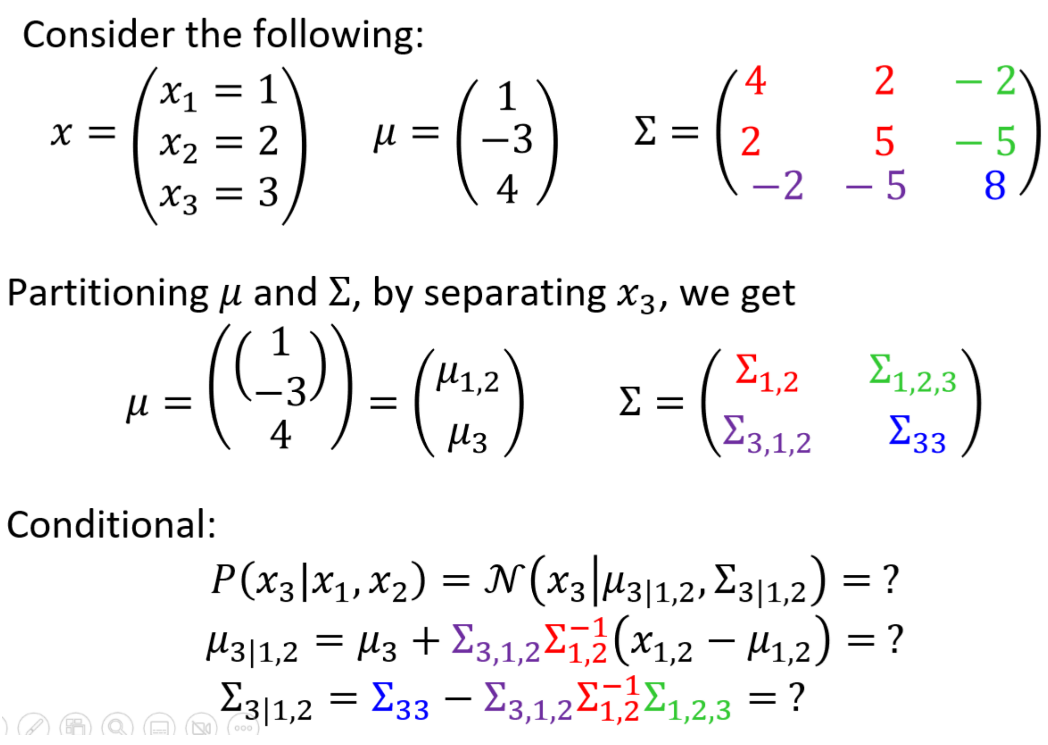
Advanced Artificial Intelligence, 2425

Workshop Task: Quiz on Gaussian Processes

Question 1

20 points

Use the exercises provided in lecture slides 22 and 24 to answer the following question marks.



Consider the following:

$$x = \begin{pmatrix} x_1 = 1 \\ x_2 = 2 \\ x_3 = 3 \end{pmatrix} \quad \mu = \begin{pmatrix} 1 \\ -3 \\ 4 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 4 & 2 & -2 \\ 2 & 5 & -5 \\ -2 & -5 & 8 \end{pmatrix}$$

Partitioning μ and Σ, by separating x_3, we get

$$\mu = \begin{pmatrix} \begin{pmatrix} 1 \\ -3 \end{pmatrix} \\ 4 \end{pmatrix} = \begin{pmatrix} \mu_{1,2} \\ \mu_3 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \Sigma_{1,2} & \Sigma_{1,2,3} \\ \Sigma_{3,1,2} & \Sigma_{33} \end{pmatrix}$$

Conditional:

$$P(x_3|x_1, x_2) = \mathcal{N}\left(x_3 \middle| \mu_{3|1,2}, \Sigma_{3|1,2}\right) = ?$$
$$\mu_{3|1,2} = \mu_3 + \Sigma_{3,1,2}\Sigma_{1,2}^{-1}\left(x_{1,2} - \mu_{1,2}\right) = ?$$
$$\Sigma_{3|1,2} = \Sigma_{33} - \Sigma_{3,1,2}\Sigma_{1,2}^{-1}\Sigma_{1,2,3} = ?$$

To carry out the calculations above, you can use manual calculations and/or a terminal with python and numpy imported. For example, the conditional probability density can be calculated according to:

```
def get_gaussian_density_from_var(self, x, mean, var):
  e_val = -np.power((x-mean), 2)/(2*var)
  return (1/(np.sqrt(2*np.pi*var))) * np.exp(e_val)
```

Prompt	Answer
1 What is the conditional mean?	something else
2 What is the conditional covariance?	-3
3 What is the conditional probability density?	0.01600

1

-1

-0.01600

Question 2

10 points

In the Radial Basis Function (RBF) kernel, the gamma parameter adjusts the similarity between datapoints. Run program **`KernelRBF-Example-MiniToyData.py`** for each of the following gamma values: 0, 0.25, 0.5, 0.75, 1. This program makes use of the first 10 training datapoints of the non-linearly separable data from last week, and calculates RBF kernel vectors on two test datapoints. Those test datapoints belong to two different classes, which can be seen when open and inspecting the dataset.

What **gamma value** separates the most those two test datapoints? Choosing this value is one of the things a computer program has to optimise when working with kernel-based methods such as Gaussian Processes.

(A) 0

(B) 0.25

(C) 0.5

(D) 0.75

(E) 1

Question 3

10 points

Train a Gaussian Process using the non-linearly separable data from last week as follows:

```
python gpytorch_GPR.py .\data\data-nonlinearlyseparable-train.csv .\data\data-nonlinearlyseparable-test.csv
```

This program is able to run you will need the following dependency: **`pip install gpytorch`**

Re-run the program but prior to that set the flag `STANDARDISE_DATA` to True.

Did data standardisation helped to obtain better classification accuracy?

(A) True

(B) False

Question 4

10 points

Run the same program as the previous question, with `STANDARDISE_DATA`=True, for each of the following choices of kernel:

- `self.mean_module = gpytorch.means.ConstantMean()`
- `self.covar_module = gpytorch.kernels.RBFKernel()`
- `self.covar_module = gpytorch.kernels.MaternKernel(nu=0.5) # matern12`
- `self.covar_module = gpytorch.kernels.MaternKernel(nu=1.5) # matern32`
- `self.covar_module = gpytorch.kernels.MaternKernel(nu=2.5) # matern52`
- `self.covar_module = gpytorch.kernels.MaternKernel(nu=3.5) # matern73`
- `self.covar_module = gpytorch.kernels.RationalQuadraticKernel()`
- `self.covar_module = gpytorch.kernels.PolynomialKernel(degree=2)`
- `self.covar_module = gpytorch.kernels.PolynomialKernel(degree=3)`

Which one performed better on the non-linearly separable data? Leave the RBF kernel as default once you are done with this question.

(A) `RBFKernel()`

(B) `MaternKernel(nu=0.5) # matern12`

(C) `aternKernel(nu=1.5) # matern32`

(D) `MaternKernel(nu=2.5) # matern52`

(E) `RationalQuadraticKernel()`

(F) `PolynomialKernel(2)`

(G) `PolynomialKernel(3)`

Question 5

10 points

Train a Gaussian Process twice using the bank note data from last week. Set the flag STANDARDISE_DATA to False then run, followed by setting it to True and re-running the program.

```
python gpytorch_GPR.py .\data\data_banknote_authentication-train.csv
.\data\data_banknote_authentication-test.csv
```

Did data standardisation helped to obtain better classification accuracy?

(A) True

(B) False

Question 6

10 points

Train a Gaussian Process twice but this time using the Cardiovascular data from last week. Set the flag STANDARDISE_DATA to False then run, followed by setting it to True and re-run the program.
`python gpytorch_GPR.py .\data\cardiovascular_data-original-train.csv .\data\cardiovascular_data-original-test.csv`

Did data standardisation helped to obtain better classification accuracy?

(A) True

(B) False

Question 7

10 points

Choose any of the datasets above (non-linearly separable toy, bank note, cardiovascular) and re-run the gpytorch_GPR.py program but set to `REDUCE_INPUTS_TO_2D` to True. The larger the dataset the longer it will take to generate a plot showing the training and test datapoints. Due to that, consider trying out a small dataset first then a larger one.

The plotted surface shows the learnt function (manifold=underlying structure of the data) over a 2D input space after applying Gaussian Process regression. To project data to 2D input space, dimensionality reduction is applied to the original data. In practice, we do not want to apply dimensionality reduction unless really justified--here is used only for visualisation purposes.

Where you able to see the datapoints and the learnt manifold of a GP model on any of our datasets?

(A) True

(B) False

Question 8

10 points

Install the following dependency, but change the cuda version according to that in your PC: **`pip install torch --index-url https://download.pytorch.org/whl/cu118`**

Train to Gaussian Process models with data standardisation but setting MAX_TRAIN_DATA to 10000 (instead of the default 2000). Run the program with GPU enabled (default) and with CPU only (self.device = 'cpu').

Was the GPU-based model quicker to train and evaluate than the CPU-based?

Ⓐ True

Ⓑ False

Question 9

10 points

Train three Gaussian Process models on the Cardiovascular data, but use the whole test set and following amounts of training data: 100, 1000, and 10000. Use MAX_TRAIN_DATA to set the amount of training data, and the following to use the full test set: `X_test, Y_test = self.load_data(datafile_test, False) # False to use the entire test set`

Do your results help to support that Gaussian Processes are data efficient?

Ⓐ True

Ⓑ False