SOLUTION: DO NOT DISTRIBUTE

TBD
09:30 BST
Duration: 2 hours
Additional Time: 30 minutes
Timed exam — fixed start time

DEGREES of MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

Introduction to Data Science and Systems ¶ COMPSCI 5089

(Answer All 4 Questions)

This examination paper is an open book, online assessment and is worth a total of 60 marks.

1. **(a)** You are designing an application for clothing shops to predict clothes size based on customer height and weight. Suppose we have a **clothing dataset** with height, weight and the corresponding T-shirt size of several customers.

customer ID	height	weight	size
U1	170	60	M
U2	172	60	M
U3	173	61	M
U4	173	64	L
U5	175	67	L
U6	175	66	L

You can represent this dataset based on their vector representations by regarding height and weight as two dimensions. Now there is a new client **Abel** (U0) whose height is 173cm and weight is 62kg. You are asked to predict the T-shirt size for Abel.

(i) Calculate the Euclidean distance (L2 Norm) between the new point and the existing points.

[3]

Solution:
0.5 mark for each correct calculation. Keeping other decimal places rather than 2 decimal places will not reduce their marks.
d(U0,U1)=3.61 **[0.5]**
d(U0,U2)=2.24 **[0.5]**
d(U0,U3)=1.00 **[0.5]**
d(U0,U4)=2.00 **[0.5]**
d(U0,U5)=5.39 **[0.5]**
d(U0,U6)=4.47 **[0.5]**

(ii) Predict the size of Abel, based on the kNN algorithm, with $k = 3$ and the above calculated distances. Justify your prediction.

[2]

Solution:
The three most closest neighbours are U2 (M), U3 (M) and U4 (L). **[1]**
So you should predict M as the T-shirt size for Abel. **[1]**

(b) *For all answers, include in your answer document both code and the output of that code.*

(i) Calculate the covariance matrix for the clothing dataset using numpy.

[1]

Solution:

```
cov = np.cov(users, rowvar=False)
# cov: array([[3.6, 5.2],
#             [5.2, 9.6]])
```

[1]

(ii) Calculate the eigenvector and eigenvalues the covariance matrix using numpy.

[2]

Solution:

```
evals, evecs = np.linalg.eig(cov)
# evals: array([ 0.59666759, 12.60333241])

# array([[-0.86594528, -0.50013875],
#        [ 0.50013875, -0.86594528]])
```

[2]

(iii) Dimensionality reduction. Map the **clothing dataset** into principal component with the largest eigenvalue of its covariance matrix.

[2]

Solution:
The largest eigenvalue is evals[1], so we should map data into evecs[:,1].

```
users@evecs[:,1]
# array([-136.98030494, -137.98058245, -139.34666648,
#         -141.94450232, -145.54261566, -144.67667038])
```

[2]

(c) (i) Find SVD for $A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$, you should include full working in your solution.

[3]

Solution:

- Solution 1:

$A = U\Sigma V^T$.
$A^T A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ with eigenvalues $\lambda_1 = 1$ and $\lambda_2 = 0$ and corresponding eigenvectors $\mathbf{u}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\mathbf{u}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Hence $U = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 \end{bmatrix}$ **[1]**
$\Sigma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ **[1]**

$AA^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ with eigenvalues $\lambda_1 = 1$, $\lambda_2 = 0$, and $\lambda_3 = 0$ and corresponding eigenvectors $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ and $\mathbf{v}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$.
Hence $V^T = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix}$ **[1]**

- Solution 2:

  Another feasible solution is to use `np.linalg.svd` to solve this question.

```
a = np.array([[1,0],[0,0],[0,0]])
np.linalg.svd(a, full_matrices=True)
# u    [1., 0., 0.],
#      [0., 1., 0.],
#      [0., 0., 1.]
# s    [1,0]
#      [0,0]
#      [0,0]
# vt   [1,0]
#      [0,1]
```

(ii) State the relations between determinant, matrix inversion and non-singular.

[2]

Solution:
The inverse of matrix exists only if its determinant value is a non-zero value **[1]**
A matrix is non-singular if the determinant is different from zero **[1]**

2. Consider a tennis player, Ed Balls, who wants to prepare for a competition match against an opponent—let's call him Frank Racket. In order to prepare for the match, Ed has acquired records of the 100 previous matches of his opponent and wants to study statistics of Frank's play to choose where to focus his training.

(Here is a quick summary of the rules of tennis: `https://protennistips.net/tennis-rules/`)

Ed is interested in studying Frank's serve as this can be an important strategic advantage.

- For a serve to be valid, it must pass the net and bounce in the diagonally opposite service box.
- If the first serve is a fault (eg, hits the net or bounces outside the service box), the player can attempt a second serve.
- If the player makes a second fault, he loses the point.

Ed wants to study where Frank's serve bounce in the service box to plan his positioning on the court. We have $N_F = 1,000$ examples of first serve from Frank, and $N_S = 1,000$ examples of second serve. We want to estimate the distributions of the bounce location $\mathbf{x}$ for Frank's first $p(\mathbf{x}|first)$ and second serves $p(\mathbf{x}|second)$.

For simplicity,

- we denote the corner closer to the net and towards the centre of the court as position (0,0), and the corner towards the outside of the court and away from the net as (1,1).
- We will ignore serves that hit the net

This means that values outside $[0,1] \times [0,1]$ indicate that the serve is a fault.

(a) How would you use the *empirical distribution* to get an estimate of $p(\mathbf{x}|first)$? Explain the steps, the parameters that need to be set and the associated trade-offs.

[4]

Solution:

(This was seen in lectures, but students need to apply it to this problem)

The empirical distribution is an approximation of a distribution with a normalised histogram. **[0.5]**

1. divide the serve box horizontally and vertically in B bins, creating a total of $B \times B$ bins. **[0.5]**
2. record the number of serves in our dataset that bounced in each bin **[0.5]**
3. normalise the histogram by the total number of serves in the dataset **[0.5]**

The parameters to consider:

- The number of bins B **[0.5]** if set too small the estimate will be imprecise, if set too large the estimate will be noisy due to the limited number of examples in the dataset. **[0.5]**
- We need to handle serves outside the serve box (faults). **[1]**

(b) Ed now wants to model Frank's serves using a *normal distribution*:

$$f_X(x) = \frac{1}{Z} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

(i) Explain the parameters, their effect on the distribution and the best way to estimate them in this scenario.

[4]

Solution:
The parameters of the normal distribution are:

- the mean vector μ **[0.5]** the mean of the distribution **[0.5]**
- the covariance matrix Σ **[0.5]** determines the shape of the distribution, how wide, narrow or skewed it is **[0.5]**

For the normal distribution, we have *standard estimators*, closed forms estimates of the distribution parameters from the available samples. **[1]**

$$\mu = \frac{1}{N_F} \sum_i \mathbf{x_i}$$

[0.5]

$$\Sigma_{ij} = \frac{1}{N-1} \sum_{k=1}^{N} (X_{ki} - \mu_i)(X_{kj} - \mu_j)$$

[0.5]
(Note: This comes from the linear algebra lectures, this is all in the lectures but requires that the student tie together the 1D example in the probabilities lecture to the linear algebra lecture)

(ii) What could be the problem with this choice of model? Give an example of a situation where it would be inappropriate (you can use a diagram to illustrate your example).

[2]

Solution:
The main problem is if the data is *not* normally distributed—typically if it has more than one mode. For example if Frank serves half of his serves in the bottom left corner of the box and the other half in the bottom right corner of the box, then the normal distribution would estimate a mean in the centre bottom of the box, with a large variance. This is misleading as Frank never actually serves at this location, whereas the model would deem this as a high probability.
[1] for pointing the multiple mode issue. **[1]** for a valid example or diagram.

(c) Ed has found that his normal model is not accurate enough for him. In order to get a more accurate modelling of the data, he decides to use a *mixture of Gaussians* to model his data.

Explain how the model would be parameterised, and how you would fit the model to the available data (provide the relevant equations).

[5]

Solution:

A mixture model is defined as follows (from the lecture motes)

$$f_X(x) = \sum_i \lambda_i n_X(x; \mu_i, \Sigma_i)$$

Where n_X is the standard multivariate Gaussian. **[1]**

In this model the parameters are:

- The number of Gaussians in the model N **[0.5]**
- Then for each Gaussian:
- λ is a scalar, the relative weight of this Gaussian in the mixture **[0.5]**
- μ is the mean vector for this Gaussian **[0.5]**
- Σ is the covariance of this Gaussian **[0.5]**

N is fixed, so we want to fit the parameter vector $\theta = [\lambda_1, \mu_1, \Sigma_1, \ldots, \lambda_N, \mu_N, \Sigma_N]$

For a mixture of Gaussian we do not have standard estimators like for the normal distribution, so we need to use the log likelihood. **[1]**

$$\arg\min_\theta \sum_{i=1}^{N_F} \log \sum_{j=1}^{N} \lambda_j n_X(x_i; \mu_j, \Sigma_j)$$

[0.5]

This could be optimised using for example gradient descent or variants of the Newton method **[0.5]**

3. Pretend that you are the new head of a local radio, *IDSS Radio* being tasked with renewing the radio's image and programme. The radio's programming and popularity has varied over the years and you want to use a data science approach to find the right type of programming for the local audience. To this end you start by categorising the programming of the radio between types of content:

$$\mathscr{C} = \{music, news, business, fiction, comedy, advertisement\}$$

You have historical records of the proportion of each content type in the radio programme for every month over the last ten years, as well as a rating r by a sample of the audience on scale between 1 and 10, where 1 means "hate it" and 10 means "love it".

Considering a programme $\mathbf{p} = [p_m, p_n, p_b, p_f, p_c, p_a] \in \mathbb{R}^5$ that gives the number of hours for each content type, we are interested in studying the function $r(\mathbf{p})$ that gives the listeners' rating for this programme.

(a) As a first attempt, you decide to assume that the function $r(\mathbf{p})$ is linear, and therefore to solve it using linear-least-squares, of the canonical form (from the lecture notes):

$$\arg\min_{\mathbf{x}} L(\mathbf{x}) = \|A\mathbf{x} - y\|_2^2$$

(i) Explain what each variable in this equation means in this scenario, specifying their dimension, and what would be the result.

[4]

Solution:
In this case - $A = [p_m, p_n, p_b, p_f, p_c, P_a, 1]$ is the number of hours for each content type, a matrix of dimension $120 \times 6\$$ **[0.5]** where the first 5 columns correspond each to one type of content **[0.5]** the last one is set to 1 to allow for the constant in the linear relation. **[0.5]** Each row correspond to one month of historical data. **[0.5]** - $\mathbf{y}$ is a vector of dimension 120, recording the average rating for the radio station for each month on record. **[0.5]** - $\mathbf{x}$ is a matrix of parameters representing the linear relation between each content and the rating, a vector of dimension 1×6 (1×5 will also be accepted as already penalised above) **[0.5]** - The outcome will be the coefficient for the linear relation $\hat{\mathbf{x}}$ **[0.5]** that best fits the recorded data **[0.5]** such that

$$\hat{r}(\mathbf{p}) \triangleq [\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4, \hat{x}_5] \cdot \mathbf{p} + \hat{x}_6$$

(The last equation is not needed to get the marks).

(ii) Could you name a reason why this may not be a good model? How could you measure this using your data?

[3]

Solution:
The main reason for this model to fail is if *the relation is not linear* - which is likely to be the case. **[1]** You test this by looking at the *residuals*: the difference between the

ratings predicted by the model and actually recorded in the data **[1]** If the model is good they should be very small **[0.5]** whereas large values would indicate an example poorly modelled **[0.5]**

(b) We want to try and fit another model, this time assuming that viewers' preferences peak for certain quantities of each program, and then decreases again if the quantity increases even more. We could model this quantity preference as a bell shaped distribution function over the quantity p_z for each type of content z:

$$B_z(p_z) = \alpha_z \exp(-\beta \ \|p_z - \mu_z\|^2)$$

and the overall predicetd preference for a program $\mathbf{p}$ as:

$$\hat{r}(\mathbf{p}) = b + \sum_z B_z(p_z)$$

(i) How many parameters do you need to estimate in this case? Explain the role of each parameter.

[3]

Solution:
There is a total of $1 + 5 \times 3 = 16$ parameters: **[0.5]** - b is a constant parameter, the base rating when all values are far from the audience's preference. **[0.5]** - For each type of content z (5 types), we have three more parameters: **[0.5]** - μ_z is the quantity of content c that would lead to the strongest effect in terms of ratings. **[0.5]** - β_z is a scaling parameter influencing how broad or narrow is the bell curve, in other words how much a small departure from μ_c will impact the ratings. **[0.5]** - a_z is the relative importance of the type of content to the rating, similar to the linear case. **[0.5]**

(ii) What would be the most appropriate approach to fit this model to your data (Note: all of the functions above are differentiable, but B_c is clearly not linear)?
Explain how you would parametrise this problem (you are not asked to solve it!)

[3]

Solution:
Because the model is differentiable, gradient descent is likely to be the most efficient optimisation in this case **[1]**
The set of parameters is: $\theta = [b, a_m, \mu_m, \beta_m, \ldots, a_a, \mu_a, \beta_a]$ **[1]**
and the problem can be stated as:

$$\arg\min_\theta \sum_p \|\hat{r}(p) - r(p)\|_2^2$$

[1]

(c) Using this model $\hat{r}$, how would you use optimisation to find the best program, knowing that you want to run the radio from 6am to midnight daily, and need at least 1 hour of

advertisement per day to cover the radio running costs. How would you resolve this optimisation?

[2]

Solution:

The statement of the problem indicates that you need to use constrained optimisation **[0.5]**

$$\begin{aligned} \arg\max_{p} \quad & \hat{r}(p) \\ s.t. \quad & \sum \mathbf{p} - 18 = 0 \\ & 1 - \sum p_a \leq 0 \end{aligned}$$

[0.5] for the maximisation part. **[1]** for the constraints (0.5 each).

4. **(a)** Consider a relation Weather(Id, Time, Longitude, Latitude, Temperature, Humidity), where the primary key (Id) is a 116-byte string hash code, Time is 8-byte Datetime, the other fields are stored by 32-bit float. Assume that the relation has 30000 tuples, stored in a file on disk organised in 4096-byte blocks. Note that the database system adopts fixed-length records – i.e., each file record corresponds to one tuple of the relation and vice versa.

(i) Compute the blocking factor and the number of blocks required to store this relation. [2]

Solution:
(Bookwork for knowledge of the concepts, Problem solving otherwise)
Each record (tuple) has size: $116+8+4+4+4+4 = 140$ bytes. Thus, the blocking factor will be: $\lfloor \text{ block size } \rfloor = \lfloor \frac{4096}{\text{record size}} \rfloor = 29$ records per block. The file will then contain:

$$\left\lceil \frac{\text{num tuples}}{\text{blocking factor}} \right\rceil = \left\lceil \frac{30000}{29} \right\rceil = 1035 \text{ blocks.}$$

[1] for the computation of the blocking factor, **[1]** for the number of file blocks.

(ii) You are told that you will need to frequently add new records and you will not often read and fetch a record. Describe in detail the file organisation that you would expect to exhibit the best performance characteristics. Explain your answer by comparing the cost of reasonable alternatives.

[3]

Solution:
(Bookwork for knowledge of the concepts, Problem solving otherwise)
The relation can be stored in a number of ways: heap file, sequential file (sorted on one or more attributes), hash file (hashed on one or more attributes). Of these, heap files are expected to have the best performance in this relation since inserting a new record is efficient, for a cost of $O(1)$. **[1]**
With sequential file, all the records are ordered by an ordering field, e.g., name, and are kept sorted at all times, which suitable for queries requiring sequential processing. **[1]**
With hash file, reading and fetching a record is faster compared to other methods as the hash key is used to quickly read and retrieve the data from database. **[1]**

(b) Consider the following three relations:

- Student(Id, FirstName, LastName, DateOfBirth) where
  - the primary key (Id) is a 32-bit integer,
  - FirstName and LastName are both 96-byte strings, and
  - DateOfBirth is a 32-bit Integer.
- Course(Id, Description, Credits), or *C*, where:
  - Id, the primary key of this relation, is a 32-bit integer,
  - Description is a 195-byte string, and

  - Credits is an 8-bit integer.
- Transcript(StudentId, CourseId, Mark), or T , where:
  - StudentId is a foreign key to the primary key (Id) in the Student relation,
  - CourseId is a foreign key to the primary key (Id) in the Course relation above
  - Mark is a 8-byte double precision floating number, and
  - the primary key consists of the combination of StudentID and CourseID.

Assume these relations are also organised in 4096-byte blocks, and that:

- Relation Course (C) has r_C = 32 records and n_C = 2 blocks, organised in a heap file,
- Relation Transcript (T) has r_T = 51200 records and n_T = 200 blocks, organised in a sequential file, ordered by StudentID.
- Relation Student (S) has r_S = 2000 records and n_S =100 blocks, stored in a heap file and has a 4-level secondary index on *StudentId*.

Further assume that the memory of the database system can accommodate $n_B = 23$ blocks for processing and that the blocking factor for the join-results block is $bfr_{RS} = 10$ records per block.

Last, assume we execute the following equi-join query:

```
SELECT * FROM Transcript AS T, Student AS S, Course AS C
  WHERE T.StudentId = S.Id AND T.CourseId = Course.Id
```

As this is a 3-way join, assume that you need to join T with C first, with each block of intermediate results stored only in RAM (in one of the n_B blocks), then joined with S.

(i) Describe the join strategy that would be the most efficient in this case and estimate its total expected cost (in number of block accesses). Show your work.

[8]

Solution:
Join T with C first, then their result with S.

- First join-Intermediate $(I) = T \bowtie C)$:

$$js_{I=T\bowtie C} = 1/\max(NDV(CourseId, Transcript), NDV(Id, Course)) = 1/\max(32, 32) = 1/32$$

. **[1]**

$$jc_{T\bowtie C} = js_{T\bowtie C} \times |T| \times |C| = 1/32 \times 51200 \times 32 = 51200$$

. **[1]**

- This join can also be computed in two ways:
  (a) Scan C at the outer loop, do binary search on T for each value $(\text{ceil}(\log_2(n_T)) = 8)$:

With 51200 records for 2000 students, each StudentID will appear in 26 records in T on average;

with 16 bytes per record in T and 4096-byte blocks, all 26 records should be in one block (on average). The cost of this join is then: $n_C + r_C \times (\text{ceil}(\log_2(n_T))) = 2 + 32 \times 8 = 258$ block accesses. **[1]**

(b) Scan T at the outer loop, full scan on C for each value

(i.e., Naive Nested Loop Join): The cost of this join is then:

$n_T + n_C \times \lceil \frac{n_T}{n_B - 2} \rceil = 200 + 2 \times 10 = 220$ block accesses. **[1]**

Of the above strategies, the second is the most efficient, with a total cost of 220 block accesses.

Second join $(I \bowtie S)$:

- This join can also be computed in two ways:

  (a) We can store the intermediate join results of each block in memory, then for each record we use $S.Id$'S 4 -level index.

$js_{I \bowtie S} = 1/\max(NDV(\text{ StudentId,I}), NDV(Id, \text{ Student })) = 1/\max(2000, 2000) = 1/2000$

. **[1]**

$$jc_{I \bowtie S} = js_{I \bowtie S} \times |I| \times |S| = 1/2000 \times 51200 \times 2000 = 51200$$

. **[1]**

There is only one strategy at this point:

For each record produced in I , search for the matching record in S.

S has a 4 -level index, hence it will take $4 + 1 = 5$ block accesses to locate each value. The cost of this join is then: $r_I \times 5 + \frac{jc_{I \bowtie C}}{bfr_{RS}} = 51200 \times 5 + \frac{51200}{10} = 261120$. **[1]**

The total cost of this plan: $220 + 261120 = 261340$ block accesses **[1]**

(b) We can also write the intermediate join results back to the disk, then use Naive Nested Loop Join:
The cost of writing back to $n_{CT} = \frac{jc_{T \bowtie C}}{10} = \frac{51200}{10} = 5120$ [1 mark]
The cost of this join is then:
$n_S + n_{CT} \times \lceil \frac{n_S}{n_B - 2} \rceil = 100 + 5120 \times \lceil \frac{100}{23 - 2} \rceil = 25700$ block accesses. [1 mark]
The total cost of this plan: $220 + 5120 + 25700 + 5120 = 36160$ block accesses. And this is the best solution. [2 mark]
Marking notes:

- 0.5 mark for each correct formula for join T with C, e.g. $n_C + r_C \times \log_2(n_T)$ or $n_T + n_C \times \lceil \frac{n_T}{n_B - 2} \rceil$.
- 0.5 mark for each correct formula for join S, e.g. $n_S + n_{CT} \times \lceil \frac{n_S}{n_B - 2} \rceil$ or $r_I \times (level + 1)$.
- 0.5 mark for each join selectivity formula or join cardinality.
- For the Second join, either of two ways will obtain a full 4 marks.

(ii) Compare the Naive Nested Loop Join and the Index-based Nested-Loop Join. Which one is faster? Explain why.

[2]

Solution:
The Nested Loop Join searches for a row in the entire inner side of the table / index. The Indexed Nested Loop Join searches for a row in the inner side of the index and seeks the index's B-tree for the searched value(s) and then stops looking further. **[1]**
The Index-based Nested-Loop Join is much faster than the Naive Nested Loop Join, since for the outer relation, he Naive Nested Loop Join need scan the entire inner relation while with Index-based Nested-Loop Join, there is an index on the join column of one relation can make it the inner and exploit the index, avoiding linear search and hence it is faster. **[1]**

Mark table

Question	Points	Score
1	15	
2	15	
3	15	
4	15	
Total:	60	

University of Glasgow

TBD
09:30 BST
Duration: 2 hours
Additional Time: 30 minutes
Timed exam — fixed start time

DEGREES of MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

Introduction to Data Science and Systems ¶ COMPSCI 5089

(Answer All 4 Questions)

This examination paper is an open book, online assessment and is worth a total of 60 marks.

1. **(a)** You are designing an application for clothing shops to predict clothes size based on customer height and weight. Suppose we have a **clothing dataset** with height, weight and the corresponding T-shirt size of several customers.

customer ID	height	weight	size
U1	170	60	M
U2	172	60	M
U3	173	61	M
U4	173	64	L
U5	175	67	L
U6	175	66	L

You can represent this dataset based on their vector representations by regarding height and weight as two dimensions. Now there is a new client **Abel** (U0) whose height is 173cm and weight is 62kg. You are asked to predict the T-shirt size for Abel.

(i) Calculate the Euclidean distance (L2 Norm) between the new point and the existing points.

[3]

(ii) Predict the size of Abel, based on the kNN algorithm, with $k = 3$ and the above calculated distances. Justify your prediction.

[2]

(b) *For all answers, include in your answer document both code and the output of that code.*

(i) Calculate the covariance matrix for the clothing dataset using numpy.

[1]

(ii) Calculate the eigenvector and eigenvalues the covariance matrix using numpy.

[2]

(iii) Dimensionality reduction. Map the **clothing dataset** into principal component with the largest eigenvalue of its covariance matrix.

[2]

(c) (i) Find SVD for $A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$, you should include full working in your solution.

[3]

(ii) State the relations between determinant, matrix inversion and non-singular.

[2]

2. Consider a tennis player, Ed Balls, who wants to prepare for a competition match against an opponent—let's call him Frank Racket. In order to prepare for the match, Ed has acquired records of the 100 previous matches of his opponent and wants to study statistics of Frank's play to choose where to focus his training.

(Here is a quick summary of the rules of tennis: `https://protennistips.net/tennis-rules/`)

Ed is interested in studying Frank's serve as this can be an important strategic advantage.

- For a serve to be valid, it must pass the net and bounce in the diagonally opposite service box.
- If the first serve is a fault (eg, hits the net or bounces outside the service box), the player can attempt a second serve.
- If the player makes a second fault, he loses the point.

Ed wants to study where Frank's serve bounce in the service box to plan his positioning on the court. We have $N_F = 1,000$ examples of first serve from Frank, and $N_S = 1,000$ examples of second serve. We want to estimate the distributions of the bounce location $\mathbf{x}$ for Frank's first $p(\mathbf{x}|first)$ and second serves $p(\mathbf{x}|second)$.

For simplicity,

- we denote the corner closer to the net and towards the centre of the court as position (0,0), and the corner towards the outside of the court and away from the net as (1,1).
- We will ignore serves that hit the net

This means that values outside $[0,1] \times [0,1]$ indicate that the serve is a fault.

(a) How would you use the *empirical distribution* to get an estimate of $p(\mathbf{x}|first)$? Explain the steps, the parameters that need to be set and the associated trade-offs.

[4]

(b) Ed now wants to model Frank's serves using a *normal distribution*:

$$f_X(x) = \frac{1}{Z} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

(i) Explain the parameters, their effect on the distribution and the best way to estimate them in this scenario.

[4]

(ii) What could be the problem with this choice of model? Give an example of a situation where it would be inappropriate (you can use a diagram to illustrate your example).

[2]

(c) Ed has found that his normal model is not accurate enough for him. In order to get a more accurate modelling of the data, he decides to use a *mixture of Gaussians* to model his data.

Explain how the model would be parameterised, and how you would fit the model to the available data (provide the relevant equations).

[5]

3. Pretend that you are the new head of a local radio, *IDSS Radio* being tasked with renewing the radio's image and programme. The radio's programming and popularity has varied over the years and you want to use a data science approach to find the right type of programming for the local audience. To this end you start by categorising the programming of the radio between types of content:

$$\mathscr{C} = \{music, news, business, fiction, comedy, advertisement\}$$

You have historical records of the proportion of each content type in the radio programme for every month over the last ten years, as well as a rating r by a sample of the audience on scale between 1 and 10, where 1 means "hate it" and 10 means "love it".

Considering a programme $\mathbf{p} = [p_m, p_n, p_b, p_f, p_c, p_a] \in \mathbb{R}^5$ that gives the number of hours for each content type, we are interested in studying the function $r(\mathbf{p})$ that gives the listeners' rating for this programme.

(a) As a first attempt, you decide to assume that the function $r(\mathbf{p})$ is linear, and therefore to solve it using linear-least-squares, of the canonical form (from the lecture notes):

$$\arg\min_{\mathbf{x}} L(\mathbf{x}) = \|A\mathbf{x} - y\|_2^2$$

(i) Explain what each variable in this equation means in this scenario, specifying their dimension, and what would be the result.

[4]

(ii) Could you name a reason why this may not be a good model? How could you measure this using your data?

[3]

(b) We want to try and fit another model, this time assuming that viewers' preferences peak for certain quantities of each program, and then decreases again if the quantity increases even more. We could model this quantity preference as a bell shaped distribution function over the quantity p_z for each type of content z:

$$B_z(p_z) = \alpha_z \exp(-\beta \ \|p_z - \mu_z\|^2)$$

and the overall predicetd preference for a program $\mathbf{p}$ as:

$$\hat{r}(\mathbf{p}) = b + \sum_z B_z(p_z)$$

(i) How many parameters do you need to estimate in this case? Explain the role of each parameter.

[3]

(ii) What would be the most appropriate approach to fit this model to your data (Note: all of the functions above are differentiable, but B_c is clearly not linear)?
Explain how you would parametrise this problem (you are not asked to solve it!)

[3]

(c) Using this model $\hat{r}$, how would you use optimisation to find the best program, knowing that you want to run the radio from 6am to midnight daily, and need at least 1 hour of advertisement per day to cover the radio running costs. How would you resolve this optimisation?

[2]

4. **(a)** Consider a relation Weather(Id, Time, Longitude, Latitude, Temperature, Humidity), where the primary key (Id) is a 116-byte string hash code, Time is 8-byte Datetime, the other fields are stored by 32-bit float. Assume that the relation has 30000 tuples, stored in a file on disk organised in 4096-byte blocks. Note that the database system adopts fixed-length records – i.e., each file record corresponds to one tuple of the relation and vice versa.

(i) Compute the blocking factor and the number of blocks required to store this relation.

[2]

(ii) You are told that you will need to frequently add new records and you will not often read and fetch a record. Describe in detail the file organisation that you would expect to exhibit the best performance characteristics. Explain your answer by comparing the cost of reasonable alternatives.

[3]

(b) Consider the following three relations:

- Student(Id, FirstName, LastName, DateOfBirth) where
  - the primary key (Id) is a 32-bit integer,
  - FirstName and LastName are both 96-byte strings, and
  - DateOfBirth is a 32-bit Integer.
- Course(Id, Description, Credits), or C, where:
  - Id, the primary key of this relation, is a 32-bit integer,
  - Description is a 195-byte string, and
  - Credits is an 8-bit integer.
- Transcript(StudentId, CourseId, Mark), or T , where:
  - StudentId is a foreign key to the primary key (Id) in the Student relation,
  - CourseId is a foreign key to the primary key (Id) in the Course relation above
  - Mark is a 8-byte double precision floating number, and
  - the primary key consists of the combination of StudentID and CourseID.

Assume these relations are also organised in 4096-byte blocks, and that:

- Relation Course (C) has $r_C = 32$ records and $n_C = 2$ blocks, organised in a heap file,
- Relation Transcript (T) has $r_T = 51200$ records and $n_T = 200$ blocks, organised in a sequential file, ordered by StudentID.
- Relation Student (S) has $r_S = 2000$ records and $n_S = 100$ blocks, stored in a heap file and has a 4-level secondary index on *StudentId*.

Further assume that the memory of the database system can accommodate $n_B = 23$ blocks for processing and that the blocking factor for the join-results block is $bfr_{RS} = 10$ records per block.

Last, assume we execute the following equi-join query:

```
SELECT * FROM Transcript AS T, Student AS S, Course AS C
  WHERE T.StudentId = S.Id AND T.CourseId = Course.Id
```

As this is a 3-way join, assume that you need to join T with C first, with each block of intermediate results stored only in RAM (in one of the n_B blocks), then joined with S.

(i) Describe the join strategy that would be the most efficient in this case and estimate its total expected cost (in number of block accesses). Show your work.

[8]

(ii) Compare the Naive Nested Loop Join and the Index-based Nested-Loop Join. Which one is faster? Explain why.

[2]

University of Glasgow

Wednesday 15th of December 2021
9:00 am — 11:30am
Duration: 2 hours
Additional time: 30 minutes
Timed exam — fixed start time

DEGREE OF MSc

INTRODUCTION TO DATA SCIENCE AND SYSTEMS (M)
COMPSCI5089

Answer all 3 questions

This examination paper is worth a total of 60 marks.

1. Computational linear algebra and optimisation

(a) Given a collection of N documents $\mathscr{D} = \{D_1, \ldots, D_N\}$, your task is to implement a functionality that provides a list of suggested 'more like this' documents. With this problem context, answer the following questions.

(i) Explain how would you represent each document $D \in \mathscr{D}$ as a (real-valued) vector $\mathbf{d}$. What is the dimension of each vector? [2]

(ii) What does the L_0 norm of a document vector indicate (in plain English) as per your definition of the document vectors in the previous question? [1]

(iii) How would you define the L_p distance between two document vectors $\mathbf{d}$ and $\mathbf{d}'$? [2]

(iv) What distance or similarity measure would you use for finding the set of 'more like this' documents for a current (given) document vector $\mathbf{d}$, and why. [2]

(b) The probability distribution function of an n dimensional Gaussian is given by

$$f(\mathbf{x}) = (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu),$$

where $\mu \in \mathbb{R}^n$ is the mean vector and $\Sigma \in \mathbb{R}^{n \times n}$ is a square and invertible matrix, called the *covariance* matrix. Consider the particular case of $n = 2$. Answer the following questions.

(i) Plot the contours of the following Gaussians. For each contour plot, show the conditional distributions along the two axes.

$$\mu = (0,0)$$
$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\mu = (0,0)$$
$$\Sigma = \begin{pmatrix} 0.5 & 0 \\ 0 & 2 \end{pmatrix}$$

$$\mu = (0,0)$$
$$\Sigma = \begin{pmatrix} 0.5 & 0.1 \\ 0.5 & 2 \end{pmatrix}$$

$$\mu = (0,0)$$
$$\Sigma = \begin{pmatrix} 0.5 & 0.1 \\ -0.5 & 2 \end{pmatrix}$$

[2]

(ii) Which one/ones of the above 4 Gaussian distributions can be reduced to a a single dimensional Gaussian with PCA on the covariance matrix without too much loss of information. Note that you do not need to explicitly compute the Eigenvalues. You should rather derive your answer from a visual interpretation of the contour plots. Clearly explain your answer. [2]

(c) With respect to linear regression, answer the following questions.

(i) Derive the expression for stochastic gradient descent for linear regression with the squared loss function. Clearly introduce your notations for the input/output instances, and the parameter vector. [4]

(ii) Explain how linear regression can be extended to polynomial (higher order) regression? What is the problem of using high degree polynomials for regression? How can that problem be alleviated? [3]

(iii) A common practice in stochastic gradient descent is to use a variable learning rate α for the parameter updates

$$\theta_j^{(t+1)} \leftarrow \theta_j^{(t)} + \alpha^{(t)} \frac{\partial L}{\partial \theta_j},$$

where $\theta_j^{(t)}$ denotes the j^{th} component of the parameter vector θ at iteration t, and $\alpha^{(t)}$ denotes the value of the learning rate at iteration t. Which of the following alternatives of the learning rate update would you prefer (α is a constant) and why?

$$a)\, \alpha^{(t)} = \frac{\alpha}{t}, \quad b)\, \alpha^{(t)} = \alpha + t.$$

[2]

2. Probabilities & Bayes rule

Consider a card game where you have 4 suits (heart, diamonds, clubs and spades) and in each suit the cards 7, 8, 9, 10, Jack (J), Queen (Q), King (K) and Ace (A). In this question we will use the following commonly used terms:

- the *the pack*: is the set of all cards that have not been drawn yet.
- to *draw*: is to pick a card at random amongst the pack of remaining cards, removing it from the pack.
- the *hand*: is the set of cards a player has drawn from the pack.
- a *payout*: is the amount of points you get for a given hand.
- to *fold*: is to stop playing and put back your cards in the pack, forfeiting any payout for this game.

(a) Assuming that you draw a single card at random from the pack, give the probabilities for the following events

(i) Drawing an Ace?

(ii) Drawing a red card?

(iii) Drawing a diamonds?

(iv) Drawing a royalty figure (Jack, Queen or King)?

(v) Drawing the Ace of spades?

[5]

(b) Now assume that you have already drawn the three cards: 10,J,Q. When drawing two more cards from the pack, what is the probability to obtain:

(i) A pair of two cards with the same value (eg, two Jacks).

(ii) Two pairs (eg, two Jacks and two Queens).

(iii) Three of a kind (eg, three Jacks).

(iv) A sequence of 5 cards (eg, 10, J, Q, K, A). Note that the cards can be of any suit, but there cannot be a break in the sequence.

[4]

(c) Now let us assume the following payout table for each hand of 5 cards:

hand	payout
sequence of 5 cards	50
three of a kind	30
two pairs	20
one pair	10
anything else	0

As before, you have the cards 10, J, Q in hand.

(i) If you draw two more cards randomly from the deck, what is the expected value of the payout for this hand? [3]

(ii) Assuming that you need to pay 5 every time you draw a card (hence you would need to pay 10 to draw two cards), should you fold your hand or draw cards? [2]

(iii) Should you fold after drawing the first card (and having paid 5), if the card is: (i) the 7 of heart, (ii) the 8 of spades or (iii) the Queen of diamonds? [6]

3. Database systems

An online retail company is trying to assess the performance of its DB systems and has asked you to investigate some of the operations. Consider a relation Seller (ID, Name, Country) – abbreviated as S – where the primary key (ID) is a 32-bit integer, the Name attribute is a 54-byte (fixed length) string, and Country is a 16-bit integer. Further consider a relation Product(ID, ProductID, ManufacturerID, Price) – abbreviated as P – with ID being a foreign key to Seller's ID, ProductID and ManufacturerID being 64-bit integers, Price being a 32-bit float, and the first three attributes making up the relation's (composite) primary key.

Assume that both relations are stored in files on disk organised in 512-byte blocks, with each block having a 10-byte header. Assume that S has $r_S = 1,000$ tuples and that P has $r_P = 100,000$ tuples. Last, assume that Product is stored organised in a sequential file sorted by its primary key and Student is stored organised in a heap file. Finally, note that the database system adopts fixed-length records – i.e., each file record corresponds to one tuple of the relation and vice versa.

(a) Compute the blocking factors and the number of blocks required to store these relations. Show your work.

[2]

(b) Consider the following query:

```
SELECT S.Name, P.ID, P.Price FROM Seller as S, Product as P
WHERE S.ID = P.ID AND S.ID >= 6,000 and S.ID <= 6,199;
```

Assume that the memory of the database system can accommodate $n_B = 22$ blocks for processing, that all seller IDs in the query range exist in the database, and that all sellers have the same number of products on average.

Explain the query processing algorithm, taking care to include the file organisation in your reasoning. Then estimate the total expected cost (in number of block accesses) of the query above (Disregard the cost associated with the writing of the result set), of the following two approaches:

(i) First, assume that S is scanned at the outer loop and show your work: [9]

(ii) Second, assume instead that P is scanned at the outer loop and show your work: [9]

University of Glasgow

Monday 26 April 2021
Available from 14:00 BST
Expected Duration: 2 hours
Time Allowed: 4 hours
Timed exam within 24 hours

DEGREE of MSc

INTRODUCTION TO DATA SCIENCE AND SYSTEMS (M)

Answer all 3 questions

This examination paper is an open book, online assessment and is worth a total of 60 marks.

1. Computational linear algebra and optimisation

You have been asked to help design the subcomponents of a music streaming service. The service has access to 101,750 music tracks (i.e. the audio files). Each music track can be summarised based on the audio content using so-called audio features resulting in a 15 dimensional vector, $\mathbf{x} \in \mathbb{R}^{1x15}$, for each track. The meaning and importance of the individual dimensions in the vector is unknown. The vectors for the individual tracks are collected in a matrix X as row vectors.

Aside from the audio file itself, the service has access to the title and artist for each track, the genre(s) associated with each track (e.g. jazz) and finally the popularity of each track as a scalar $y \in \mathbb{R}$.

(a) The team wants to develop a function called "What is this track called?" where users can upload an audio file with the purpose to identify the name of the track and artist. To this end we are interested in computing Euclidian distances between the music tracks based on their vector representations.

(i) Certain aspects of X is summarised in Table 1. Explain why it is a good idea to normalise the data in X before computing the similarity between the tracks and suggest a suitable normalisation approach. Justify your approach and make reference to specific elements in Table 1. [3]

(ii) Design a simple search routine which can find the closest match between the uploaded track and a track in the existing dataset. Write the procedure using equations or NumPy code (1-3 lines). Determine how many individual distances you will need to compute and discuss any potential scaleability issues. [3]

(b) A subcomponent of the system relies on a mapping from tracks to popularity. This can be formulated as a matrix problem: $X\mathbf{w}^T - \mathbf{y} = \mathbf{0}$, where X is a matrix containing the music features for the tracks. $\mathbf{w}$ is a 15 dimensional vector and $\mathbf{y}$ is vector containing the popularity scores for each track. The team is interested in the most efficient and robust method for finding $\mathbf{w}$ using the squared error as the loss function.

(i) Specify the dimensions of the matrix X and determine if $\mathbf{w}$ and $\mathbf{y}$ are considered row or column vectors, respectively. [1]

(ii) Determine a method for solving the matrix equation wrt. $\mathbf{w}$. Justify your approach. [2]

Dim.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
μ	0.1	-1.5	78.1	0.1	1.1	0.8	0.0	-159.3	0.2	0.0	0.0	0.1	0.0	-0.5	0.3
σ^2	0.01	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	10.1	1.0	1.0	1.0
Min	0.09	-4.0	75.1	-2.4	-1.9	-1.5	-2.9	-162.3	-3.7	-2.5	-2.4	-24.7	-2.6	-3.6	-2.6
Max	0.12	1.6	80.5	3.0	3.8	3.2	2.3	-156.9	2.5	2.6	2.3	31.0	2.0	2.1	2.5

Table 1: Basic statistics for each dimension in X.

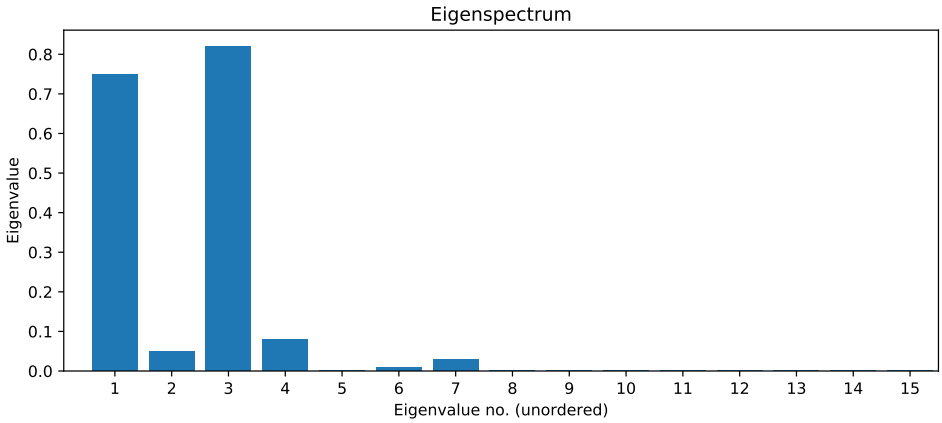

Figure 1: Eigenspectrum (unordered)

(c) The user interface team has requested that you provide a procedure for projecting the music tracks to 2D or 3D based on the vector representation so they can visualise the music tracks on a computer screen. You must use a linear map due to computational constraints.

(i) Outline a procedure for finding the 3D coordinates so the projection preserves most of the variance and can be implemented using only basic Python and NumPy by a junior data scientist. You should not provide the code, but explain the individual steps in the procedure using text or equations and only recommend the suitable NumPy commands. You must specify the dimension of the all vectors or matrices required to compute the projection. [4]

(ii) The eigenspectrum of the covariance matrix of X is shown in Figure 1. Discuss what the eigenspectrum says about the vectors representing the audio files and how this could be leveraged to make the system more efficient. Discuss if the team's idea of a 2D or 3D interface is justified. [3]

(d) Your team is contemplating a new subcomponent which would enable users to generate a new music track. The team has already developed a function, $r(\mathbf{x})$, which makes it possible to map from the vector representation, $\mathbf{x}$, to the audio file.

The aim is to create a new track based on a genre profile which is a 5 dimensional vector, $\mathbf{g} \in \mathbb{R}^5$. Your team has provided a **non-linear function**, $f : \mathbf{x} \rightarrow \mathbf{g}$, that maps from the track vector to the 5D genre profile. Provide a solution in the form of an optimisation problem and determine a suitable method to solve the stated problem. Justify your choice of method and explain under which circumstances it is guaranteed to converge to a sensible solution in this scenario. You will need to make assumptions which must be clearly stated. [4]

2. Probabilities & Bayes rule

Consider a scenario where you are in charge of analysing the data and modelling a pandemic. We consider a given disease (let's call it 'VIRUS'), which has an unknown prevalence ρ in the population (we will assume that $\rho \in [0,1]$ is the proportion of the population that has the disease). We will write the probability that a person is diseased as $p(D) = \rho$.

(a) Your lab has developed a fast testing procedure to detect this disease. In order to evaluate the accuracy and reliability of this test, you have conducted trials on 132 subjects, and compared the results of your test with perfectly accurate (supposedly more expensive) diagnostic. The results of those trials are collated in the following table:

	positive	negative
diseased	28	3
healthy	12	89

(i) Using Bayes formula, and the trial data in the table, provide an estimate of the probabilities:
- $p(D|T)$, that a subject who tested positive is truly diseased; and
- $p(D|\bar{T})$, that a subject who tested negative is actually diseased. [4]

(ii) Taking into consideration the test accuracy and reliability as evidenced in the trials, would this test be appropriate for the following situations:
1. regular testing of people working with vulnerable populations;
2. deciding on whether to administer a treatment with severe side effects; or
3. applying to the whole population to find all diseased individuals

(justify your answers). [3]

(iii) You administer a test with probabilities $p(D|T) = 0.7$ and $p(D|\bar{T}) = 0.01$ to a sample of 1000 subjects drawn randomly from the population. The test returns 980 negatives and 20 positives.
From this data, calculate an estimate of the prevalence $p(D)$ explaining your reasoning. [4]

(b) Let us consider that you are experimenting with a vaccine against the disease. You have 1000 subjects in group A who take the vaccine and 1000 in group B who take a placebo. Let us assume that you test the subjects in both groups daily, and after one month you obtain the following results: 2 subjects from group A tested positive at some point during the month, and 40 subjects from group B. In this part we will assume that we are using a test with the following statistics:
- the probability of having the disease if tested positive is $p(D|T) = 0.7$
- the probability of having the disease if tested negative is $p(D|\bar{T}) = 0.01$.

(i) Accounting for the limitations of the test, how many subjects in group A and B did possibly catch the disease during this month? [5]

(ii) The efficacy of a vaccine is typically calculated as

$$E_V = \frac{p(D|\bar{V}) - p(D|V)}{p(D|\bar{V})}.$$

Use your results from above to calculate the efficacy of the vaccine. Discuss what would happen if your test were less accurate: What would happen if $p(D|T)$ would be lower? If $p(D|\bar{T})$ would be higher? [4]

3. Database systems

Consider a relation Student(ID, Name, StudyPlan) – abbreviated as S – where the primary key (ID) is a 64-bit integer, the Name attribute is a 40-byte (fixed length) string, and StudyPlan is a 16-bit integer. Further consider a relation Marks(ID, CourseID, AssessmentID, Mark) – abbreviated as M – with ID being a foreign key to Student's ID, CourseID and AssessmentID being 16-bit integers, Mark being a 64-bit float, and the first three attributes making up the relation's (composite) primary key.

Assume that both relations are stored in files on disk organised in 512-byte blocks, with each block having a 10-byte header. Assume that S has $r_S = 1,000$ tuples and that M has $r_M = 100,000$ tuples. Last, assume that Student is stored organised in a heap file, and Marks is stored organised in a sequential file sorted by its primary key. Note that the database system adopts fixed-length records – i.e., each file record corresponds to one tuple of the relation and vice versa.

(a) Compute the blocking factors and the number of blocks required to store these relations. Show your work.

[2]

(b) Consider the following query:

```
SELECT S.Name, M.ID, M.Mark FROM Student as S, Marks as M
WHERE S.ID = M.ID AND S.ID >= 10,000 and S.ID <= 10,199;
```

Assume that the memory of the database system can accommodate $n_B = 22$ blocks for processing, that all student IDs in the query range exist in the database, and that all students have the same number of marks on average.

Consider the following two approaches: (a) S is scanned at the outer loop (9 marks total) and (b) M is scanned at the outer loop (9 marks total). For each approach, explain the query processing algorithm (3 marks for each approach) taking care to include the file organisation in your reasoning, and estimate the total expected cost (in number of block accesses) for these two strategies (6 marks for each approach). Disregard the cost associated with the writing of the result set. Show your work.

[18]

University of Glasgow

Thursday 19 December 2019
1.00 pm – 2.30 pm
(1 hour 30 minutes)

DEGREE of MSc

INTRODUCTION TO DATA SCIENCE AND SYSTEMS (M)

Answer all 3 questions

This examination paper is worth a total of 60 marks.

The use of calculators is not permitted in this examination.

INSTRUCTIONS TO INVIGILATORS: Please collect all exam question papers and exam answer scripts and retain for school to collect. Candidates must not remove exam question papers.

1. Linear algebra, probability, visualisation and optimisation

Your data science team has been asked to analyse a subsystem for a car manufacturer. After some experimentation it is clear that the system you are considering can be described by the following set of coupled equations:

$$\begin{array}{rcl} -14+x\alpha+z\gamma & = & -y\beta \\ 2x\alpha-yz\beta+8 & = & -x\gamma+x\alpha \\ -z\gamma & = & -5-y\beta \end{array} \tag{1}$$

where $x=1, y=2, z=3$ are scalar inputs to the system and the output of the system is denoted by $\mathbf{c} = f\left([x,y,z]^T, [\alpha,\beta,\gamma]^T\right) = [14,-8,-5]^\top$. $\mathbf{b} = [\alpha,\beta,\gamma]^\top$ is a vector containing the parameters of the system.

(a) Convert the set of coupled equations in Eq. (1) into the matrix form $\mathbf{Ab} = \mathbf{c}$. [3]

(b) You are now asked to find the parameters, **b**, of the system using a numerical optimization method without the availability of standard solvers and matrix inversion.

(i) Define an optimization problem that would allow you to solve a problem of the type $\mathbf{Ab} = \mathbf{c}$ with respect to **b** with the constraint that you cannot use matrix inversion but have access to partial derivatives of **A**, **b** and **c** with respect to **b**. [2]

(ii) State a form of the update equations for standard gradient descent which will allow you to solve the optimization problem outlined in the previous question and explain under which conditions your gradient descent optimization algorithm is guaranteed to converge. [4]

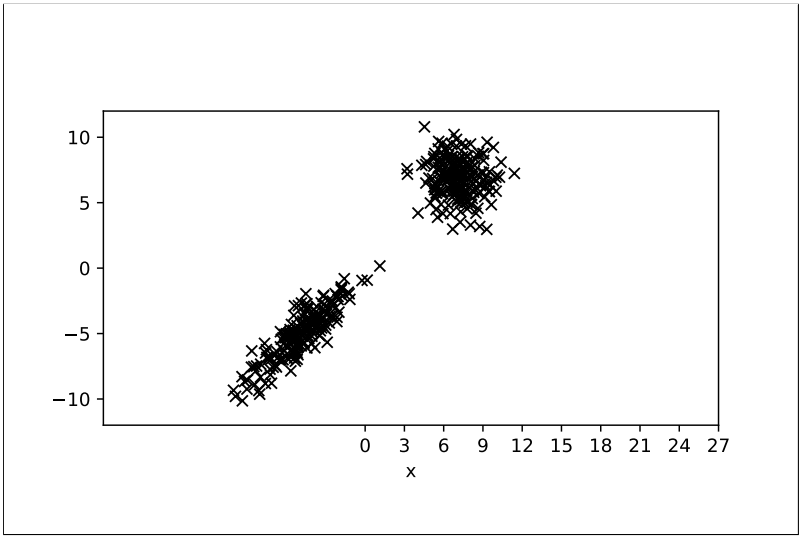

Figure 1: A scatter plot illustrating two datasets. The two different datasets can clearly be identified as two distinct clusters (as validated by the manager)

(c) Your manager has provided you with two datasets obtained on two different days each containing several observation of x and y.

(i) Your manager has illustrated the observations in Figure 1. Criticise this graph, and redraw a sketch that corrects the issues you have identified. [3]

(ii) Your manager asks you to summarise each of the datasets using a separate Normal distribution for each dataset. Explain how you would parameterise the Normal distributions needed to model the (x,y) values from the two individual datasets, including a description of the array shape of any parameters that the distribution would have. [3]

(iii) Explain how eigendecomposition could be used on the parameters estimated in the previous question to identify the major axis of variation. Draw a simple sketch to show: the data points; the estimated normal distributions; the relevant eigenvectors (for each dataset separately). [5]

2. Text processing in data science

(a) (i) Consider two documents with term frequency vectors as follows: D1 = [4, 2, 0] and D2 = [2, 0, 4]. Calculate the cosine similarity between these two documents. Give the formula for cosine similarity and show your workings. Note: the final result may be in the form of a formula. [3]

(ii) Name and describe an application where cosine could be used. Justify why cosine should be used for this application, explain the key geometric properties of cosine similarity and why it is important for the application. [2]

(iii) Your are given the following list of documents in python: `docs = ('The sky is green', 'The sun is yellow', 'We can see the shining sun, the bright sun in the sky')`. Write python code to compute the TF-IDF cosine similarity matrix of the `docs` list using the appropriate Sci-Kit Learn libraries. [3]

(iv) Define the concept of *lemmatization*. Compare and contrast it with *stemming*. [2]

(b) (i) Explain the k-means clustering algorithm using pseudo-code or precise word descriptions. Name and describe three key clustering properties of k-means. [3]

(ii) You work at a large social media company with an advertising network. Describe a task where k-means could be applied and describe how it would be implemented. Provide details including specifying appropriate textual features and their representation, the similarity function, and how to address issues of scale on large datasets. [3]

(iii) The default k-means algorithm runs on the task from part 2.b.(ii) for a very large data collection. The clustering is too slow and takes too long to complete. The product requirements dictate that the number of clusters and features are fixed. Discuss why it is slow and suggest a modification to the k-means algorithm that will speed it up. [2]

(iv) How many clusters would you guess the data illustrated in Figure 2 has? Describe the method you would use to determine a correct value of k. Does it matter if the value is determined over a single run vs many runs of the algorithm? Explain why or why not. [2]

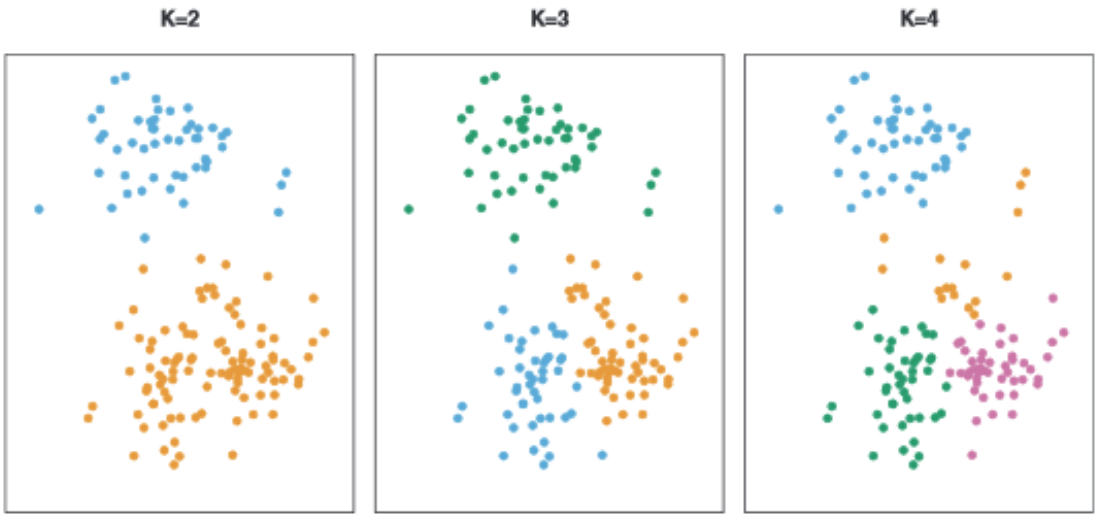

Figure 2: Example cluster data

3. Database systems

(a) Consider a relation Employee(ID, Name, Age) where the primary key (ID) is a 64-bit integer, Age is an 8-bit integer, and that we need 51 bytes for the Name attribute. Assume that the relation has 1000 tuples, stored in a file on disk organised in 512-byte blocks each having a 24-byte header. Note that the database system adopts fixed-length records – i.e., each file record corresponds to one tuple of the relation and vice versa.

(i) Compute the blocking factor and the number of blocks required to store this relation? [2]

(ii) Consider the following SQL query:

```
SELECT Name FROM Employee WHERE ID >= 101 AND ID <= 115
```

Estimate the expected query processing cost in terms of number of block accesses, when the relation file is organised as (a) a heap file, (b) a sequential file, ordered by primary key, and (c) a hash file using external hashing with the primary key as the hash key field, 256 buckets each containing one block, and no overflow buckets. [5]

(b) Consider two relations Employee E and Department D such that:

- Relation E has $n_E = 100$ blocks and $r_E = 1000$ records
- Relation D has $n_D = 50$ blocks and $r_D = 10$ records.

Assume that relation Employee (E) has an attribute SSN (Social Security Number) as its primary key, and that relation Department (D) has a unique attribute Mgr_SSN (Manager's SSN) being a foreign key referencing relation E's SSN attribute. Note: Mgr_SSN is unique and hence there is a single manager per department. Further assume that the memory of the database system can accommodate $n_B = 12$ blocks for processing and the blocking factor for the join-results block is $bfr_{RS} = 10$ records per block. Last, assume we execute the following equi-join query:

```
SELECT * FROM E, D WHERE D.Mgr_SSN = E.SSN
```

(i) Assume that the query is processed using the nested-loop join algorithm. Estimate the total expected cost (in number of block accesses) for the various strategies and conclude which strategy is the most efficient. Show your work. [6]

(ii) Assume that there is a Level 2 Secondary Index over the Department relation for the unique attribute Mgr_SSN, i.e., level $x_D = 2$, and a Level 2 Primary Index over the Employee relation for the primary key SSN, i.e., level $x_E = 2$. Propose two index-based nested-loop join strategies and explain which one is the best using these indexes. [7]