbf{q}, \mathbf{x}^i) \leq d(\mathbf{q}, \mathbf{x}^j)$. Draw a picture to describe this mathematical result.

   [4 marks]

   ii. Explain in detail how Orchard's algorithm works.

   [4 marks]

   iii. Give an example dataset and query for which Orchard's algorithm will not be faster than simply explicitly evaluating the nearest neighbour by computing all distances from the query point $\mathbf{q}$.

   [2 marks]

   iv. Give an example dataset and query for which Orchard's algorithm will be faster than simply explicitly evaluating the nearest neighbour by computing all distances from the query point $\mathbf{q}$.

   [2 marks]

   c. Consider a metric distance and a set of datapoints $\mathbf{x}^i$, $i \in I$ for which $d(\mathbf{q}, \mathbf{x}^i)$ has already been computed.

   i. Show that we form the lower bound

   $$d(\mathbf{q}, \mathbf{x}^j) \geq \max_{i \in I} \left\{ d(\mathbf{q}, \mathbf{x}^i) - d(\mathbf{x}^i, \mathbf{x}^j) \right\} \equiv L_j$$

   [2 marks]

ii. Explain how the Approximating and Elimination Algorithm (AESA) uses the above result to attempt to speed up the computation of the nearest neighbour of a query point $\mathbf{q}$.

[2 marks]

d. The KD tree is a hierarchical data-structure that can be used to potentially speed up nearest neighbour search.

i. Explain how to form a KD tree, giving an example of a dataset (of two-dimensional data) and the corresponding KD tree.

[3 marks]

ii. Consider a query vector $\mathbf{q}$. Let's imagine that we have partitioned the datapoints into those with first dimension x_1 less than a defined value v (to its 'left'), and those with a value greater or equal to v (to its 'right').

$$\mathcal{L} = \{\mathbf{x}^n : x_1^n < v\}, \qquad \mathcal{R} = \{\mathbf{x}^n : x_1^n \geq v\}$$

Let's also say that our current best nearest neighbour candidate is $\mathbf{x}^i \in \mathcal{R}$ and that this point has squared Euclidean distance $\delta^2 = (\mathbf{q} - \mathbf{x}^i)^2$ from $\mathbf{q}$. Show that if $(v - q_1)^2 \geq \delta^2$, then $(\mathbf{x} - \mathbf{q})^2 > \delta^2$.

[3 marks]

iii. Explain how the above result can be used with the KD tree to search for the nearest neighbour of a query. Use your example KD tree above and an example query point to describe how to find the nearest neighbour using the KD tree.

[3 marks]

[Total 33 marks]

2014

3. This question concerns nearest neighbour methods.

a. i. Explain what is meant by nearest neighbour classification for a dataset of N examples, $\mathcal{D} = \{(\mathbf{x}^n, c^n), n = 1, \ldots, N\}$, for D-dimensional inputs $\mathbf{x}$ and discrete class labels c. Explain how to classify a new input $\mathbf{x}$.

[2 marks]

ii. For the Euclidian distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{D} (x_i - y_i)^2}$$

and the dataset $\mathcal{D}$ above, describe the computational complexity (both time and storage) of computing the nearest neighbour classifier for a novel input $\mathbf{x}$.

[2 marks]

iii. Explain what is meant by a metric distance and show that the Euclidean distance is a metric.

[4 marks]

b. Orchard's algorithm is a way to speed up the calculation of the nearest neighbour (for a metric distance) of a query $\mathbf{q}$ to a set of D-dimensional training vectors $\{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$.

i. For a metric distance show that if $d(\mathbf{q}, \mathbf{x}^i) \leq \frac{1}{2} d(\mathbf{x}^i, \mathbf{x}^j)$ then $d(\mathbf{q}, \mathbf{x}^i) \leq d(\mathbf{q}, \mathbf{x}^j)$. Draw a picture to describe this mathematical result.

[4 marks]

ii. Explain in detail how Orchard's algorithm works.

[4 marks]

iii. Give an example dataset and query for which Orchard's algorithm will not be faster than simply explicitly evaluating the nearest neighbour by computing all distances from the query point $\mathbf{q}$.

[2 marks]

iv. Give an example dataset and query for which Orchard's algorithm will be faster than simply explicitly evaluating the nearest neighbour by computing all distances from the query point $\mathbf{q}$.

[2 marks]

c. Consider a metric distance and a set of datapoints $\mathbf{x}^i$, $i \in I$ for which $d(\mathbf{q}, \mathbf{x}^i)$ has already been computed.

i. Show that we may form the lower bound

$$d(\mathbf{q}, \mathbf{x}^j) \geq \max_{i \in I} \left\{ d(\mathbf{q}, \mathbf{x}^i) - d(\mathbf{x}^i, \mathbf{x}^j) \right\} \equiv L_j$$

[2 marks]

ii. Explain how the Approximating and Elimination Search Algorithm (AESA) uses the above result to attempt to speed up the computation of the nearest neighbour of a query point $\mathbf{q}$.

[2 marks]

d. The KD tree is a hierarchical data-structure that can be used to potentially speed up nearest neighbour search.

i. Explain how to form a KD tree, giving an example of a dataset (of two-dimensional data) and the corresponding KD tree.

[3 marks]

ii. Consider a query vector $\mathbf{q}$. Let's imagine that we have partitioned the datapoints into those with first dimension x_1 less than a defined value v (to its 'left'), and those with a value greater or equal to v (to its 'right'):

$$\mathcal{L} = \{\mathbf{x}^n : x_1^n < v\}, \qquad \mathcal{R} = \{\mathbf{x}^n : x_1^n \geq v\}$$

Let's also say that our current best nearest neighbour candidate is $\mathbf{x}^i$ and that this point has squared Euclidean distance $\delta^2 = (\mathbf{q} - \mathbf{x}^i)^2$ from $\mathbf{q}$. Show that if $q_1 \geq v$ and $(v - q_1)^2 > \delta^2$, then $(\mathbf{x} - \mathbf{q})^2 > \delta^2$.

[3 marks]

iii. Explain how the above result can be used with the KD tree to search for the nearest neighbour of a query. Use your example KD tree above and an example query point to describe how to find the nearest neighbour using the KD tree.

[3 marks]

[Total 33 marks]

Gaussian Mixture models (appears in the 2011 and 2013 exams)

2011

1. This question concerns fitting Gaussian Mixture Models (GMMs)

$$p(\mathbf{x}) = \sum_{i=1}^{K} w_i \mathcal{N}(\mathbf{x}|\mathbf{m}_i, \Sigma_i)$$

where $0 \leq w_i \leq 1$, $\sum_{i=1}^{K} w_i = 1$ and

$$\mathcal{N}(\mathbf{x}|\mathbf{m}, \Sigma) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m})^{\mathsf{T}}\Sigma^{-1}(\mathbf{x}-\mathbf{m})}$$

The GMM will be trained to fit a set of N datapoints $\mathcal{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$.

a. Explain why a GMM can be used to cluster the datapoints.

[4 marks]

b. For independently and identically distributed data, show that the log likelihood is

$$\sum_{n=1}^{N} \log \sum_{i=1}^{K} w_i \mathcal{N}(\mathbf{x}^n|\mathbf{m}_i, \Sigma_i)$$

[2 marks]

c. Show that by setting a GMM mean $\mathbf{m}_i$ to a datapoint $\mathbf{x}^n$, the log likelihood becomes infinite when the corresponding covariance matrix $\Sigma_i = \mathbf{0}$. Explain why for maximum likelihood training of the GMM, one must place constraints on the covariance matrices in order to find sensible solutions.

[4 marks]

d. Explain a reasonable initialisation procedure for training the GMM on the datapoints $\mathcal{X}$.

[2 marks]

e. For the case of constrained covariance matrices $\Sigma_i = \sigma_i^2 \mathbf{I}$ (that is, each covariance matrix is proportional to the identity matrix), show that the M-step of the EM algorithm is given by

$$\sigma_i^2 = \frac{1}{D} \sum_{n=1}^{N} p^{old}(n|i)(\mathbf{x}^n - \mathbf{m}_i)^2$$

where D is the dimension of each datapoint and $p^{old}(n|i)$ is suitably defined.

[4 marks]

f. Explain the relation between the GMM under Expectation Maximisation training, and the K-means algorithm.

[4 marks]

g. Explain how the K-means algorithm can be used to send compressed versions of the datapoints.

[3 marks]

h. Explain why the log likelihood of the GMM can be numerically difficult to represent and explain how to effectively approximate the quantity

$$\log \sum_{i=1}^{K} w_i e^{-a_i}$$

where each $a_i \geq 0$ and one or more of the a_i is very large.

[4 marks]

i. You have two classes of datapoints. Class 1 data is represented by the set of datapoints $\mathcal{X}_1$ and class 2 data by the datapoints $\mathcal{X}_2$. You fit a GMM to each of the classes separately. This gives parameters $\{\mathbf{w}^1, \mathbf{m}_i^1, \Sigma_i^1, i = 1, \ldots, K\}$ for the GMM for class 1 and parameters $\{\mathbf{w}^2, \mathbf{m}_i^2, \Sigma_i^2, i = 1, \ldots, K\}$ for the GMM for class 2.

   i. Explain how to make a classifier $p(c|\mathbf{x})$ using the two GMMs.

   [3 marks]

   ii. Explain why the classification of a test point $\mathbf{x}$ very far from the training data is likely to be very extreme (only one class will dominate $p(c|\mathbf{x})$), and suggest a way to heal this behaviour.

   [3 marks]

[Total 33 marks]

2013

b. Consider a Gaussian mixture model of D-dimensional data

$$p(\mathbf{x}) = \sum_{k=1}^{K} \frac{1}{(2\pi\sigma^2)^{(D/2)}} \exp\left(-\frac{1}{2\sigma^2}\left(\mathbf{x}-\mu^k\right)^2\right)$$

i On a 32 bit machine, Matlab considers 2^z equal to zero, for $z < -1075$. For a point $\mathbf{x}$, the largest contribution to the mixture model is given by a particular k, namely

$$\frac{1}{(2\pi\sigma^2)^{(D/2)}} \exp\left(-\frac{1}{2\sigma^2}\left(\mathbf{x}-\mu^k\right)^2\right)$$

Setting $\sigma^2 = 1$, show that for the mixture model to have a non-zero probability in Matlab, we must have (for the natural log)

$$D \leq \frac{2 \times 1075 \log 2}{\log(2\pi)}$$

and hence estimate the largest dimension D for which the Matlab can assign a non-zero probability.

[5 marks]

ii. Explain how the `logsumexp` trick can be used to compute the log likelihood of the mixture model.

[3 marks]

iii. On a 32 bit machine, Matlab considers 2^z equal to infinity, for $z > 1024$. Show that the maximal dimension D for which the log likelihood is computable in Matlab is

$$D \leq \frac{2^{1025}}{\log(2\pi)}$$

and comment on the difference between this maximal dimension, and the maximal dimension of the raw probability itself.

[3 marks]

c. Training data from class 1 is represented by the Gaussian mixture model placing a Gaussian component on each datapoint from class 1:

$$p(\mathbf{x}|c=1) = \sum_k \frac{1}{(2\pi\sigma^2)^{(D/2)}} \exp\left(-\frac{1}{2\sigma^2}\left(\mathbf{x}-\mathbf{x}^k\right)^2\right)$$

where the sum over the indices k includes only those datapoints $\mathbf{x}^k$ that belong to class 1. Similarly, data from class 2 is represented by the Gaussian mixture model placing a Gaussian component on each datapoint from class 2:

$$p(\mathbf{x}|c=2) = \sum_l \frac{1}{(2\pi\sigma^2)^{(D/2)}} \exp\left(-\frac{1}{2\sigma^2}\left(\mathbf{x}-\mathbf{x}^l\right)^2\right)$$

where the sum over the indices l includes only those datapoints $\mathbf{x}^l$ that belong to class 2.

i. Explain how to form a classifier for a novel input vector $\mathbf{x}^*$, paying attention to any practical numerical difficulties that might be encountered.

[4 marks]

ii. Explain how to adapt the above classifier to make it robust to outliers (data-points $\mathbf{x}^*$ that are not close to any of the training data points from either class).

[3 marks]

iii. Show that in the limit $\sigma^2 \to 0$, the above classifier is equivalent to the nearest neighbour classifier.

[3 marks]

Where the fuck is Markov? (appears in the 2011 and 2012 exams)

2011

b. A Hidden Markov Model (HMM) is a standard model used in the analysis of time-series.

i. For discrete outputs (visible variables) $v_t \in \{1,\ldots,V\}$ and discrete latent (hidden) variables $h_t \in \{1,\ldots,H\}$, $t = 1,\ldots,T$, give a probabilistic description of the HMM.

[5 marks]

ii. Explain what is meant by 'filtering' and show that, for $\alpha(h_t) \equiv p(h_t, v_{1:t})$, the following recursion holds

$$\alpha(h_t) = p(v_t|h_t) \sum_{h_{t-1}} p(h_t|h_{t-1})\alpha(h_{t-1})$$

and describe a suitable initialisation for this recursion.

[4 marks]

iii. Explain the practical difficulties involved in implementing the above recursion and explain how these problems can be overcome.

[2 marks]

iv. 'Smoothing' is the computation of $p(h_t|v_{1:T})$. Derive a recursive way to compute these quantities.

[6 marks]

v. Explain why the 'filtering' and 'smoothing' recursions can be applied to continuous valued data v_t.

[3 marks]

c. You're hired by a hedge fund that wishes to track the changes in 'volatility' $f(h_t)$. The volatility values are discretised and indexed by $h_t \in \{1,\ldots,H\}$, for some given function f. So, for example, when h is in state 3, we have some volatility value $f(3)$. The volatility is a latent (hidden) variable, from which an observation is generated by

$$p(v_t|h_t) = \mathcal{N}(v_t|0, f(h_t))$$

That is, the observation distribution is a zero mean Gaussian with variance $f(h_t)$.

i. Explain how to use a HMM to track 'volatility' changes, and explain what filtering, smoothing and prediction of volatility refer to in this case.

[3 marks]

ii. For a large number of discrete states H, explain the computational complexity of tracking volatility under the constraint that, given the volatility state at h_t, only a few states $L < H$ at h_{t+1} will be accessible.

[3 marks]

2012

d. For continuous latent states h_t and observations v_t, $t = 1, \ldots, T$, consider a model

$$p(v_{1:T}, h_{1:T}) = p(v_1|h_1)p(h_1)\prod_{t=2}^{2} p(v_t|h_t)p(h_t|h_{t-1})$$

Filtering relates to inference of $p(h_t|v_{1:t})$.

i. Show that the exact filtering updates are given by the recursion

$$p(h_t|v_{1:t}) \propto p(v_t|h_t) \int_{h_{t-1}} p(h_t|h_{t-1})p(h_{t-1}|v_{1:t-1})$$

[4 marks]

ii. Explain how particle filtering can be used to approximate the filtered distribution $p(h_t|v_{1:t})$.

[4 marks]

Time Series and Auto-Regression (appears in the 2011 and 2014 exams)

2011

4. a. Time-series form an important practical arena for machine learning problems. Give a list of areas in which time-series analysis is important, and explain the role of time-series models in each application.

[7 marks]

2014

f. Explain what is meant by an Auto-Regressive (AR) model for a time-series $y_1, \ldots, y_T$ and derive an explicit expression for the optimal AR coefficients in the least squares sense.

[7 marks]

Sampling and MCMC (appears in the 2012 exam)

2012

1. This question concerns sampling.

   a. i. Explain what is meant by sampling and how S samples $x^1, \ldots, x^S$ from a univariate distribution $p(x)$ may be used to approximate an expectation

   $$\bar{f} \equiv \int p(x) f(x) dx$$

   [4 marks]

   ii. Explain what is meant by 'perfect' (also called 'exact') sampling and why this property is useful in terms of the accuracy with which S samples approximate $\bar{f}$.

   [4 marks]

   b. Consider a Belief-Network distribution on a set of variables $\mathbf{x} = \{x_1, \ldots, x_N\}$

   $$p(\mathbf{x}) = \prod_{i=1}^{N} p(x_i | \text{pa}(x_i))$$

   where $\text{pa}(x_i)$ are the parental variables of variable x_i.

   i. Explain what is meant by ancestral (also called 'forward') sampling and how this can be used to draw exact samples from a Belief Network.

   [4 marks]

   ii. In the case that there is evidence, explain how to adapt ancestral sampling to draw samples from the distribution on non-evidential variables $\mathbf{x}_{nonev}$, conditioned on the evidential variables $\mathbf{x}_{ev}$:

   $$p(\mathbf{x}_{nonev} | \mathbf{x}_{ev}), \qquad \mathbf{x} = \mathbf{x}_{ev} \cup \mathbf{x}_{nonev}$$

   Describe the efficiency of this approach.

   [4 marks]

   iii. Describe Gibbs sampling and explain how this works in the case of sampling from a Belief Network with evidence.

   [4 marks]

   iv. Explain under what circumstances Gibbs sampling can be made to be a perfect sampler.

   [3 marks]

   c. Explain what is meant by Markov-Chain Monte Carlo sampling and whether in general this is an exact sampling method.

   [2 marks]

Data scaling/transformation (appears in the 2013 exam)

2013

2. a. Consider a dataset of inputs $\{\mathbf{x}^n, n = 1, . . , N\}$.

i. Show that the transformation

$$\mathbf{z}^n = \mathbf{x}^n - \mu$$

where

$$\mu = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}^n$$

results in a new dataset that has zero mean, that is

$$\sum_{n=1}^{N} \mathbf{z}^n = \mathbf{0}$$

[3 marks]

ii. For zero mean data $\sum_{n=1}^{N} \mathbf{w}^n = \mathbf{0}$, describe a 'scaling' transformation

$$z_i^n = \lambda_i x_i^n$$

that makes each component of $\mathbf{z}$ have unit variance, that is

$$\frac{1}{N}\sum_{n=1}^{N} (z_i^n)^2 = 1$$

[3 marks]

iii. For zero mean data $\sum_{n=1}^{N} \mathbf{w}^n = \mathbf{0}$, describe and explain the computational cost of finding a 'whitening' transformation matrix $\mathbf{M}$

$$\mathbf{z}^n = \mathbf{M}\mathbf{x}^n$$

that makes $\mathbf{z}$ have unit covariance. That is

$$\frac{1}{N}\sum_{n=1}^{N} \mathbf{z}^n (\mathbf{z}^n)^\top = \mathbf{I}$$

[4 marks]

iv. With the help of a diagram, explain the difference between the two above 'scaling' and 'whitening' transformations.

[2 marks]

Random shit (appears in the 2011 and 2012 exams)

2011

5. External speakers from industry gave presentations throughout the course. Write a brief summary of one (or more) speaker's presentation (you may choose the speakers), discussing the application area, the challenges faced and any issues involved with applying machine learning. Where possible, include specific technical details of the speaker's presentation.

[Total 33 marks]

2012

5. The coursework consisted of developing solutions to the Heritage Healthcare challenge. Write a description of your approach, taking into account the following points:

- Describe the approaches that your team took and in particular highlight your personal contribution.
- Your description should include the technical details of how you trained and tested the models, what significant design decisions you took and how the project was managed.
- Your description should include sufficient technical detail such that one of your fellow students would be able to implement your most successful approach.

[Total 33 marks]