Brunel University, London

MA5634/5663 - Fundamentals of Machine Learning

Assignment 2022/23

Student:
Vasileios Diplas (2273633)

Module leader:
Dr. Simon Shaw

1 TASK 1

1.1 Overview of the k-Nearest Neighbour (k-NN) method

1.1.1 Description of the data set

The whole data set of `breast cancer` test results is comprised of $N_B = 569$ observations, 30 features, and 1 target variable. From assumption, it is requested to do analysis with 5 features and 1 categorical binary variable suppose $\mathbf{Y}$ where it's been converted from {*'Benign'*,*'Malignant'*} to $\{0,1\}$. The features are $\mathbf{X}_1$ = '*worst concave points*', $\mathbf{X}_2$ = '*worst symmetry*', $\mathbf{X}_3$ = '*smoothness error*', $\mathbf{X}_4$ = '*concavity error*', $\mathbf{X}_5$ = '*mean perimeter*'. An important piece of information is that there is no missing data. Also, we normalized the data, because we would like to make the attributes to be on a similar scale.

1.1.2 Short overview of k-NN algorithm

We were asked about performing k-NN in the data set $\mathcal{B} = (\mathbf{X}, \mathbf{Y})$ where $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4, \mathbf{X}_5)$ with given hyperparameters $k = 5$ and $p = 6$. The k-Nearest Neighbour algorithm is a non-parametric supervised machine learning algorithm, which can be used for classification or regression. According to an article from Inna Logunova on K-Nearest Neighbors Algorithm for ML [1]: " *the main concept behind k-nearest neighbors is as follows. Given a point whose class we do not know, we can try to understand which points in our feature space are closest to it. These points are the k-nearest neighbors. Since similar things occupy similar places in feature space, it's very likely that the point belongs to the same class as its neighbors. Based on that, it's possible to classify a new point as belonging to one class or another.*" The numerical features $\mathbf{X}$ will be used to find the similarities between the features, and the categorical target variable $\mathbf{Y}$ will help in classification.

1.1.3 Hyperparameters

Suppose that $u = (k, p)$. In the analysis $u = (5, 6)$. For a choice of $k = 5$, the goal is to find for a new data point the 5 nearest neighbors. The new data point will be classified in "`class i`" , if

$$\text{Prob}(\texttt{"neighbors of class i"}) = \frac{N_i}{5} > \text{Prob}(\texttt{"neighbors of class j"}) = \frac{N_j}{5}$$

for restrictions $i, j = \{0,1\}$ and $i \neq j$. Also, the idea to use *Minkowski* distance is to find the distance (similarity) between the new sample suppose $\mathbf{x}^* \in \mathbb{R}^5$ and training data points, and then find the k=5-nearest neighbors. For $p = 6$ the *Minkowski* is:

$$d\left(x^*, X^{(train)}\right) = \left(\sum_{m=1}^{N_{train}=341} \left|x^* - x_m^{(train)}\right|^6\right)^{\frac{1}{6}} = \left|\left|x^* - x^{(train)}\right|\right|_6$$

1.2 Explain the choice of train/test split.

We have to train this algorithm by splitting the data into training and testing sets. The training set will train the algorithm and the test set will do predictions. So, $\mathcal{B} = \mathbf{q} \times \mathcal{B}_{\mathbf{train}} + (\mathbf{1} - \mathbf{q}) \times \mathcal{B}_{\mathbf{test}}$. It is important to give more data in the training set, in order to train the algorithm because if we train it well, it will give better classification results. For that reason, $\mathbf{q} = \mathbf{0.6}$ is a good choice $(60\% Training : 40\% Testing)$.

1.3 Explain the calculation of $\text{Prob}(P|\{-\})$

The sample space here is $\mathbf{\Omega} = \{\mathbf{P}, \mathbf{N}, \{+\}, \{-\}\}$, where $\mathbf{P}, \mathbf{N}$ (Positive and Negative respectively) are the events that labeled as **True** and $\{+\}, \{-\}$ (Positive and Negative respectively) that labeled as **Predicted**. The confusion

[1] URL: `https://serokell.io/blog/knn-algorithm-in-ml`, September 20th, 2022, "How does the kNN classification algorithm work?", access date: 13/05/2023

matrix for the True positive class *"Malignant"* and True negative class *"Benign"* is shown below. **TP** represents the True Positives, **FN** represents the False Negatives, **FP** represents the False Positives and **TN** represents the True Negatives.

$$\left[\begin{array}{c:c:c:c} N_B & \textbf{Label} & \overset{\{+\}}{\textbf{Predicted}} & \overset{\{-\}}{\textbf{Predicted}} \\ \hdashline \textbf{Label} & \textbf{Class} & \mathit{Malignant} & \mathit{Benign} \\ \hdashline \textbf{True} & \mathit{Malignant} & TP & FN \\ \hdashline \textbf{True} & \mathit{Benign} & FP & TN \end{array}\right] = \left[\begin{array}{c:c:c:c} N_B = 228 & \textbf{Label} & \overset{\{+\}}{\textbf{Predicted}} & \overset{\{-\}}{\textbf{Predicted}} \\ \hdashline \textbf{Label} & \textbf{Class} & \mathit{Malignant} & \mathit{Benign} \\ \hdashline \textbf{True} & \mathit{Malignant} & 145 & 4 \\ \hdashline \textbf{True} & \mathit{Benign} & 7 & 72 \end{array}\right]$$

The Probability for a Positive data point (Malignant) given that the prediction is Negative (Benign) is the False Negative Rate.

$$\text{Prob}(\mathbf{P}|\{-\}) = \frac{\text{Prob}(\mathbf{P} \cap \{-\})}{\mathbf{P}(\{-\})} = \frac{\mathbf{FN}}{\mathbf{FN}+\mathbf{TP}} = \frac{4}{4+145} = 0.026846$$

2 TASK 2

2.1 Short overview of PCA

According to Wikipedia: "*Principal component analysis (PCA) is a popular technique for analyzing large datasets containing a high number of dimensions/features per observation, increasing the interpretability of data while preserving the maximum amount of information, and enabling the visualization of multidimensional data.*" [2]. So, in PCA we create uncorrelated variables that maximize variance. More concretely, we consider the previous standardized training data set

$$\mathbf{X_{train}} = \left(\mathbf{X_1^{(train)}}, \mathbf{X_2^{(train)}}, \ldots, \mathbf{X_5^{(train)}}\right), \mathbf{X_i}^{\mathbf{(train)}} \in \mathbb{R}^{\mathbf{341 \times 1}} \qquad (i = 1,2,3,4,5)$$

The data covariance matrix is:

$$\mathbf{S} = \frac{\mathbf{1}}{\mathbf{341}} \mathbf{X_{train}^T X_{train}}, \quad \text{where} \quad \mathbf{S} \in \mathbb{R}^{\mathbf{5 \times 5}}$$

From the theorem of eigendecomposition, the matrix $\mathbf{S}$ can be written as $\mathbf{VDV^T}$, where $\mathbf{V}$ is an $\mathbf{5 \times 5}$ matrix of eigenvectors and $\mathbf{D}$ is a diagonal matrix whose entries are the eigenvalues of $\mathbf{S}$. The new coordinates for the data points are given from the following formula $\mathbf{Z} = \mathbf{X_{train} V}$. Also, if $(\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_5})$ is an orthonormal basis of $\mathbb{R}^5$ with the property that for every j and $i \neq j$: $\mathbf{v_j^T v_j} = \mathbf{1}, \mathbf{v_i^T v_j} = \mathbf{0}$, then $\tilde{\mathbf{z}} = \mathbf{v_j v_j^T x}$ is the orthogonal projection of $\mathbf{x}$ onto the $\mathbf{v_j}$. In the end, every principal component has a variance equal to its corresponding eigenvalue.

2.2 Variance captured by the value of $n_c = 2$

After ordering the eigenvalues by the descending method the variance explained by the first two components is given by the following:

$$\mathsf{V_{n_c}} = \mathsf{v_1^T S v_1} + \mathsf{v_2^T S v_2} = \lambda_1 \mathsf{v_1^T v_1} + \lambda_2 \mathsf{v_2^T v_2} = \lambda_1 + \lambda_2 = 3.572286445505426$$

The proportion of variance explained (in %) is 71.23621064350999%.

[2] Wikipedia, https://en.wikipedia.org/wiki/Principal_component_analysis, access date: 14/05/2023

2.3 k-NN with PCA VS k-NN without PCA

We only kept $n_c = 2$ components. So, the new data is $\mathbf{Z_{train}} = \mathbf{X_{train}V}$, where $\mathbf{Z_{train}} \in \mathbb{R}^{\mathbf{341\times 2}}$ and $\mathbf{Z_{test}} = \mathbf{X_{test}V}$,where $\mathbf{Z_{test}} \in \mathbb{R}^{\mathbf{228\times 2}}$. We performed the k-NN algorithm and we excluded that the accuracy is 0.9122807017543859. We define with $\mathbf{M'}$ the process with PCA and $\mathbf{M}$ without PCA. The confusion matrix is as follows:

		$\{+\}$	$\{-\}$
M	**Label**	**Predicted**	**Predicted**
Label	**Class**	*Malignant*	*Benign*
True	*Malignant*	145	4
True	*Benign*	7	72

VS

		$\{+\}$	$\{-\}$
M′	**Label**	**Predicted**	**Predicted**
Label	**Class**	*Malignant*	*Benign*
True	*Malignant*	137	12
True	*Benign*	8	71

The accuracy of k-NN after PCA is approximate **0.912** and for k-NN without PCA is approximate **0.952** with $|Acc_{\mathbf{M}} - Acc_{\mathbf{M'}}| \approx 4\times 10^{-2}$. The outcome is outstanding because by reducing the dimension of data we achieve also high performance.

Method	**Sensitivity**	**Specificity**	**FPR**	**FNR**	**PPV**	**NPV**
M	0.973	0.911	0.089	0.027	0.954	0.947
M′	0.919	0.899	0.101	0.081	0.945	0.855

The similarities between **M** and **M′** methods are the **Specificity**, **FPR**, and **PPV**. By reducing the dimensions of data **Specificity**, **FPR**, and **PPV** are approximately near. On the other hand, we observe a greater difference in Sensitivity, **FNR**, and **NPV**. So we understand that by performing PCA and choosing $\mathbf{n_c = 2}$ components the k-NN lacks in predicting Negative results (*'Benign'*) and has approximately the same Positive predictive power (*'Malignant'*). Considering all of the above the use of PCA with $\mathbf{n_c = 2}$ is recommended.

3 Task 3

3.1 Short overview of SVD

According to Wikipedia: "*The singular value decomposition (SVD) is a factorization of a real or complex matrix. It generalizes the eigendecomposition of a square normal matrix with an orthonormal eigenbasis to any $m \times n$ matrix.*" [3] SVD theorem states that for a rectangular matrix $A \in \mathbb{R}^{m\times n}$ with rank $r \in [0, \min\{m,n\}] \cap \mathbb{N}$ the decomposition can be in the following form:

$$\mathbf{A} = \mathbf{U\Sigma V^T}$$

- $\mathbf{U} \in \mathbb{R}^{m\times m}$ orthogonal matrix with column vectors $u_i, i = 1,2,...,m$ with the property $\mathbf{U^TU} = \mathbf{I_m}$
- $\mathbf{V} \in \mathbb{R}^{n\times n}$ orthogonal matrix with column vectors $v_j, j = 1,2,...,n$ with the property $\mathbf{V^TV} = \mathbf{I_n}$
- $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$, which is a diagonal matrix, for which the diagonal contains the singular values. $\Sigma_{ii} = \begin{cases} 0 & ,i > r \\ \sigma_i^2 & ,i \leq r \end{cases}$ where $r \in [0, \min\{m,n\}] \cap \mathbb{N}$

In our case $\mathbf{m = 341}$ and $\mathbf{n = 5}$. So, $\mathbf{U} \in \mathbb{R}^{341\times 341}$, $\mathbf{V} \in \mathbb{R}^{5\times 5}$ and $\mathbf{\Sigma} \in \mathbb{R}^{341\times 5}$. But matrix $\mathbf{\Sigma}$ contains zeros when $i-th$ row and $i-th$ column are greater than the $rank(\mathbf{X_{train}})$. So, $\mathbf{U} \in \mathbb{R}^{341\times rank(\mathbf{X_{train}})}$, $\mathbf{\Sigma} \in \mathbb{R}^{rank(\mathbf{X_{train}})\times 5}$.

[3] Wikipedia,https://en.wikipedia.org/wiki/Singular_value_decomposition,Singular value decomposition,access date: 15/05/2023

So, **SVD** is an important method because by decreasing the dimensionality of a given data set we make the problem more simple. An important notice is that we want to keep the most information.

3.2 Pairplot and rank of the data

- After loading the data `dfth = TSLAhistory.csv` and `dftu =TSLAupdate.csv`, we did not find missing values. Also, we standardized the data, because the variable "*Volume*" has different values than the other four variables. From the pair plot of the `dfth` data, we conduct that between "*Volume*" and the other variables there is a big variation. We observe also that for "*Adj Close*" and "*Close*" are highly positively correlated because the data is concentrated in a straight line. The relationship between the remaining variables shows also a positive correlation but not to such an extent as we saw for "*Adj Close*" and "*Close*". For that reason, we expect to have $6-1=5$ dominant independent components.

- The *rank* of a matrix corresponds to the maximum of linearly independent variables of $\mathbf{X_{train}}$ set. We found that the $rank(\mathbf{X_{train}}) = 5$.

3.3 Finding the $\mathbf{K_c}$

Let the covariance matrix $\mathbf{C}$ which is given by $\mathbf{C} = \frac{\mathbf{1}}{\mathbf{341}}\mathbf{X^T_{train}}\mathbf{X_{train}}$. From the eigendecomposition theorem, we know that matrix $\mathbf{C}$ can be written in the following form: $\mathbf{C} = \mathbf{VDV^T}$,where $\mathbf{V}$ is the matrix of eigenvectors and $\mathbf{D}$ is the diagonal matrix of eigenvalues. If we now perform singular value decomposition for $\mathbf{X_{train}}$, we know that:

$$\mathbf{X_{train}} = \mathbf{U\Sigma V^T}$$

As we said before $\mathbf{X_{train}V}$ contains the principal component scores. $\mathbf{X_{train}V} = \mathbf{U\Sigma V^T V} = \mathbf{U\Sigma}$. We conduct that the columns of $\mathbf{U\Sigma}$ are principal components and the $\mathbf{X_{train}V}$ is the transformation of the $\mathbf{X_{train}}$ into a new data set that belongs to a new coordinate system. In this particular coordinate system, the dimensionality will be decreased. It depends on us how many singular values we have to choose. According to the spectral theorem, if we want to work with **c** singular component(s) where $c \in [1,5] \cap \mathbb{N}$ the $\mathbf{X_{train}}$, $\varepsilon_\mathbf{c}$ (error) can be written as:

$$\mathbf{X_{train}} = \sum_{\mathbf{k=1}}^{\mathbf{c}} \sigma_\mathbf{k}\mathbf{u_k v_k^T} = \mathbf{X^{(c)}_{train}}, \quad \varepsilon_\mathbf{c} = \left|\left|\mathbf{X_{train}} - \mathbf{X^{(c)}_{train}}\right|\right|_\mathbf{5}$$

Taking into consideration all of the above we define the variable $\mathbf{K_c}$, where $c \in [1,5] \cap \mathbb{N}$. Also with $\mathbf{V_c}$, we define the matrix of eigenvectors including the c first column(s). For instance $\mathbf{V_2} = (\mathbf{v_1}, \mathbf{v_2})$

$$\underbrace{\mathbf{X_{train}}}_{\mathbb{R}^{\mathbf{341\times 5}}} \xrightarrow{\text{SVD}} \underbrace{K_c}_{\mathbb{R}^{\mathbf{341}\times\mathbf{c}}} = \mathbf{X_{train}V_c} = (\sigma_\mathbf{1}\mathbf{u_1}, \sigma_\mathbf{2}\mathbf{u_2}, \ldots, \sigma_\mathbf{c}\mathbf{u_c})$$

3.4 Finding the $\mathbf{Q_c}$

We 'trained' the SVD by computing the matrices $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}$ for the $\mathbf{X_{train}}$ data set. So our strategy is to keep the same $\mathbf{V}, \mathbf{\Sigma}$ ($\mathbf{V}$ is a matrix of right eigenvectors of $\mathbf{X^T_{train}X_{train}}$ matrix and $\mathbf{\Sigma}$ diagonal matrix of singular values) and complete the SVD construction of $\mathbf{X_{test}}$ data set by finding the left singular vectors that are orthonormal. Suppose $\tilde{\mathbf{U}}$ is the new matrix of left singular vectors. Particularly:

$$\mathbf{X_{test}} = \tilde{\mathbf{U}}\mathbf{\Sigma V^T} = \mathbf{X_{test}V} = \tilde{\mathbf{U}}\mathbf{\Sigma} \Rightarrow \tilde{\mathbf{U}} = \mathbf{X_{test}V\Sigma^{-1}} \Rightarrow \tilde{\mathbf{u_c}} = \frac{\mathbf{1}}{\sigma_\mathbf{c}} \times \mathbf{X_{test}v_c} \quad \text{,where} \quad c \in [1,5] \cap \mathbb{N}$$

We understand that the transformation of the $\mathbf{X_{test}}$ data set is given by the following formula: $\mathbf{X_{test}V}$. Taking into consideration the above we define the variable $\mathbf{Q_c}$, where $c \in [1,5] \cap \mathbb{N}$. Also with $\mathbf{V_c}$, we define the matrix of eigenvectors including the c first column(s).

$$\mathbf{Q_c} = \mathbf{X_{test}} \mathbf{V_c} = (\sigma_\mathbf{1} \mathbf{\tilde{u}_1}, \sigma_\mathbf{2} \mathbf{\tilde{u}_2}, ..., \sigma_\mathbf{c} \mathbf{\tilde{u}_c}) \quad , \mathbf{Q_c} \in \mathbb{R}^{\mathbf{228 \times c}}$$

3.5 Graphical representation for $\mathbf{c = 1}$

(With blue color we denote the training data and its derivatives and with red color the testing data and its derivatives) We demonstrate three plots in Figure 1a. The First plot $\mathcal{G}_1^{(1)}$ shows the relationship between the variables "*Open*" and "*Volume*" for the $\mathbf{X_{train}}$ and (red points) $\mathbf{X_{test}}$ data (blue points). The second plot $\mathcal{G}_2^{(1)}$ shows again the relationship between the variables "*Open*" and "*Volume*" for the $\mathbf{X_{test}}$ data (blue points). The red straight line in $\mathcal{G}_2^{(1)}$ is the $\mathbf{K_1} = \sigma_\mathbf{1} \mathbf{u_1} \in \mathbb{R}^{341 \times 1}$. It makes sense because we have chosen only **one** singular value. So, we transformed the $\mathbf{X_{train}}$ data, and the new data $\mathbf{K_1}$ contains the information with error $\varepsilon_1 \approx 15.53$ in one dimension. Concluding, the difference between $\mathcal{G}_1^{(1)}$, $\mathcal{G}_2^{(1)}$ is that in $\mathcal{G}_1^{(1)}$ we have the original $\mathbf{X_{train}}$ for the variables "*Open*" and "*Volume*" and in $\mathcal{G}_2^{(1)}$ we have the transformed $\mathbf{X_{train}}$ using **one** singular value, containing the most information of **all** the features.

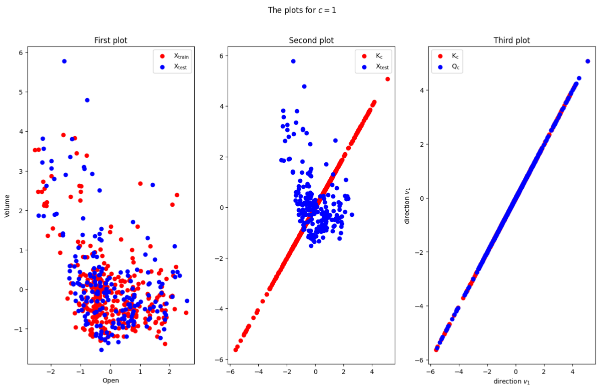
On the other hand if we transform also the $\mathbf{X_{test}}$ for $\mathbf{c = 1}$ we conduct that: $\mathbf{Q_1} = \sigma_\mathbf{1} \mathbf{\tilde{u}_1} \in \mathbb{R}^{228 \times 1}$. So, the information with error $\tilde{\varepsilon}_1 \approx 15.29$ is on a straight line too. That is clearly seen in the Third plot $\mathcal{G}_3^{(1)}$. We noticed that the straight line of $\mathbf{K_1}$ is touching the straight line of $\mathbf{Q_1}$. This means that $\mathbf{u_1}, \mathbf{\tilde{u}_1}$ are approximately the same. So, this amount of training data can give nice results. Concluding, the difference between $\mathcal{G}_3^{(1)}$ and $\mathcal{G}_1^{(1)}$ is that in $\mathcal{G}_3^{(1)}$ we observe the transformed data for the $\mathbf{X_{train}}$, $\mathbf{X_{test}}$ data containing the most information in straight lines ($\mathbf{c = 1}$) respectively. In $\mathcal{G}_1^{(1)}$ we observe the relationship between the variables "*Open*" and "*Volume*" for the $\mathbf{X_{train}}$ and $\mathbf{X_{test}}$ data. The difference between $\mathcal{G}_3^{(1)}$ and $\mathcal{G}_2^{(1)}$ is that in $\mathcal{G}_3^{(1)}$ we have the transformed data for each training and testing set, but in $\mathcal{G}_2^{(1)}$ we have the original $\mathbf{X_{test}}$ for variables "*Open*" and "*Volume*".

3.6 Graphical representation for $\mathbf{c = 2}$

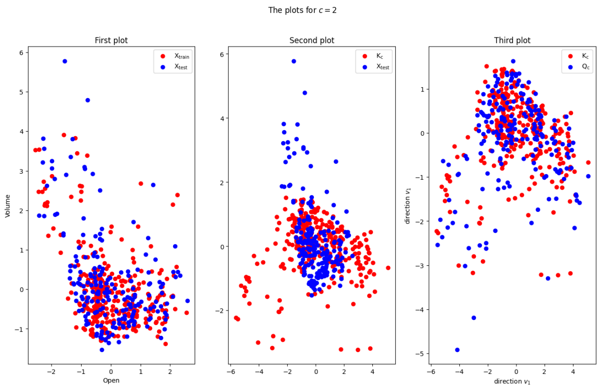
For the value $\mathbf{c = 2}$ the plots $\mathcal{G}_2^{(1)}$, $\mathcal{G}_3^{(1)}$ will change, but the $\mathcal{G}_1^{(1)}$ will be the same. Now, the information for **two** singular values is on 2 Dimensional and not a straight line as before. The transformed data $\mathbf{K_2} = (\sigma_\mathbf{1} \mathbf{u_1}, \sigma_\mathbf{2} \mathbf{u_2}) \in \mathbb{R}^{341 \times 2}$, $\mathbf{Q_2} = (\sigma_\mathbf{1} \mathbf{\tilde{u}_1}, \sigma_\mathbf{2} \mathbf{\tilde{u}_2}) \in \mathbb{R}^{228 \times 1}$ with errors $\varepsilon_2 \approx 2.12$ and $\tilde{\varepsilon}_2 \approx 1.77$ respectively belong to $\mathbb{R}^2$. We suppose the basis for $\mathbf{U} = \{\mathbf{u_1}, \mathbf{u_2}\}$ and for $\mathbf{\tilde{U}} = \{\mathbf{\tilde{u}_1}, \mathbf{\tilde{u}_2}\}$. Then according to the spectral theorem, we can write in $\mathbb{R}^2$:

$$\mathbf{K_2} = \sigma_\mathbf{1} \mathbf{u_1} + \sigma_\mathbf{2} \mathbf{u_2}, \quad \mathbf{Q_2} = \sigma_\mathbf{1} \mathbf{\tilde{u}_1} + \sigma_\mathbf{2} \mathbf{\tilde{u}_2}$$

In Figure 1b we observe the $\mathcal{G}_1^{(2)}$,$\mathcal{G}_2^{(2)}$,$\mathcal{G}_3^{(2}$ plots. The $\mathcal{G}_3^{(2)}$ plot is very important, because we observe that $\mathbf{K_2}, \mathbf{Q_2}$ are approximately the same as in $\mathcal{G}_3^{(1)}$ with the only difference the transformed data are in **different** dimensions.



(a) The plots $\mathcal{G}_1^{(1)}$,$\mathcal{G}_2^{(1)}$,$\mathcal{G}_3^{(1)}$for $\mathbf{c = 1}$



(b) The plots $\mathcal{G}_1^{(2)}$,$\mathcal{G}_2^{(2)}$,$\mathcal{G}_3^{(2)}$for $\mathbf{c = 2}$