BACHELOR

Estimating the number of clusters in a block Markov chain

Cronk, Thomas N.

Award date:
2023

Eindhoven University of Technology
Department of Mathematics and Computer Science

Estimating the number of clusters in a block Markov chain

Authors:
Thomas Cronk
(1611232)
t.n.cronk@student.tue.nl

Supervisor:
dr. Jaron Sanders
jaron.sanders@tue.nl

Abstract

In this thesis we look at clustering in Block Markov Chains (BMCs). A Block Markov Chain is a Markov Chain with the property that the transition matrix consists of K clusters. For any pair of states within the same cluster, the distribution that they move to any other state within the state space should be the same for the pair. Now suppose that we are given a sample path of some BMC. In this paper we will present an algorithm to extract the number of clusters (which is an unknown) to the algorithm. Furthermore, this algorithm also clusters the states according to their original groups. We will outline a proof that this algorithm does this in an asymptotically accurate way. Firstly, we take inspiration from different papers [13], [14], [19] which allow us to create an algorithm which can both estimate K and create clusters from a BMC. Secondly, we will test our algorithm on both synthetic and real data to get an idea of the real world capabilities of our algorithm.

August 14, 2023

Contents

1 Introduction

In recent times much research has been conducted in the field of the Stochastic Block Model (SBM). The Stochastic Block Model first introduced in [7] and then expanded in [16] can best be seen as a random graph with n vertices in which one observes if there is an edge between vertices x and y and with the added property that each state belongs to a group of similar behaving states. This group is called a cluster and if two states belong to the same cluster they have the same probability distribution to move to any other state in the state space. Furthermore, the amount of clusters in a SBM will be denoted by $K \leq n$.

As an example one can think of the case when the set of n vertices denoted by V is split in two different sets V_1 and V_2 implying $K = 2$, such that each state can only be in one of the two sets. This example is visualized in Figure 1.

Assume we have two states $x, y \in V$, then x, y belong to the same group (i.e., there is an edge between vertices x and y) with probability $p \in (0, 1)$. The probability that they do not belong to the same group (i.e., there is no edge between vertices x and y) is given by $1 - p \in (0, 1)$. A property of this model is that the probability that one observes an edge between two vertices is independent of observing an edge between any other pair of vertices.

We study the SBM because it lends itself as a good test for clustering algorithms. The reason for studying clustering algorithms is that if we detect certain group structures within a random graph, we can reduce the dimensionality of a graph which often allows for an easier analysis.

Now the SBM is quite restrictive and people have researched other models while relaxing some of the conditions of a SBM. For example, in a SBM one looks at observing an edge between vertices i and j. However, one can also introduce the property that this edge needs to have a certain value called a label. Thus, now one looks at observing an edge with label $l \in \{0, \ldots, L\}$ between vertices i and j. This model is called the Labeled Stochastic Block Model (LSBM) [19]. Notice that if $L = 1$, then one recovers the original SBM. The reason for studying this model is that it has some overlap with the model we will study in this bachelor thesis and we can use some of the results found in the setting of a LSBM and transfer them over to our setting.

The setting of this bachelor thesis considers a sample path of a Markov Chain with a set of n vertices with a group structure in its transition matrix. This model is known in the literature as a Block Markov Chain (BMC) [13]. A Block Markov Chain can be seen as a random graph consisting of n vertices in which the probability of moving between vertices i and j is given by $p(i, j)$. The goal of this bachelor thesis is to detect the number of clusters within a BMC when we are given a sample path of length T denoted by $X_0, \ldots, X_T$. Each state in a cluster is assumed to have the same transition probability to move to any other cluster as any other state within the same cluster. An important observation to make is that the Markov property tells us that the observations one has from a BMC are dependent on the previous state. Hence, the probability of moving between vertices i and j is dependent

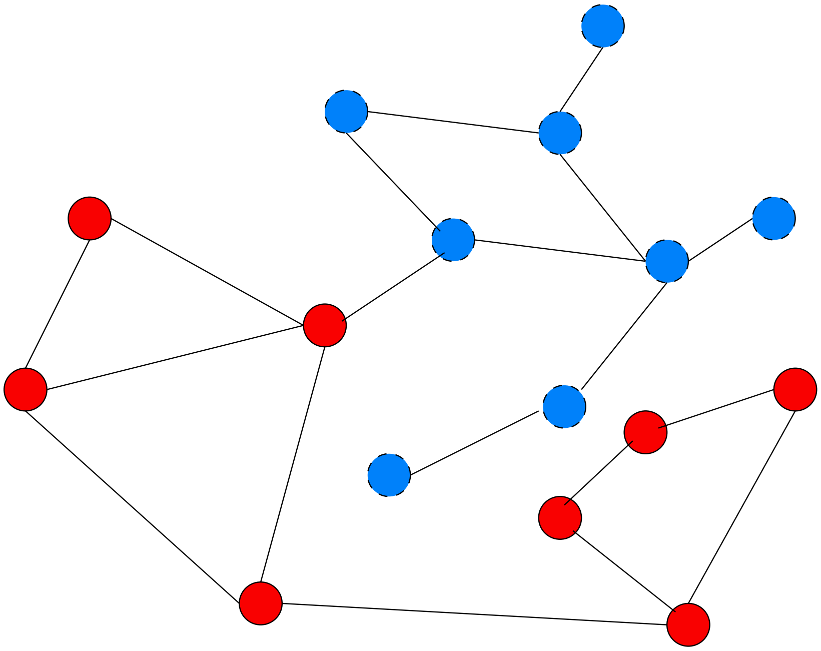

Figure 1: Example of a random graph of a SBM with $K = 2$ clusters.

on which group vertex i belongs to.

An example of a BMC can be the movement of a cow in a field [17]. Suppose we have three general areas in this field; a barn, a pond, and a field. Now it is easy to imagine that when looking at accurate GPS data from this cow, the probability of moving from somewhere next to the pond to somewhere in the field would be somewhat the same for all possible positions in the pond and the field. Hence this process could be seen as a BMC where we assume that the next movement of the cow is only based on the previous position of the cow. This example of a BMC can be visualized in Figure 2 in which the open circles denote possible positions for the cow to be in, and the arrows denote the probability of moving from a state to another state. Now with dashed arrows we have drawn a sample path of states $X_0, X_1, \ldots, X_T$ in this BMC.

Clustering in a BMC is different compared to clustering in a SBM because in a BMC consecutive samples of the sample path are dependent while in a SBM they are independent. However, the reason for studying clustering algorithms for BMCs, is that they are useful in various scientific fields like biology, machine learning and sociology. Some examples of applications of the clustering methods are: "detecting patterns in the stock market", "recognizing words in text", and "detecting codon pairing in human DNA" [17]. The detecting of codon pairing in human DNA is an example we will work out in detail in this bachelor thesis as well.

Besides SBMs, there are many more variants random graphs that have been researched. An example of a variant of the SBM would be the degree corrected SBM [8], [12], which

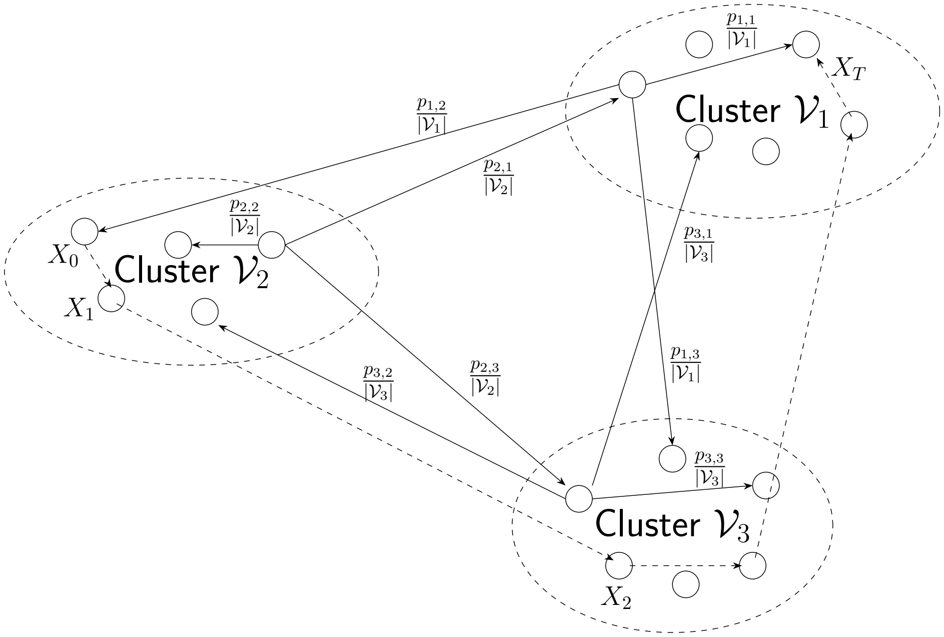
$\frac{p_{1,1}}{|\mathcal{V}_1|}$
X_T
$\frac{p_{1,2}}{|\mathcal{V}_1|}$
Cluster $\mathcal{V}_1$
$\frac{p_{2,1}}{|\mathcal{V}_2|}$
$\frac{p_{2,2}}{|\mathcal{V}_2|}$
X_0
Cluster $\mathcal{V}_2$
$\frac{p_{3,1}}{|\mathcal{V}_3|}$
X_1
$\frac{p_{1,3}}{|\mathcal{V}_1|}$
$\frac{p_{3,2}}{|\mathcal{V}_3|}$
$\frac{p_{2,3}}{|\mathcal{V}_2|}$
$\frac{p_{3,3}}{|\mathcal{V}_3|}$
Cluster $\mathcal{V}_3$
X_2

Figure 2: Example of a sample path denoted by $X_0, X_1, X_2, \ldots, X_T$ in a BMC with $K = 3$ clusters and transition probabilities

$$(p_{i,j})_{i,j\in\{1,2,3\}} = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix}$$

like the SBM preserves the expected value of their underlying matrix; but unlike the SBM they also preserve the degree of the edges of the random graph. Another variant of the SBM which allows for clusters to be overlapping, meaning that states can belong to more than one cluster, is known as the overlapping SBM which is studied [9], [10].

1.1 Focus of this bachelor thesis

In [13] two algorithms are presented. A so-called Spectral Clustering Algorithm that generates a estimate cluster assignment from a given sample path of a BMC, and a so-called Cluster Improvement Algorithm which improves any initial cluster assignment element-wise by maximizing a log-likelihood function. This specific Spectral Clustering Algorithm depends on the knowledge of knowing K up front, and we observe that in [19] a Spectral Algorithm which does not rely on the knowledge of K is given in the context of LSBMs.

Hence, in this thesis we focus our attention on the missing part of [13] which is the ability to estimate the amount of clusters K in the setting of a BMC. In order to estimate the number of clusters in a BMC, a great deal of inspiration was taken from the spectral algorithm given in [19] in the setting of a LSBM. This led to the following questions:

- How can we estimate the number of clusters K in a BMC from a given path?
- How long would the sample path of the BMC need to be, to have enough information to estimate K?

In order to convert some of the stopping criteria given in the algorithm in the context of a LSBM into a BMC some more sources were needed. [14] was helpful in giving insight into these bounds. This paper gives us bounds on the size of the singular values of a random counting matrix in the context of a BMC. These bounds will also provide our first estimator for K denoted by $\hat{K}^{\text{pre}}$. However, as we will see in the results, $\hat{K}^{\text{pre}}$ is not a very accurate estimator of K. Thus we use [19] to determine another estimator of K denoted by $\hat{K}^{\text{post}}$. As the results will show, $\hat{K}^{\text{post}}$ is a consistent and accurate estimator of K under the conditions that the sample path is long enough and that the sample path comes from a BMC. We will also explore situations in which the sample path does not come from a true BMC which shows us that if the path is long enough that both $\hat{K}^{\text{pre}}$ and $\hat{K}^{\text{post}}$ do not converge to K but to the amount of states n. Furthermore, if the length of the sample path is not long enough we will see that our ability to cluster states correctly is severely impaired.

1.2 Methodology

In order to answer the question if the number of clusters can be estimated from a BMC, assume that a sample path of a Block Markov Chain of length T_n is given. We will consider the behavior of our algorithm in the three different regimes $T_n = \omega(n \ln n)$, $T_n = o(n \ln n)$, $T_n = \Theta(n \ln n)$, which will be referred to as the dense, sparse, and critical regime. Our algorithm will identify two estimators of K referred to as $\hat{K}^{\text{pre}}$ and $\hat{K}^{\text{post}}$. As one might now expect from the notation, $\hat{K}^{\text{post}}$ is dependent on $\hat{K}^{\text{pre}}$ but not the other way around. Thus $\hat{K}^{\text{post}}$ can be seen as an improved version of $\hat{K}^{\text{pre}}$; however, the concepts on which $\hat{K}^{\text{post}}$ are built are fundamentally different than $\hat{K}^{\text{pre}}$.

The first thing which will be discussed in this bachelor thesis, is when it is possible to detect clusters using a given sample path of a BMC. This is done by referring to [13] which has used information theory to study the possibility of detecting clusters. Next, we will analyze a lower bound on the number of misclassified states for a specific set of clustering algorithms in a BMC. After this, we will present an algorithm which will be constructed in such a way that this algorithm approaches this fundamental limit as n grows large. Hence, our algorithm which will be called the *Spectral Clustering Algorithm with unknown K* works well in the cases when $n \to \infty$. The Spectral Clustering Algorithm with unknown K will consist of a couple of steps.

Firstly, one calculates the empirical transition matrix denoted by $\hat{N}$ consisting of the amount of transitions between any pair of states from a given sample path of a BMC. Secondly, if T_n is in a sparse regime and sometimes in the critical regime, we have to trim $\hat{N}$ in order to reduce the noise this empirical transition matrix suffers from which results in the matrix $\hat{N}_\Gamma$. After we obtain the trimmed the matrix, we make a rank-K approximation of $\hat{N}_\Gamma$. This is done using a concept called singular values thresholding and was first used in [4]. Using singular value thresholding we obtain our first estimator which we introduced as $\hat{K}^{\text{pre}}$. This allows us to create a rank-$\hat{K}^{\text{pre}}$ approximation of the trimmed empirical transition matrix $\hat{N}_\Gamma$. Now our estimator $\hat{K}^{\text{post}}$ is calculated using an adapted K-means algorithm which accounts for the fact that K is unknown to create the clusters of the underlying BMC. This adapted K-means algorithm accounts for the unknown K by providing an lower bound on the size of a new cluster which is based on Algorithm 2 of [19].

The idea of the singular value thresholding and the adapted K-means algorithm stems from [19]. Here it is done in the context of a LSBM. However, one has to be careful when adapting results obtained in the context of a LSBM and transferring them to the context of a BMC. For example, the empirical transition matrix in a LSBM is symmetric. Since if the label between two states x and y is known, then the label between the states y and x is the same and hence the edge has the same label. Now in the case of the BMC, the number of transitions from a state x to y is in general not the same as the number of transitions from y to x. Hence, in the context of a BMC one has to be careful when dealing with this empirical transition matrix since it is not symmetric.

In [13], the researchers provide two algorithms. The first being the Spectral Clustering Algorithm and the second being the Cluster Improvement Algorithm. The Spectral Clustering Algorithm with unknown K is as the name suggest similar to the Spectral Clustering Algorithm in [13], however, it does not rely on knowing K. The Cluster Improvement Algorithm takes the initial cluster assignment and assigns states elementwise to their optimal cluster. Now this Cluster Improvement Algorithm has not been adapted to account for the knowledge that K is unknown since using our Spectral Clustering Algorithm with unknown K we obtain an estimate for K which can be fed in to the Cluster Improvement Algorithm. However, in the results we will still analyze the performance increase gained from using the Cluster Improvement Algorithm after the Spectral Clustering Algorithm with unknown K.

1.3 Related work

1.3.1 Clustering in SBMs

As previously mentioned, clustering in SBMs has been researched in various papers. In [1], [3], the authors look at clustering in the setting of a SBM. [1] looks at how to retrieve exact clusters and provides an algorithm which can retrieve these clusters. However, this algorithm assumes that there are two clusters with a probability p that two states belong to the same group and probability $1-p$ that these states do not belong to the same group. This algorithm has been build on the research done in [6] in which SBMs consisting of two groups known as the binary symmetric Stochastic Block Model is studied. They show using the semidefinite programming relaxation of the maximum-likelihood estimator that their algorithm is optimal. The problem of how to deal with unknown parameters of the SBM was dealt with in [3]. Here an optimal algorithm was proposed in which the number of clusters is not needed.

For the optimality of these algorithms it is important to look at the research conducted in [2]. This paper ([2]) identified an important information gap in SBMs leading to a lower bound on the optimality of any clustering algorithm for a SBM. When one comes up with a new clustering algorithm in the setting of a SBM, it is important to prove that the algorithm is optimal. This is the reason why the lower bound on the optimality of any clustering algorithm for a SBM is an important result. It allows for the verification of optimality of any given clustering algorithm in the setting of a SBM, by only showing that an upper bound on the optimality of a clustering algorithm coincides with the lower bound identified in [2].

1.3.2 Clustering in BMCs

In [13] and [17] the researchers looked at recovering clusters from a Block Markov Chain with a known number of clusters K. The difference between clustering in a BMC compared to a SBM is that in the case of a BMC the samples are dependent, while in a SBM the observations one has consists of the edges of a random graph which are independent of each other.

[13] presents two algorithms in which the first algorithm is known as the Spectral Clustering Algorithm and the second algorithm is known as the Cluster Improvement Algorithm. The Spectral Clustering Algorithm consists of two steps. The first step is making a rank-K approximation of a random matrix in which each element in the i, jth position denotes the number of times the path jumped from state i to state j. After this a K-means algorithm is applied to assign each state to each of the K clusters. The Cluster Improvement Algorithm improves these initial estimated clusters by using a local maximization of a log-likelihood ratio.

Furthermore, [13] provides a proof that both the Spectral Clustering Algorithm and the Cluster Improvement Algorithm are optimal. This is done in a similar fashion as in the setting of the SBM. Firstly, a lower bound on the error is given which every clustering algorithm in the setting of a BMC will satisfy. Secondly, for both algorithms an upper bound is given on their performance which asymptotically matches this lower bound. These algorithms are then tested in real-life data which is done in [17]. We will use one of these real-life dataset to test our own model.

1.3.3 Clustering in the LSBM

Recall that in the introduction we introduced a property known as a label on the edges of a random graph of a SBM, which led us to the setting of the Labeled Stochastic Block Model (LSBM). [19] provides an optimal clustering algorithm in the setting of a LSBM which does not depend on the knowledge of the amount of clusters K. This algorithm is based on the authors previous research [18] which provides conditions for the ability to recover clusters and a fundamental lower bound on the performance of any clustering algorithm in the setting of a LSBM. The algorithm povided in [19] gives two estimators for the number of clusters K, which are denoted by $\tilde{K}$ and $\hat{K}$. We do have to note that [19] uses $\hat{K}$ as the only estimator of K.

The presented algorithm is a spectral algorithm comprised of a couple of steps. The first step in the spectral algorithm, makes a $\tilde{K}$-rank approximation by using singular value thresholding based on theory from [4]. After this step, $\hat{K}$ clusters are formed using a greedy K-means algorithm and stopping the process when the cluster size is below a certain threshold. After the K-means algorithm has made the initial $\hat{K}$ clusters, the remaining elements are assigned to their best fitting cluster.

1.3.4 Singular values of random matrices with dependencies

Studying the singular values of random graphs is done in many different papers. In [5] the researchers look at the eigenvalues of random matrices in a sparse regime of a SBM. They study the gap between the eigenvalues. This same concept of looking at a gap between singular values will also be needed in our case. The paper which describes this gap in singular values is [14]. This paper studies the behaviour of a random matrix in the setting of a BMC. The results of this paper are vital to our research which is why in an upcoming section we will discuss this gap in the singular values and the results of [14] in more detail. In [15] they look at the limiting distribution of the singular values in the setting of a BMC. Furthermore, [15] also looks at the limiting distributions of the empirical transition matrix of a BMC.

1.4 Notation

Let $\mathbf{1}$ be a vector consisting of all ones and let Δ^n denote the n dimensional probability simplex such that if $\beta \in \Delta^n$ we have that $\forall_{x\in\beta}|x| \leq 1$, $\beta \in \mathbb{R}^{n+1}$ and $\mathbf{1}^T\beta = 1$. Furthermore, the set of left-stochastic matrices are denoted by $\Delta^{K\times K}$ in which a matrix $X \in \mathbb{R}^{n\times n}$ with rows $\vec{x}_i = (x_{i,1}, \dots, x_{i,n})$ is in $\Delta^{K\times K}$ if all rows $\vec{x}_i$ of X satisfy for all $i = 1, \dots, n$ $(\vec{x}_i)^T \in [0,1]^n$ and $\sum_{j=1}^n x_{i,j} = 1$. Secondly, we give our definition of a l_p norm: let $\vec{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n$, then

$$\|\vec{x}\|_p = \left(\sum_{i=1}^n |x_i|^p\right)^{1/p} \qquad \text{where } p \in [1,\infty). \tag{1}$$

Now for any $m \times n$ matrix $A \in \mathbb{R}^{m\times n}$, we will indicate the rows of this matrix by A_i for $i \in \{1, \dots, m\}$ and its columns by $A_{\cdot,j}$ for $j \in \{1, \dots, n\}$. Furthermore, if we have an $m \times n$ matrix $A \in \mathbb{R}^{m\times n}$ then we define $A^0 = [A, A^T]$. Now the Frobenius norm and the spectral norm of a matrix $A \in \mathbb{R}^{m \times n}$ are defined as

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{i,j}^2}, \quad \|A\| = \sup_{b \in \mathbb{S}^{n-1}} \{\|Ab\|_2\} \tag{2}$$

there $\mathbb{S}^{n-1}$ denotes the n-dimensional unit sphere, i.e., $\mathbb{S}^{n-1} = \{\vec{x} \in (0,1)^n : \|\vec{x}\|_2 = 1\}$.

In this paper we mainly look at the asymptotic behavior of certain processes and functions. Therefore we introduce some notation to capture this asymptotic behavior.

- $f(n) \sim g(n)$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$;
- $f(n) = o(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$;
- $f(n) = O(g(n))$ if $\limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$.

Now we will introduce similar notation for random variables. Let $\{X_n\}_{n\geq 1}$ be a sequence of real-valued random variables and let $\{a_n\}_{n\geq 1}$ be a deterministic sequence. We define

- $X_n = o_{\mathbb{P}}(a_n) \iff \mathbb{P}\left[\frac{X_n}{a_n} \geq \delta\right] \to 0\,\forall_{\delta>0} \iff \forall_{\epsilon,\delta>0}\exists_{N_{\epsilon,\delta}\in\mathbb{N}}\mathbb{P}\left[\frac{X_n}{a_n} \geq \delta\right] \leq \epsilon\,\forall_{n\geq N_{\epsilon,\delta}}$
- $X_n = O_{\mathbb{P}}(a_n) \iff \forall_{\epsilon>0}\exists_{\delta_\epsilon>0,N_\epsilon\in\mathbb{N}}\mathbb{P}\left[\frac{X_n}{a_n} \geq \delta\right] \leq \epsilon\forall_{n\geq N_\epsilon}$
- $X_n = \Omega_{\mathbb{P}}(a_n) \iff \forall_{\epsilon>0}\exists_{\delta_\epsilon>0,N_\epsilon\in\mathbb{N}}\mathbb{P}\left[\frac{X_n}{a_n} \leq \delta\right] \leq \epsilon\,\forall_{n\geq N_\epsilon}$

Properties on these sequences have been proven in SM6.5 of [13].

2 State of the art

Inspiration for this thesis has been obtained from [13]. In this paper the researchers propose an algorithm for detecting clusters within a BMC. They also prove that this algorithm is asymptotically optimal under certain criteria. The proposed algorithm consists of two parts which will be referred to as the Spectral Clustering Algorithm and the Cluster Improvement Algorithm. Before we can analyze these two parts we have to delve a bit deeper in to when it is possible to cluster.

2.1 Block Markov Chains

Assume that one has been given a BMC consisting of K clusters and with the state space $\{1, \dots, n\} = V$. Each state $x \in V$ belongs to a cluster $\mathcal{V}_k$; and we assume that there exists $\alpha \in \Delta^{K-1}$ such that $\lim_{n\to\infty} \frac{|\mathcal{V}_k|}{n} = \alpha_k$ for $k \in \{1, \dots, K\}$. Furthermore, let the map $\sigma : \{1, \dots, n\} \to \{1, \dots, K\}$ which maps each state $x \in V$ to the kth cluster $\mathcal{V}_k$ such that $x \in \mathcal{V}_k$. It is important to note that a state $x \in V$ can only belong to one of the K clusters. Hence $\mathcal{V}_k \cap \mathcal{V}_l = \varnothing$ for all $k \neq l$. Now any BMC can be characterized by n, $\alpha \in \Delta^{K-1}$ and $p \in \Delta^{K\times(K-1)}$. We suppose that K, α, p are fixed but not known to our algorithm. Specifically a BMC with a path $\{X_t\}_{t\geq 0}$ has a transition matrix $P \in \Delta^{n\times(n-1)}$ given by

$$P_{x,y} = \frac{p_{\sigma(x),\sigma(y)}}{|\mathcal{V}_{\sigma(y)}|} \quad \text{for all} \quad x, y \in V. \tag{3}$$

One can understand that we require some restrictions on these constants $\alpha \in \Delta^{K-1}$ and $p \in \Delta^{K\times(K-1)}$. Indeed, we require that the smallest cluster size $\alpha_{\mathsf{min}} \triangleq \min_{k\in\{1,\dots,K\}} \alpha_k$ grows at the same rate as n. Thus as $n \to \infty$, we have that $\alpha_{\mathsf{min}} > 0$. Furthermore, p must satisfy that

$$\exists_{\eta>1} : \max_{i,j,k}\{\frac{p_{j,i}}{p_{k,i}}, \frac{p_{i,j}}{p_{i,k}}\} \leq \eta.$$

These conditions on the parameters of a BMC are necessary for the ability to detect the amount of clusters since they guarantee a minimum level of separability of the parameters.

Assume that the equilibrium distribution of $\{X_t\}_{t\geq 0}$ exists and is denoted by Π_x for all $x \in V$. Note that by symmetry, we have that for any x, y in the same cluster $\mathcal{V}_k$, it holds that $\Pi_x = \Pi_y \triangleq \bar{\Pi}_k$. We also introduce the scaled quantity

$$\pi_k \triangleq \lim_{n\to\infty} \sum_{x\in\mathcal{V}_k} \Pi_x = \lim_{n\to\infty} |\mathcal{V}_k|\bar{\Pi}_k. \quad \text{for all clusters } k \in \{1 \dots, K\} \tag{4}$$

Proposition 1 from [13] guarantees that π satisfies $\pi^T p = \pi^T$. We recall Definition 1 from [13], which states that

Definition 1. *For $\alpha \in \Delta^{K-1}$ and $p \in \Delta^{(K-1)\times K}$, let*

$$I(\alpha, p) \triangleq \min_{a,b} I_{a,b}(\alpha, p) \tag{5}$$

where $I_{a,b}(\alpha, p) \triangleq \left\{\sum_{k=1}^{K} \frac{1}{\alpha_a}\left(\pi_a p_{a,k} \ln \frac{p_{a,k}}{p_{b,k}} + \pi_k p_{k,a} \ln \frac{p_{k,a}\alpha_b}{p_{k,b}\alpha_a}\right) + \left(\frac{\pi_b}{\alpha_b} - \frac{\pi_a}{\alpha_a}\right)\right\}$.

$I(\alpha, p)$ denotes an important information measure which tells us how difficult it is to cluster states in a BMC. For the algorithm which will be provided in this thesis we do not need the exact definition of $I(\alpha, p)$. However, we will use some of the results from [13] which contain this information measure which is one of the reasons we have given the definition of

$I(\alpha, p)$. Another reason for mentioning this information quantity is that it gives us insight in to why and when we are able to detect clusters in a BMC.

Let the sets $\hat{\mathcal{V}}_1, \dots, \hat{\mathcal{V}}_K$ denote an estimated cluster assignment which is the output of any given clustering algorithm. In this thesis we will look at the set of misclassified states $\mathcal{E}$ given by

$$\mathcal{E} \triangleq \bigcup_{k=1}^{\max\{K, \hat{K}^{\mathsf{post}}\}} \hat{\mathcal{V}}_{\gamma^{\mathsf{opt}}(k)} \backslash \mathcal{V}_k \quad \text{where} \quad \gamma^{\mathsf{opt}} \in \underset{\gamma \in \mathsf{Perm}(K)}{\arg\min} \left| \bigcup_{k=1}^{K} \hat{\mathcal{V}}_{\gamma(k)} \backslash \mathcal{V}_k \right| \tag{6}$$

in which $\mathcal{V}_k \triangleq \varnothing$ if $k > K$.

Assume that we are given a sample path of length $T_n = \omega(n)$ depending on the amount of states n of a BMC. Similar to [13] we introduce that an algorithm is (ϵ, c)-locally good at (α, p), if it satisfies $\mathbb{E}\left[|\mathcal{E}|\right] \leq \epsilon$. We can conclude from Theorem 1 of [13] there exists a finite constant $C > 0$ independent of n, such that any clustering algorithm constructed from p and partitions satisfying $||\mathcal{V}_k| - \alpha_k n| \leq 1$ for all k: is not $(\epsilon, 1)$-locally good at (α, p) when

$$\epsilon < Cn \exp\left(-I(\alpha, p)\frac{T_n}{n}(1 + o(1))\right).$$

This information also allows us to get conditions for when asymptotic detection of the clusters is possible. We identify two cases: asymptotic accurate detection which implies $\mathbb{E}_P[|\mathcal{E}|] = o(n)$, and asymptotically exact detection, i.e., $\mathbb{E}_P[|\mathcal{E}|] = o(1)$. For asymptotic accurate detection we need only that $I(\alpha, p) > 0$ and $T_n = \omega(n)$. The conditions for asymptotically exact detection are more difficult to derive. However, a necessary condition for the existence of asymptotically exact algorithms is $I(\alpha, p) > 0$ and $T_n = \omega(n \ln n)$. Another sufficient condition is $T_n = \Theta(n \ln n)$ and $I(\alpha, p) > 1$.

2.2 Spectral Clustering Algorithm

Given a sample path of a BMC denoted by $X_0, X_1, \dots, X_T$, the first part of the Spectral Clustering Algorithm found in [13] generates a rank-K approximation of the matrix $\hat{N}$ given by

$$\hat{N} \triangleq \left(\sum_{t=0}^{T-1} \mathbb{1}\left[X_t = x, X_{t+1} = y\right] \right)_{x,y \in V}.$$

Before one generates a rank-K approximation of $\hat{N}$, the matrix $\hat{N}$ is trimmed by setting both the rows and columns to 0 for the $\lfloor n \exp -\frac{T_n}{n} \ln \frac{T_n}{n} \rfloor$ most visited states, which results in the matrix $\hat{N}_\Gamma$. The reason for trimming is that when our path is relatively short ($T_n = o\left(n \ln n\right)$) the states which are visited unusually often can skew the spectral analysis. After generating the trimmed matrix $\hat{N}_\Gamma$ we are going to make a rank-K approximation of it. We do this by first calculating the Singular Value Decomposition (SVD) of the matrix $\hat{N}_\Gamma$. This gives us the matrices $U, \Sigma, V^T \in \mathbb{R}^{n \times n}$ such that

$$\hat{N}_\Gamma = U \Sigma V^T.$$

With the property that $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$ for some sequence of singular values $\sigma_1, \sigma_2, \ldots, \sigma_n \geq 0$. Without loss of generality we assume that the singular values $\sigma_1, \ldots, \sigma_n$ are in descending order and sort the rows and columns of U, V^T accordingly. After calculating the SVD we can calculate the rank-K approximation of $\hat{N}_\Gamma$ denoted by $\hat{R}$ and defined as:

$$\hat{R} \triangleq \sum_{k=1}^{K} \sigma_k U_{\cdot,k} V_{\cdot,k}^T \,. \tag{7}$$

Algorithm 1: Pseudo-code for the Spectral Clustering Algorithm found in [13]

```
Data: $n$ and a trajectory$X_0, \ldots, X_T$
Result: An approximate cluster assignment
for $x \leftarrow 1$ to $n$ do
    for $y \leftarrow 1$ to $n$ do
        $\hat{N}_{x,y} \leftarrow \sum_{t=0}^{T-1} \mathbb{1}\left[X_t = x, X_{t+1} = y\right]$;
    end
end
Calculate the trimmed matrix $\hat{N}_\Gamma$;
Calculate the Singular Value Decomposition (SVD) $U \Sigma V^T$ of $\hat{N}_\Gamma$;
Order $U, \Sigma, V^T$ such that the singular value $\sigma_1 \geq \ldots \geq \sigma_n \geq 0$ are in descending
  order;
Construct a rank-$K$ approximation $\hat{R} = \sum_{k=1}^{K} \sigma_k U_{\cdot,k} V_{\cdot,k}^T$;
Apply a $K$-means algorithm to $[\hat{R}, \hat{R}^T]$ to determine $\hat{\mathcal{V}}_1, \ldots, \hat{\mathcal{V}}_K$;
return $\left(\hat{\mathcal{V}}_k\right)_{k=1,\ldots,K}$
```

After this step, the researchers in [13] apply a K-means algorithm to the matrix $[\hat{R}, \hat{R}^T]$ to get the estimated clusters $\hat{\mathcal{V}}_1, \ldots, \hat{\mathcal{V}}_K$. The pseudo-code for the Spectral Clustering Algorithm can be found in Algorithm 1. Note that the researchers do provide a detailed K-means algorithm, which will be discussed in Chapter 5. The researchers in [13] proved in Theorem 2 that when $I(\alpha, p) > 0$ and $T_n = \omega(n)$ this Spectral Clustering Algorithm had an performance measure of

$$\frac{|\mathcal{E}|}{n} = O_{\mathbb{P}}\left(\frac{n}{T_n} \ln \frac{T_n}{n}\right) = o_{\mathbb{P}}\left(1\right). \tag{8}$$

Notice that this algorithm achieves asymptotically accurate detection when $I(\alpha, p) > 0$ and $T_n = \omega(n)$. However, as the researchers in [13] point out, it can not be guaranteed that asymptotic exact detection is achieved when using this Spectral Clustering Algorithm .

2.3 Cluster Improvement Algorithm

In order to ensure that the error of the approximate cluster assignments goes to the fundamental limit, the researchers proposed a second algorithm known as the Cluster Improvement Algorithm. This algorithm takes as input, the output of the Spectral Clustering Algorithm (Algorithm 1) and improves the cluster assignment as by maximizing a log-likelihood function. The exact specifics of the Cluster Improvement Algorithm are shown in Algorithm 2. After running the Cluster Improvement Algorithm given by Algorithm 2 t times, the fraction of misclassified states is bounded by

$$\frac{|\mathcal{E}^{[t]}|}{n} = O_{\mathbb{P}}\Big(e^{-t(\ln \frac{T_n}{n} - \ln \ln \frac{T_n}{n})} + e^{-\frac{\alpha_{\text{min}}^2}{720\eta^3\alpha_{\text{max}}^2}\frac{T_n}{n} I(\alpha,p)}\Big) \tag{9}$$

Notice that if $T_n = \omega(n)$, $I(\alpha, p) > 0$, then as $n \to \infty$, $\lim_{n\to\infty} \frac{|\mathcal{E}^{[t]}|}{n} = 0$ since $\lim_{n\to\infty} \ln \frac{T_n}{n} - \ln \ln \frac{T_n}{n} = \infty$. Thus the Cluster Improvement Algorithm achieves asymptotically exact detection under the (nearly tight) sufficient condition $I(\alpha, p) > 0$ and $T_n - \frac{n \ln n}{C \cdot I(\alpha,p)} = \omega(1)$.

We have to make an important remark here. In this thesis our focus is mainly on making the Spectral Clustering Algorithm as given in Algorithm 1 of [13] independent on the knowledge of K. However, when applying the Spectral Clustering Algorithm and Cluster Improvement Algorithm to real data as done in [17], the results one sees are by first using the Spectral Clustering Algorithm and after that the Cluster Improvement Algorithm. Most of the results which are discussed in this thesis come directly from our Spectral Clustering Algorithm with unknown K. The reason this observation is important is that the Cluster Improvement Algorithm can in theory empty one of the clusters estimated by the Spectral Clustering Algorithm with unknown K. This would mean that our estimate of K can still decrease by running the Cluster Improvement Algorithm a sufficient number of times.

Thus, next to $\hat{K}^{\text{pre}}$ and $\hat{K}^{\text{post}}$, there exists a third estimator of K which we will denote by $\hat{K}_t^{\text{CI}}$. This estimator is given by counting the number of nonempty clusters after running the Cluster Improvement Algorithm t times. In theory, this estimator could be lower than our estimator $\hat{K}^{\text{post}}$. However, when testing on both synthetic data and real data, $\hat{K}^{\text{post}}$ and $\hat{K}_t^{\text{CI}}$ tend to coincide which means that we will not focus our attention on $\hat{K}_t^{\text{CI}}$. Later on in thesis we will analyze the performance increase with respect to the ability to cluster, when coupling the Spectral Clustering Algorithm with unknown K with the Cluster Improvement Algorithm over just using the Spectral Clustering Algorithm with unknown K.

2.4 Spectral norm bounds on BMCs

Due to recent advances in creating spectral norm bounds on random matrices associated with Block Markov Chains, some of the bounds and choices we will make in our Spectral Clustering Algorithm with unknown K differ slightly from similar bounds in Algorithm 1 in [13]. One of the pieces of research we would like to highlight in this section is paper

Algorithm 2: Pseudo-code for the Cluster Improvement Algorithm found in [13]

```
Data: An approximate assignment $\hat{\mathcal{V}}_1^{[t]}, \dots, \hat{\mathcal{V}}_{\hat{K}^{\text{post}}}^{[t]}$, and matrix $\hat{N}$
Result: A revised assignment $\hat{\mathcal{V}}_1^{[t+1]}, \dots, \hat{\mathcal{V}}_{\hat{K}^{\text{post}}}^{[t+1]}$
$n \leftarrow \text{dim}(\hat{N})$, $\mathcal{V} \leftarrow \{1, \dots, n\}$, $T \leftarrow \sum_{x \in \mathcal{V}} \sum_{y \in \mathcal{V}} \hat{N}_{x,y}$;
for $a \leftarrow 1$ to $K$ do
    $\hat{\pi}_a \leftarrow \hat{N}_{\hat{\mathcal{V}}_a^{[t]}, \mathcal{V}}/T$, $\hat{\alpha}_a \leftarrow \left|\hat{\mathcal{V}}_a^{[t]}\right|/n$, $\hat{\mathcal{V}}_a^{[t+1]} \leftarrow \varnothing$;
    for $b \leftarrow 1$ to $K$ do
        $\hat{p}_{a,b} \leftarrow \hat{N}_{\hat{\mathcal{V}}_a^{[t]}, \hat{\mathcal{V}}_b^{[t]}}/\hat{N}_{\hat{\mathcal{V}}_a^{[t]}, \mathcal{V}}$;
    end
end
for $x \leftarrow 1$ to $n$ do
    $c_x^{\text{opt}} \leftarrow \arg\max_{c=1,\dots,K} \left\{\sum_{k=1}^{K} \left(\hat{N}_{x, \hat{\mathcal{V}}_k^{[t]}} \ln \hat{p}_{c,k} + \hat{N}_{\hat{\mathcal{V}}_k^{[t]}, x} \ln \frac{\hat{p}_{k,c}}{\hat{\alpha}_c}\right) - \frac{T_n}{n} \cdot \frac{\hat{\pi}_x}{\hat{\alpha}_c}\right\}$;
    $\hat{\mathcal{V}}_{c_x^{\text{opt}}}^{[t]} \leftarrow \hat{\mathcal{V}}_{c_x^{\text{opt}}}^{[t+1]} \cup \{x\}$;
end
return $\left(\hat{\mathcal{V}}_k^{[t+1]}\right)_{k=1,\dots,\hat{K}^{\text{post}}}$
```

[14]. This paper contains a couple of important theorems which will be used throughout this bachelor thesis.

Firstly, recall the definition of $\hat{N}$ given in Section 2.2. In [13] this matrix is trimmed by removing the $\left\lfloor n \exp\left(-\frac{T_n}{n} \ln \frac{T_n}{n}\right)\right\rfloor$ states with the highest amount of visits. Assume instead that we remove the $\left\lfloor n \exp\left(-\frac{T_n}{n}\right)\right\rfloor$ states with the most of amount of visits. Then Corollary 4 of [14] guarantees that when $\omega(n) = T_n = o(n^2)$, the ith singular value of the $\hat{N}_\Gamma$ satisfies

$$\sigma_i(\hat{N}_\Gamma) = \begin{cases} \Theta_{\mathbb{P}}\left(\frac{T_n}{n}\right) & i \leq K, \\ O_{\mathbb{P}}\left(\sqrt{\frac{T_n}{n}}\right) & i > K. \end{cases} \tag{10}$$

3 Spectral Clustering Algorithm with unknown K

Now when one looks at Section 2.2 we notice that Algorithm 1 depends on knowing the number of clusters K up front. However, one can argue that this is not a very realistic assumption. Hence we would like to make Algorithm 1 independent of the knowledge of the number of clusters K [1].

[1] We highlight a difference between this current thesis and the algorithms presented in [13]. That is; our transition matrix allows for self transitions, hence the probability of moving from state i to state i is

3.1 Trimming and rank-$\hat{K}^{\mathsf{pre}}$ approximation

Our Spectral Clustering Algorithm with unknown K will consist of a similar setup as the Spectral Clustering Algorithm. We will again start by calculating $\hat{N}$ as defined in Section 2.2. Since we would like to use (10) we will have to trim the matrix identically. So let $\Gamma^c \subset V = \{1, \ldots, n\}$ be the set consisting of the $\lfloor n \exp -\frac{T_n}{n} \rfloor$ states which have the highest amount of visits, and define $\hat{N}_\Gamma$ element-wise as

$$(\hat{N}_\Gamma)_{x,y} = \begin{cases} \hat{N}_{x,y} & \text{if } x, y \in \Gamma, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

After trimming $\hat{N}$, we make a rank-K approximation of $\hat{N}_\Gamma$. But because K is unknown to us, we instead construct some rank-$\hat{K}^{\mathsf{pre}}$ approximation of $\hat{N}_\Gamma$. This is done using singular value thresholding. Assume that we have our singular value decomposition of $\hat{N}_\Gamma$ given by the matrices $U\Sigma V^T$, and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$ are the singular values in descreasing order. Define

$$\hat{K}^{\mathsf{pre}} \triangleq \max\left\{ k : \sigma_k \geq \sqrt{\frac{T_n}{n} \ln \frac{T_n}{n}} \right\}$$

which implies that $\hat{K}^{\mathsf{pre}}$ is the number of singular values greater than $\sqrt{\frac{T_n}{n} \ln \frac{T_n}{n}}$.

After we have our estimator $\hat{K}^{\mathsf{pre}}$ one can make the $\hat{K}^{\mathsf{pre}}$-approximation of $\hat{N}_\Gamma$. This is defined as

$$\hat{R} \triangleq \sum_{k=1}^{\hat{K}^{\mathsf{pre}}} \sigma_k U_{\cdot,k} V_{\cdot,k}^T \tag{12}$$

This can be interpreted as keeping the important information of $\hat{N}_\Gamma$ which is attributed to having a high singular value and removing the noise which is attributed having a low singular value. One can also raise the fair question as to why we don't use $\hat{K}^{\mathsf{pre}}$ as an estimator for K. As our results will show, the gap between the Kth and $(K+1)$th singular value is difficult to consistently and precisely predict. Hence, in turn, $\hat{K}^{\mathsf{pre}}$ is not a very consistent or precise estimator.

3.2 The K-means algorithm

Algorithm 1 states that one can use an arbitrary K-means algorithm applied to the matrix $\hat{R}^0$ to determine the estimated clusters $\hat{\mathcal{V}}_1, \ldots, \hat{\mathcal{V}}_K$. However, since K is unknown to us, this will require a different method.

One could propose that since $\hat{K}^{\mathsf{pre}}$ is an estimator of K, we could use a K-means algorithm applied to $\hat{R}^0$ to generate the clusters $\hat{\mathcal{V}}_1, \ldots, \hat{\mathcal{V}}_{\hat{K}^{\mathsf{pre}}}$. As mentioned before and

not necessarily 0. In [13] self transitions are not allowed and thus the diagonal of the transition matrix is always 0. However, this difference is inconsequential to working of the presented algorithms.

as we will discuss in the results, $\hat{K}^{\text{pre}}$ is not a good estimator of K. Hence in turn, the performance of our Spectral Clustering Algorithm with unknown K would then be undermined. Therefore, we have come up with a different way to estimate K and to generate the estimated clusters $\hat{\mathcal{V}}_1, \ldots, \hat{\mathcal{V}}_{\hat{K}^{\text{post}}}$, in which $\hat{K}^{\text{post}}$ will be our estimator of K.

The first step in clustering the states is by calculating the neighborhood $\mathcal{N}_x$ of every state $x \in V$. This can be interpreted as wanting to find for each state the states that are "close" to this state. This neighborhood is defined as:

$$\mathcal{N}_x \triangleq \left\{ y \in V \,\middle|\, \sqrt{\|\hat{R}_{x,\cdot} - \hat{R}_{y,\cdot}\|_2^2 + \|\hat{R}_{\cdot,x} - \hat{R}_{\cdot,y}\|_2^2} \leq \sqrt{\frac{T^2}{n^3 \ln \frac{T_n}{n}}} \right\}.$$

After the neighborhoods for each state $x \in V$ have been calculated we can sequentially select the clusters. We do this in the following way: first we determine the centers $z_1^*, z_2^*, \ldots \in V$ of the clusters, secondly we generate the estimated clusters $\hat{\mathcal{V}}_1, \hat{\mathcal{V}}_2, \ldots$. However, since K is unknown to the algorithm we have to know when to stop this recursive generation of clusters. Therefore we iterate until the cardinality of the ith cluster denoted by ρ becomes "too small": that is we iterate for as long as $\rho \geq \frac{n^2 \ln \frac{T_n}{N}}{T_n}$.

$$\hat{\mathcal{V}}_k \leftarrow \mathcal{N}_{z_k^*} \backslash \left\{ \bigcup_{l=1}^{k-1} \hat{\mathcal{V}}_l \right\} \quad \text{where} \quad z_k^* \leftarrow \underset{x \in V}{\arg\max} \left\{ \mathcal{N}_x \backslash \bigcup_{l=1}^{k-1} \hat{\mathcal{V}}_l \right\} \tag{13}$$

$$\rho \leftarrow |\hat{\mathcal{V}}_k| \tag{14}$$

After this we conclude that $\hat{K}^{\text{post}} = k - 1$, and this will be our estimator for K. To finish the process, the states which have not been assigned to the first $\hat{K}^{\text{post}}$ clusters are now assigned to the cluster closest to it. Hence for all $x \in \left\{ V \backslash \left(\bigcup_{k=1}^{\hat{K}^{\text{post}}} \hat{\mathcal{V}}_k \right) \right\}$

$$\hat{\mathcal{V}}_{k_*} \leftarrow \hat{\mathcal{V}}_{k_*} \cup \{x\} \;\text{ where }\; k_* \leftarrow \underset{k \in 1, \ldots, \hat{K}^{\text{post}}}{\arg\min} \left\{ \sqrt{\|\hat{R}_{x,\cdot} - \hat{R}_{z_k^*,\cdot}\|_2^2 + \|\hat{R}_{\cdot,x} - \hat{R}_{\cdot,z_k^*}\|_2^2} \right\}. \tag{15}$$

From this we obtain our estimated clusters $\hat{\mathcal{V}}_1, \ldots, \hat{\mathcal{V}}_{\hat{K}^{\text{post}}}$. For our revised Spectral Clustering Algorithm for *unknown* in Algorithm 3, Theorem 1 gives an upper bound on the number of misclassified states after executing this algorithm.

Theorem 1. *Assume that $\frac{T_n}{n} = \omega(n)$ and $I(\alpha, p) > 0$. Then after running Algorithm 3, $\hat{K}^{\text{post}} = K$ with high probability, and the proportion of misclassified states satisfies:*

$$\frac{|\mathcal{E}|}{n} = O_{\mathbb{P}}\left(\frac{n}{T_n}\right) = o_{\mathbb{P}}(1).$$

Algorithm 3: Pseudo-code for the Spectral Clustering Algorithm with unknown K

```
Data: $n$ and a trajectory $X_0, \dots, X_T$
Result: An approximate cluster assignment
Perform lines $1 - 8$ of the Spectral Clustering Algorithm 1 (with
  $|\Gamma^c| = \lfloor n \exp\left(-\frac{T_n}{n}\right) \rfloor$);
$\hat{K}^{\text{pre}} \leftarrow \max\left\{k : \sigma_k \geq \sqrt{\frac{T_n}{n} \ln \frac{T_n}{n}}\right\}$;
Construct a rank-$\hat{K}^{\text{pre}}$ approximation;
$\hat{R} = \sum_{k=1}^{\hat{K}^{\text{pre}}} \sigma_k \hat{U}_{\cdot,k} \hat{V}_{\cdot,k}^T$;
for $x \leftarrow 1$ to $n$ do
    $\mathcal{N}_x \leftarrow \left\{ y \in V \,\middle|\, \sqrt{\|\hat{R}_{x,\cdot} - \hat{R}_{y,\cdot}\|_2^2 + \|\hat{R}_{\cdot,x} - \hat{R}_{\cdot,y}\|_2^2} \leq \sqrt{\frac{T_n^2}{n^3 \ln \frac{T_n}{n}}} \right\}$;
end
$\hat{\mathcal{V}}_1 \leftarrow \varnothing, \quad k \leftarrow 0, \quad \rho \leftarrow |\Gamma|$;
while $\rho \geq \frac{n^2 \ln \frac{T_n}{n}}{T_n}$ do
    $k \leftarrow k + 1, \quad z_k^* \leftarrow \arg\max_{x \in V} \{\mathcal{N}_x \setminus \bigcup_{l=1}^{k-1} \hat{\mathcal{V}}_l\}$;
    $\hat{\mathcal{V}}_k \leftarrow \mathcal{N}_{z_k^*} \setminus \bigcup_{l=1}^{k-1} \hat{\mathcal{V}}_l, \quad \rho \leftarrow |\hat{\mathcal{V}}_k|$;
end
$\hat{K}^{\text{post}} \leftarrow k - 1$;
for $x \in \Gamma \setminus \bigcup_{l=1}^{\hat{K}^{\text{post}}} \hat{\mathcal{V}}_l$ do
    $k_* \leftarrow \arg\min_{k \in 1,\dots,\hat{K}^{\text{post}}} \left\{ \sqrt{\|\hat{R}_{x,\cdot} - \hat{R}_{z_k^*,\cdot}\|_2^2 + \|\hat{R}_{\cdot,x} - \hat{R}_{\cdot,z_k^*}\|_2^2} \right\}$;
    $\hat{\mathcal{V}}_{k_*} \leftarrow \hat{\mathcal{V}}_{k_*} \cup \{x\}$;
end
return $\left(\hat{\mathcal{V}}_k\right)_{k=1,\dots,\hat{K}^{\text{post}}}$
```

3.3 Perturbed BMCs

Real data often doesn't perfectly follow a BMC and is often of a higher degree. In order to study this we look at objects which consist of a large part of a rank-K object mixed with a rank-n object. Let P_{BMC} be a transition matrix of a BMC of rank K and let Λ with rank n be a transition matrix of some first order Markov chain. We introduce the concept of a *pertubed BMC* which mixes a BMC and a generic first-order Markov chain on the state space $\Omega = \{1, \dots, n\}$ such that the transition matrix of the pertubed BMC denoted by P_{Pertubed} is given by

$$P_{\text{Pertubed}} \triangleq (1 - \epsilon) P_{\text{BMC}} + \epsilon \Lambda. \tag{16}$$

The parameter $\epsilon \in [0, 1]$ influences how much influence the non-BMC part has on the

BMC. By mixing a rank-K and a rank-n object we hope to see how robust our algorithm is against perturbation in the data by taking different values of ϵ.

There are different ways to find a rank n matrix Λ which are laid out in Supplement 12 of [17]. We choose to generate Λ in an uniform stochastic way. This mean that we will sample each row independently from a Dirichlet$(1/n, \dots, 1/n)$ distribution.

3.4 Sequences of codons

Every cell in the human body has DNA. This DNA encodes all information about a person. Now DNA is build up by four main ingredients called nucleotides which are:

1. Adenine (A).

2. Cytosine (C).

3. Thymine (T).

4. Guanine (G).

Certain sequences of these ingredients are what we call genes and are subsequences of your total DNA sequence. A gene might encode what your physical characteristics are. Think in this case of the color of your hair, color of your eyes or the way your hair curls. We will focus our attention on the OCA2 gene in the human DNA. It is believed that this gene plays a part in controlling the skin color variation and the determines if a color of a person's eye is brown or blue. We must note that our Spectral Clustering Algorithm with unknown K should work on any gene and produce similar results.

As mentioned before a string of DNA is a sequence of four ingredients denoted by A, C, T, G. A sequence of three of these letters together form a codon. A codon encodes a specific amino acid which used to copy parts of the DNA within our cells. If we have a sequence of nucleotides say

$$ATC\,CGA\,AAA\,CTG\,AGT\,CCT\,TGA\,ATA\,AGT\dots\text{et cetera},$$

we can transform this sequence nucleotide in to a chain of codons such as

$$X_0 = ATC, X_1 = CGA, X_2 = AAA, X_3 = CTG, X_4 = AGT\dots\text{et cetera}.$$

Assuming no restrictions on these codons we can see that there are $n = 4^3 = 64$ possible combinations of codons.

4 Main results

In order to test our algorithm on synthetic data from a BMC we took values for $\alpha \in \Delta^{K-1}$ and $p \in \Delta^{K\times(K-1)}$. Based on these two constants the whole transition matrix is defined as done in (3) and thus a synthetic sample path can be generated. Hence for all these tests we thus fix these constants and take a path of length T_n. After defining T_n we can generate a sample path adhering to the transition matrix and path length. After generating this sample path, our algorithm tries to find K by using the Algorithm 3. The results of the tests will be discussed in this chapter.

4.1 Verification of Algorithm 3 using singular values

4.1.1 Analysis of the first K singular values

In [14] it is proposed that each kth singular with $k \geq K+1$ value of $\hat{N}_\Gamma$ is of order $O_{\mathbb{P}}\left(\sqrt{\frac{T_n}{n}}\right)$, and that the first K singular values are of order $\Theta_{\mathbb{P}}\left(\frac{T_n}{n}\right)$. Since our algorithm calculates the singular value decomposition (SVD) of $\hat{N}_\Gamma$, it gives us a way to test if this hypothesis seems to align with our implementation of the Spectral Clustering Algorithm with unknown K. Thus we calculate the SVD of $\hat{N}_\Gamma$ and then divide the singular values by the order they should belong to as proposed in Corollary 4 of [14]. We will refer to these singular values divided by their theoretical order as normalized singular values. An important thing to notice is that the theoretical order depends at which eigenvalue one is looking at. The first K normalized eigenvalues are defined as $\frac{\sigma_i}{T_n/n}$ while the the rest of the $n-K$ eigenvalues are defined as $\frac{\sigma_i}{\sqrt{T_n/n}}$.

Firstly, we will analyze the case for the first K singular values. In Figure 3 one can see the results of our synthetic data testing. We will analyze the case that T_n is sparse. Recall that this implies $\omega(n) = T_n = o(n \ln n)$. This case is tested in $T_n = n(\ln n)^{1/2}$. Notice that as n grows the normalized singular values also seem to grow. However, by looking at the distance between the different lines representing the states, it looks like their growth is slowing down as n grows which could imply asymptotic growth of the singular values. This is in accordance with the theory covered in Section 2.4 which states that the be of $\Theta(\frac{T_n}{n})$. Thus this implies that the normalized eigenvalues should be bounded from above by some $k_2 \in \mathbb{R}$ such that $\frac{\sigma_i}{T_n/n} \leq k_2$. As one can observe in the case that T_n is sparse in Figure 3, this seems to be the case since the size of the singular values seems to grow asymptotically.

Next we will analyze the case that T_n is dense. Thus recall that this implies that $\omega(n \ln n) = T_n$. We will try to apply the same theory as we did in the case that T_n is sparse. For the cases that T_n is dense we look at the cases that $T_n = n(\ln n)^{3/2}$, $T_n = \frac{n^2}{\ln n}$ and $T_n = n^2$. We observe as n grows the size of the normalized eigenvalues seems to go down. However, it looks like this happens in an asymptotic way since the lines representing different values of n as getting closer together. We conclude that when T_n is dense then

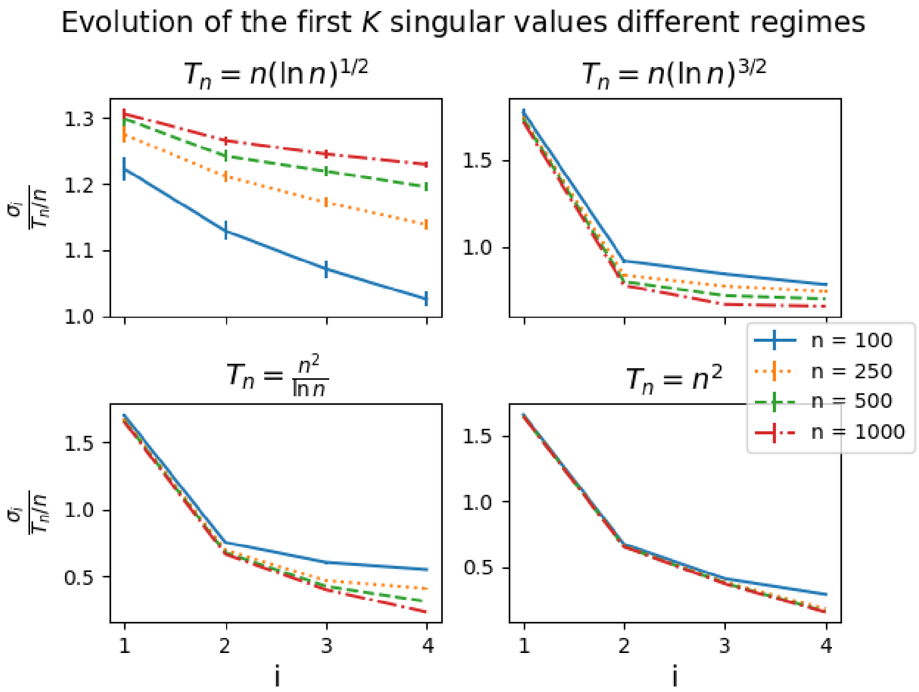

Figure 3: Singular values of $\hat{N}_\Gamma$ in different regimes of T. Taking $\alpha = \begin{pmatrix}2/10 & 1/10 & 5/10 & 2/10\end{pmatrix}$ and probabilities of moving between the clusters given by

$$(p_{i,j})_{i,j\in\{1,2,3,4\}} = \begin{pmatrix} 0.1 & 0.2 & 0.1 & 0.6 \\ 0.4 & 0.3 & 0.1 & 0.2 \\ 0.2 & 0.1 & 0.4 & 0.3 \\ 0.2 & 0.5 & 0.3 & 0 \end{pmatrix}$$

there exist $k_1, k_2 \in \mathbb{R}$ such that $k_1 \leq \frac{\sigma_i}{T_n/n}$. When combining both regimes of T_n we can conclude that the theory discussed in Chapter 2.4 is satisfied.

Another thing one can notice is that as $T_n = \omega(n \ln n)$ and as T_n grows with respect to n, the convergence of the normalized eigenvalues seem to go faster. This can be observed by comparing the cases that $T_n = n(\ln n)^{3/2}$, $T_n = \frac{n^2}{\ln n}$ and $T_n = n^2$. As T_n gets bigger with respect to n the lines representing the different states are closer to each other. A good explanation for this is that as T_n grows with respect to n, the matrix $\hat{N}_\Gamma$ contains less noise which in turn implies that one would expect that the difference between the first K singular values and the $n - K$ singular values after those is more pronounced. In turn, the normalized singular values are close to their theoretical limit if n is not that big. Hence, the convergence of $\frac{\sigma_i}{T_n/n}$ as n grows happens faster for bigger T_n.

4.1.2 Analysis of the next $n - K$ singular values

In the previous Section we looked at the first K singular values. Thus we will now look at the remaining $n - K$ singular values. We know from the theory discussed in Chapter 2.4 that we have that $\sigma_i = O_{\mathbb{P}}\left(\sqrt{\frac{T_n}{n}}\right)$. Thus this implies that the normalized singular values should be bounded from above by some constant $M \in \mathbb{R}$ as $n \to \infty$. Hence if we look at our results we hope to see that the size of the remaining $n - K$ singular values seems to convergence to some upper bound. We will do the analysis in a similar fashion as before. First we analyze the case that T_n is sparse and then the case that T_n is dense. Secondly, we will study the influence the size of T_n with respect n has on the size of the normalized singular values.

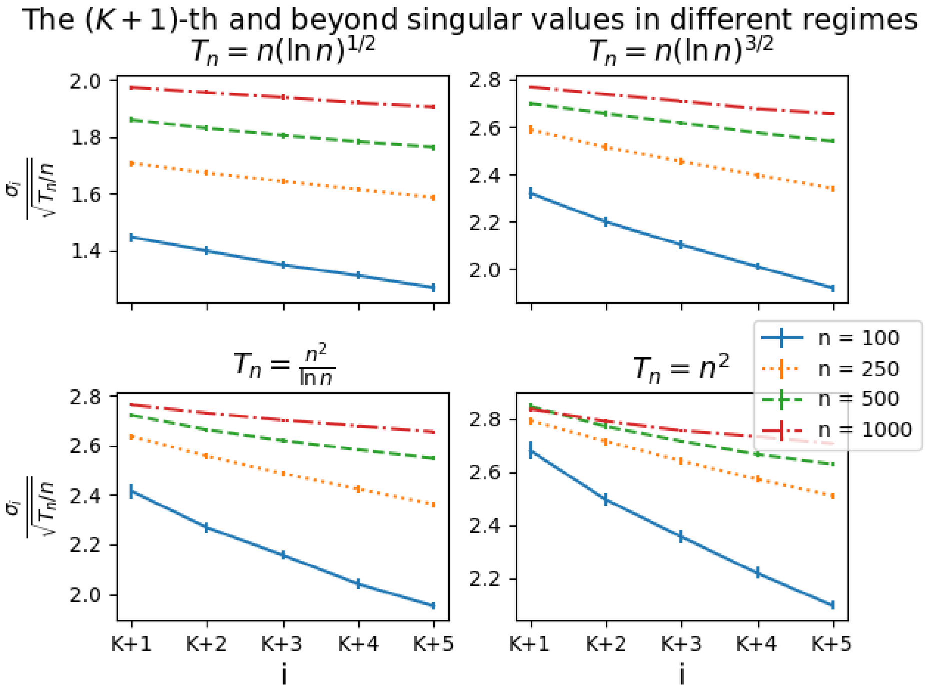

Figure 4: Singular values of $\hat{N}_\Gamma$ in different regimes of T. Taking $\alpha = \begin{pmatrix} 2/10 & 1/10 & 5/10 & 2/10 \end{pmatrix}$ and probabilities of moving between the clusters given by

$$(p_{i,j})_{i,j\in\{1,2,3,4\}} = \begin{pmatrix} 0.1 & 0.2 & 0.1 & 0.6 \\ 0.4 & 0.3 & 0.1 & 0.2 \\ 0.2 & 0.1 & 0.4 & 0.3 \\ 0.2 & 0.5 & 0.3 & 0 \end{pmatrix}$$

We analyze the case that T_n is sparse, thus in Figure 4 we look at the case that $T_n = n(\ln n)^{1/2}$. Now notice that as n grows the size of the normalized singular values seems to go up. However, the size of the gaps between the different lines seems to decrease.

Thus this could imply that the size of the normalized eigenvalues is asymptotically increasing. Hence this would mean that the size of the normalized eigenvalues is bounded from above by some constant $M \in \mathbb{R}$. This would be in accordance with our theory and gives the results of the simulation some credibility.

Next we will analyze the case that T_n is dense, meaning that in Figure 4 we look at the cases that $T_n = n(\ln n)^{3/2}$, $T_n = \frac{n^2}{\ln n}$ and $T_n = n^2$. We see a similar pattern as in the case that T_n is sparse. The distance between the lines seems to converge as n grows which could imply that the size of the normalized eigenvalues is asymptotically increasing. Thus this implies that the size of the normalized eigenvalues as $n \to \infty$ is bounded from above which satisfies the theory discussed in Chapter 2.4.

Lastly, we will look at the influence of the size of T_n with respect to the size of n on the rate of convergence of this asymptotic growth of the size of the normalized eigenvalue. Now when looking at all four cases of T_n notice that the distance between the lines gets smaller as T_n gets larger with respect to n. Furthermore, observe that the size of the normalized eigenvalues is also higher in the case that T_n is large with respect to n. Thus this suggests that the rate of convergence to the theoretical limit is faster as T_n gets bigger with respect to n. The explanation for this is similar to the explanation for why the first K singular values seem to convergence faster to their theoretical limit as T_n gets bigger with respect to n. If T_n is large with respect to n then $\hat{N}_\Gamma$ contains less noise then if T_n is small with respect to n. Thus the difference between the first K singular values and the rest of the $n - K$ singular values should be more pronounced when T_n is large with respect to n. Thus implying that the singular values convergence faster to their theoretical limit for bigger T_n.

4.2 Estimates for K

We will analyze the performance of our model in estimating the number of clusters K. Recall that the Spectral Clustering Algorithm with unknown K gives two estimates for K, denoted by $\hat{K}^{\text{pre}}$ and $\hat{K}^{\text{post}}$. We claimed that $\hat{K}^{\text{post}}$ was a more consistent and precise estimator for K and thus we use $\hat{K}^{\text{post}}$ as our estimator for K. However, since this algorithm also calculates $\hat{K}^{\text{pre}}$ we can still analyze the behavior of this estimator as well. Similar to the verification of the singular values we have taken different instances of n and T_n to analyze their influence on the estimates of K. First we will analyze the performance of our estimator $\hat{K}^{\text{post}}$. After this analysis, we will analyze the performance of the $\hat{K}^{\text{pre}}$ estimator and compare the two. From this we will also obtain evidence why $\hat{K}^{\text{post}}$ is a better estimator of K than $\hat{K}^{\text{post}}$.

4.2.1 Performance of the $\hat{K}^{\text{post}}$ estimator

In the left graph of Figure 5 we see that for the performance of $\hat{K}^{\text{post}}$ gets better as n grows. However, we also see that T_n needs to be quite large with respect to n in order for $\hat{K}^{\text{post}}$ to be able to estimate K accurately. This could also mean that for the cases where T_n is not that big with respect to n ($T_n = n(\ln n)^{1/2}$ and $T_n = n(\ln n)^{3/2}$) we need to run

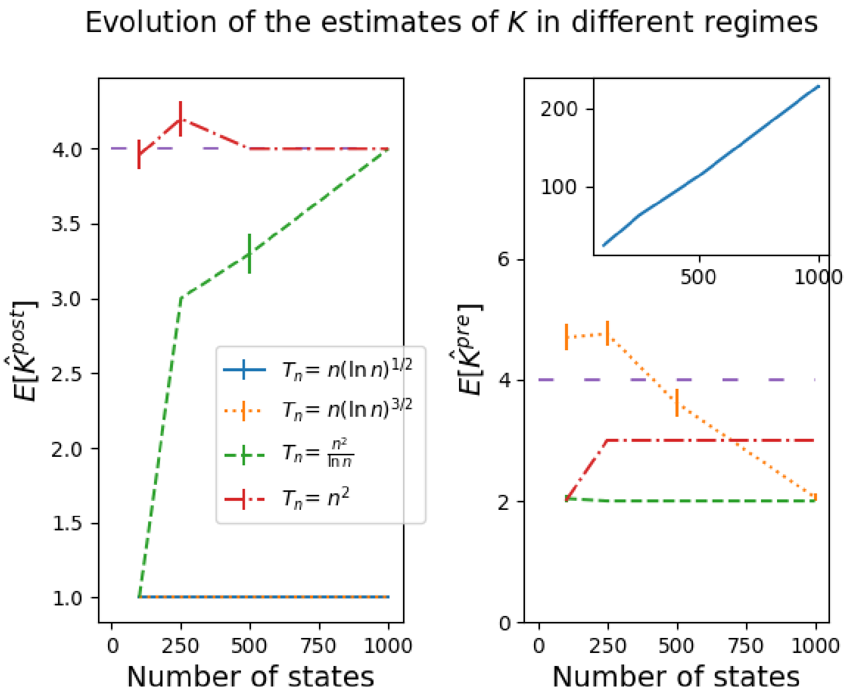

Figure 5: Singular values of $\hat{N}_\Gamma$ in different regimes of T. Taking $\alpha = \begin{pmatrix} 2/10 & 1/10 & 5/10 & 2/10 \end{pmatrix}$ and probabilities of moving between the clusters given by

$$(p_{i,j})_{i,j\in\{1,2,3,4\}} = \begin{pmatrix} 0.1 & 0.2 & 0.1 & 0.6 \\ 0.4 & 0.3 & 0.1 & 0.2 \\ 0.2 & 0.1 & 0.4 & 0.3 \\ 0.2 & 0.5 & 0.3 & 0 \end{pmatrix}$$

the simulation for bigger n. Since for the case that $T_n = \frac{n^2}{\ln n}$ it starts off by not being able to estimate K accurately however, as n grows the performance of the model increases to the correct number of clusters in the system.

4.2.2 Performance of the $\hat{K}^{\text{pre}}$ estimator

One observes by looking at the right graph in Figure 5 that if T_n is sparse that the estimate for $\hat{K}^{\text{pre}}$ grows as n grows large. A possible reason is that in the case that T_n is sparse, the matrix $\hat{N}_\Gamma$ contains a lot of noise. Most of this noise should be filtered out by the trimming of this matrix, however our estimate for $\hat{K}^{\text{pre}}$ is still not accurate in the case that T_n is sparse. Furthermore, one observes that if T_n is dense that the estimates for $\hat{K}^{\text{pre}}$ are close to K but quite often still off by 1 or 2. As previously discussed the exact value of the size of the ith singular value is difficult to predict. However, if T_n and n are very large then our estimates for the size of the ith singular value are closer to the observed value than in the case that either T_n is small with respect to n or n is not that big. This leads us on to discussion which off the two estimators is a better estimator. From Figure 5 it is quite clear that $\hat{K}^{\textsf{post}}$ is a better estimator for K than $\hat{K}^{\textsf{pre}}$. This is because $\hat{K}^{\textsf{post}}$ approaches the real value of K faster than compared to the $\hat{K}^{\textsf{pre}}$ and if T_n is sparse $\hat{K}^{\textsf{post}}$ does not blow up while $\hat{K}^{\textsf{pre}}$ does.

4.3 Perturbed BMCs

In Section 3.3 the notion of a perturbed BMC was introduced to see how our algorithm performs when the data is not a true BMC. In this section we will analyze the bahviour of the Spectral Clustering Algorithm with unknown K on synthetic data from a perturbed BMC. We will look at the performance of our estimated number of clusters and the fraction of misclassified states. The testing of the fraction of misclassified states is split in two cases. In the first case we assume that K is unknown and thus the pseudocode of Algorithm 3 is still valid. In the second case we will look at when Algorithm 3 knows K a-priori. Hence line 2 of Algorithm 3 becomes $\hat{K}^{\textsf{pre}} = K$ and line 7 changes in to a for loop ranging from $k = 1$ to $k = K + 1$.

4.3.1 Estimated number of clusters in a perturbed BMC

Figure 6 suggests that our estimated number of clusters $\hat{K}^{\textsf{post}}$ has a small rise around $\epsilon = 0.2$ before it then goes down to 1. The reason for this is as epsilon increases the noise in the matrix $\hat{N}_\Gamma$ increases as well. This will lead to states becoming seperated as epsilon increases. For small perturbation around $\epsilon = 0.2$ the clusters will therefore break in to smaller clusters which increases $\hat{K}^{\textsf{post}}$. However, if epsilon gets too large (in this case $\epsilon > 0.25$) then the states are so far apart from each other that the neighbourhood of each state only contain the state itself. Thus there is no neighbourhood that satisfies the requirement of being larger then $\frac{n^2 \ln \frac{T_n}{n}}{T_n}$. Hence the estimated number of clusters goes down to 1.

Now when looking at the performance of $\hat{K}^{\textsf{pre}}$, one can observe that as ϵ increases $\hat{K}^{\textsf{pre}}$ increases as well. The explanation for this is quite straightforward. As ϵ increases the transition matrix becomes more of a rank n matrix and less of a rank K matrix, implying that we would expect that $\hat{N}_\Gamma$ becomes more of a rank n matrix. Thus $\hat{K}^{\textsf{pre}}$ which estimates the rank of $\hat{N}_\Gamma$ will also increase to n. This is also the behavior we see in Figure 6.

4.3.2 Fraction of misclassified states in a parturbed BMC

In Figure 7 we observe the performance of the ability to correctly cluster states in the Spectral Clustering Algorithm with unknown K. We will first look at the performance when K is unknown to our model denoted by the Not-fixed label. We notice that there is a spike is the number of misclassifications around $\epsilon = 0.2$ after which the fraction of misclassified states decreases to α_2. It might be strange to notice that the fraction of the number of

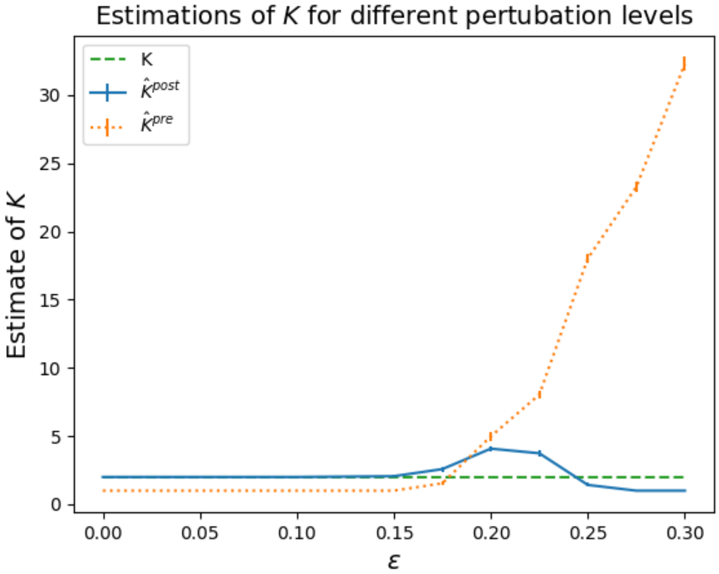

Figure 6: The performance of our estimations of the underlying amount of clusters K. The parameters of the BMC with $K = 2$ are $p_{1,1} = 0.6 = 1 - p_{1,2}$ and $p_{2,1} = 0.3 = 1 - p_{2,2}$ and $\alpha = (3/5 \quad 2/5)$. Furthermore, we took $n = 250$ and $T_n = n^2$.

misclassifications is higher than α_2 since if one would estimate that every state was in the first cluster, then the fraction of misclassifications is α_2. However, if we recall that in the previous section when looking at the performance of $\hat{K}^{\text{post}}$, we observed that the estimated number of clusters was significantly higher than K around $\epsilon = 0.2$. We can conclude that in the case around $\epsilon = 0.2$, Algorithm 3 predicts a lot of clusters which are of small size. If we assume that $\hat{\mathcal{V}}_k$ are ordered in a descending way based on their cardinality, then the number of misclassified states satisfies

$$|\mathcal{E}| \geq \left| \bigcup_{i=K+1}^{\hat{K}^{\text{post}}} \hat{\mathcal{V}}_i \right| .$$

This explains why our fraction of misclassified states can be worse than saying all states belong to cluster 1.

When looking at the performance with K known to the algorithm, the algorithm performs as expected. As the perturbation level increases the noise in the $\hat{N}_\Gamma$ from the rank n perturbation transition matrix becomes greater. Thus it becomes more and more difficult to cluster states correctly and thus the fraction of misclassified states also increases. Now this conclusion was also obtained in Figure 2a of [17]. They derived in a similar fashion that the performance of their algorithms also worsened as the perturbation level ϵ grew.

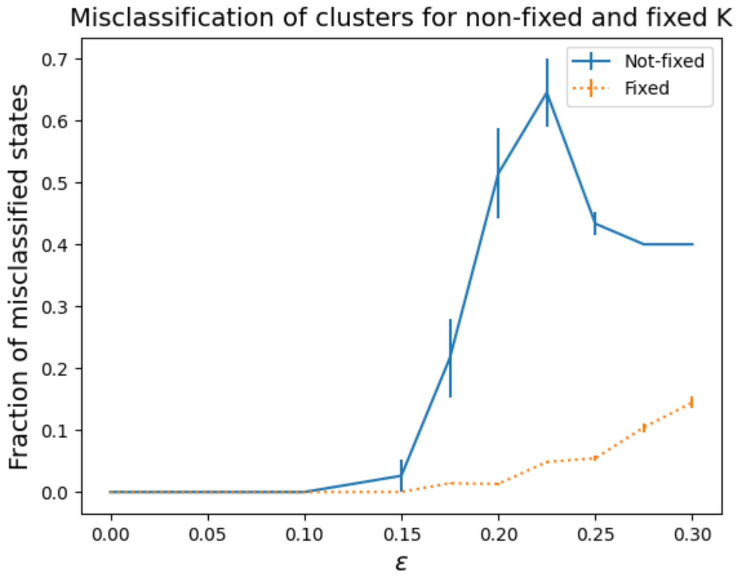

Figure 7: The performance of our algorithm in clustering correctly in the case that the algorithm does not know K and when the algorithm does know K. The parameters of the BMC with $K = 2$ are $p_{1,1} = 06 = 1 - p_{1,2}$ and $p_{2,1} = 0.3 = 1 - p_{2,2}$ and $\alpha = (3/5 \quad 2/5)$. Furthermore, we took $n = 250$ and $T_n = n^2$.

4.4 The influence of the path length on the performance of our algorithm

In this section we will analyze the behavior of our Spectral Clustering Algorithm with unknown K in the setting of perturbed BMC. The main goal of this section is to understand how the path length influences the outcome of the algorithm. Similar to the previous section we will first try to understand the influence the path length has on the estimators $\hat{K}^{\text{pre}}$ and $\hat{K}^{\text{post}}$. Secondly, we will try to understand how the path length influences the fraction of misclassified states. In a true BMC we would expect if n is fixed and as T_n grows, that for any pair of states $x, y \in V$ the empirical transition probability $\frac{\hat{N}_{x,y}}{\sum_{i=1}^{n} \hat{N}_{x,i}}$ approaches the fundamental transition probability $(P_{\text{BMC}})_{x,y}$. Thus we expect that clustering becomes easier and our estimators $\hat{K}^{\text{pre}}$ and $\hat{K}^{\text{post}}$ should converge to K and the fraction of misclassified states should go to zero as our path length increases.

4.4.1 Estimated number of clusters in a perturbed BMC

We start off by looking at the influence the path length has on the estimations of K. We expect that the longer the sample path of a perturbed BMC with an $\epsilon > 0$ is, the more the Spectral Clustering Algorithm with unknown K should pick up the signal of the rank n

transition matrix.

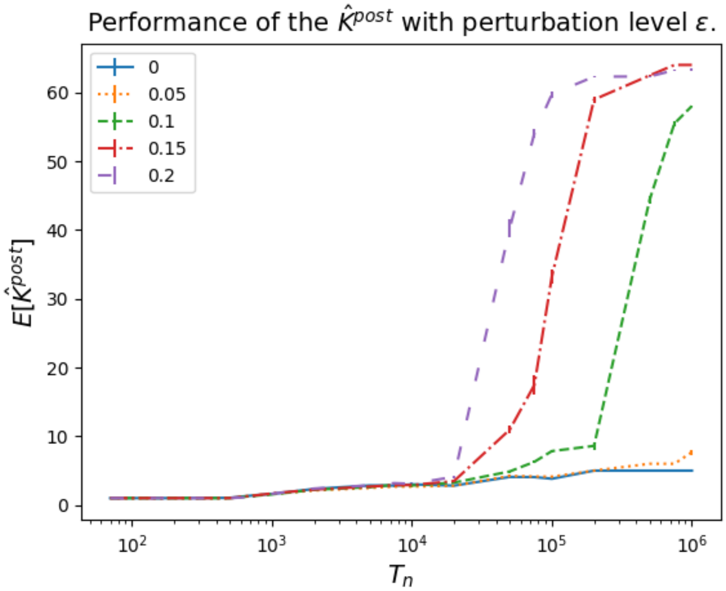

Figure 8: The influence of the length of the path on the accuracy of $\hat{K}^{\text{post}}$ to predict K. The parameters of the BMC is $K = 5$ and α and the cluster transition matrix p are given in Appendix 7.1.

First, we analyze the behavior of $\hat{K}^{\text{post}}$. Figure 8 suggests that as ϵ gets larger the shorter the sample path needs to be for the estimated amount of clusters to go to $n = 64$. This makes sense as ϵ determines how much $P_{\text{Perturbed}}$ is perturbed. As one would expect, the greater the part of the rank n matrix is of $P_{\text{Perturbed}}$, the less information should be required to estimate that the rank is of order n. As a control case it is good to notice that for $\epsilon = 0$ we have a true BMC and in this case the Spectral Clustering Algorithm with unknown K goes to $K = 5$ and seems to plateau there. This is behavior we had hoped to see as it would be devastating to our model if a too long of a sample path would make the prediction of K bad. This behavior would also be counter intuative as the longer the sample path is, the more information one has and the easier it should be to estimate the number of clusters in a BMC.

Next we will try to understand the behavior of $\hat{K}^{\text{pre}}$ in the setting of a perturbed BMC. From Figure 9 we see that as T_n gets large, the estimator $\hat{K}^{\text{pre}}$ goes up. We expect the estimator of $\hat{K}^{\text{pre}}$ to behave similar to $\hat{K}^{\text{post}}$ in the regime where T_n is large with respect to n. However, we can not see from this graph if $\hat{K}^{\text{pre}}$ would grow asymptotically to $n = 64$ like the estimator $\hat{K}^{\text{post}}$ does. For this one would need to run bigger simulations in order to establish a numerical claim. An important observation one can make in the case of a true BMC meaning $\epsilon = 0$ is that the estimator $\hat{K}^{\text{pre}}$ is asymptotically growing to $K = 5$.

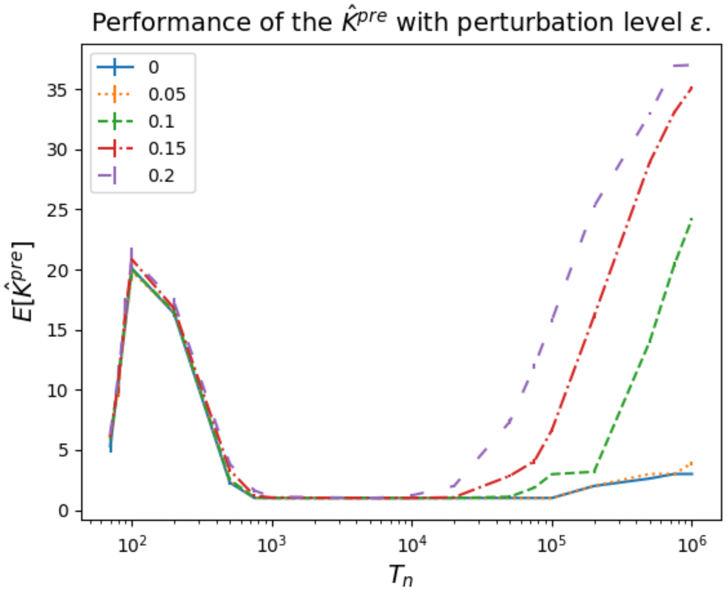

Figure 9: The influence of the length of the path on the accuracy of $\hat{K}^{\text{pre}}$ to predict K. The parameters of the BMC is $K = 5$ and α and the cluster transition matrix p are given in Appendix 7.1.

When one looks again at Figure 9 we observe that for small T_n there is a jump in the graph for all epsilons. This is most likely because if T_n is small with respect to n we would expect that the matrix $\hat{N}_\Gamma$ contains quite some noise. Thus the singular values of this matrix will not have such a clear gap between the Kth and the $(K+1)$th singular value as in the case when T_n is large with respect to n. Thus we expect that less mass is located in the first K singular values and more mass is located in the next $n - K$ singular values. This in turn would lead to a high $\hat{K}^{\text{pre}}$ which we see in Figure 9.

4.4.2 Fraction of misclassified states in a perturbed BMC

In this section the fraction of misclassified is analyzed in the setting of perturbed BMCs. In the previous sections we focused our attention on the ability to estimate the number of clusters K. However, we should not forget that our Spectral Clustering Algorithm with unknown K should still be able to cluster states correctly. Thus in this section the performance measure as described in Section 2.1 will be analyzed. It is also important to mention that one of our research questions we devised in Section 1.1 was how long the sample path of a BMC needed to be, in order to cluster states correctly. Hence, we will look at the fraction of misclassified states against the length of the sample path of a perturbed BMC. Similar to the tests conducted in the previous section we have taken fixed values for K, α and p after which we tested the performance of the Spectral Clustering Algorithm with unknown K on synthetic data from a the performance. Firstly we will look at the ability to cluster states correctly using only our Spectral Clustering Algorithm with unknown K. Secondly, we will use the Cluster Improvement Algorithm discussed in Section 2.3 and run it for $t = 10$ times. There is no particular reason for choosing $t = 10$, however in [13] they often achieved good performances of their Cluster Improvement Algorithm in when taking $t \in \{1, \ldots, 5\}$.

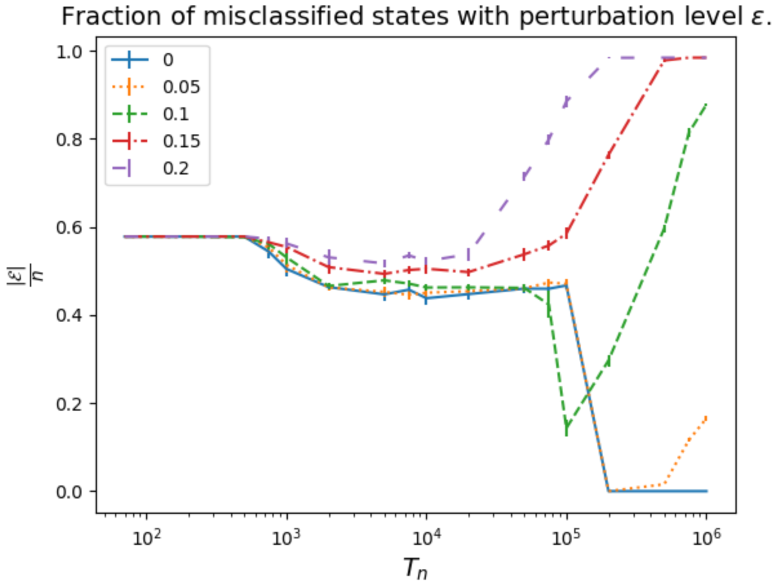

Figure 10: The influence of the path length on the performance of the Spectral Clustering Algorithm with unknown K. The parameters of the BMC is $K = 5$ and α and the cluster transition matrix p are given in the appendix 7.1.

The fraction of misclassified states when using only the Spectral Clustering Algorithm with unknown K is given in Figure 10. Firstly, we can see that if T_n is small that it does not matter what the size of ϵ is, since the Spectral Clustering Algorithm with unknown K will predict $\hat{K}^{\text{post}} = 1$. When looking at the parameters for this BMC we can see that the size of the biggest cluster is $\alpha_1 = \frac{27}{64}$ which is why our $\frac{|\mathcal{E}|}{n} = 1 - \alpha_1 \approx 0.578$ if $\hat{K}^{\text{post}} = 1$. If T_n grows larger, we see that our ability to cluster states correctly first increases for all levels of perturbation. Now if T_n grows large enough we see that if $\epsilon > 0$ it will mean that the fraction of misclassified states will go up to 1. This may seem strange as the guess that all states are in the same cluster is a better guess than the output of the algorithm. However, we arrive at the same conclusion as we made when looking at the performance of $\hat{K}^{\text{post}}$. If $\epsilon > 0$ and if T_n is large enough than $\hat{K}^{\text{post}} = n$. Thus each state belong to it's own cluster meaning that $\frac{|\mathcal{E}|}{n} = 1 - \frac{1}{n}$. Now the size of T_n in order for $\frac{|\mathcal{E}|}{n}$ to be close to zero depends on the size of ϵ. The bigger ϵ the faster this convergence happens. This is similar behavior as our estimator $\hat{K}^{\text{post}}$ showed in Figure 8.

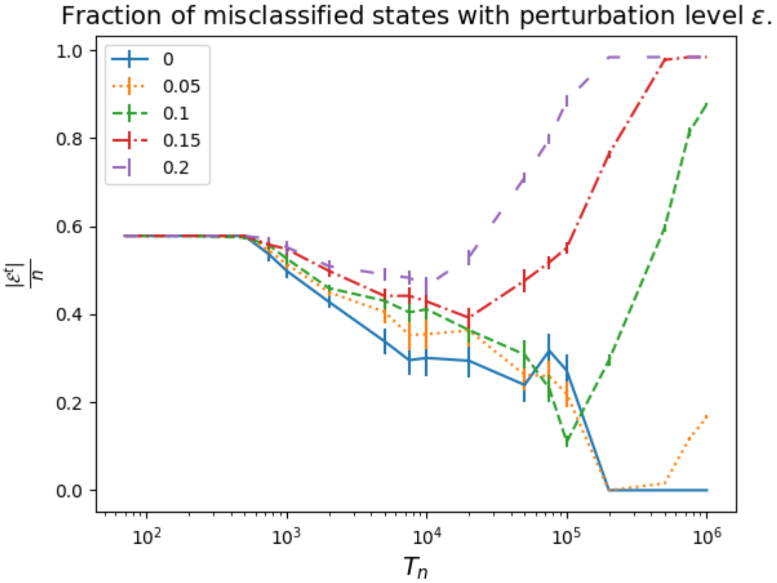

Figure 11: The influence of the path length on the performance of the Spectral Clustering Algorithm with unknown K combined with the Cluster Improvement Algorithm. The parameters of the BMC is $K = 5$ and α and the cluster transition matrix p are given in the appendix 7.1.

Next we will analyze how the Cluster Improvement Algorithm influences the ability to cluster states correctly. As discussed in Section 2.3 the estimator $\hat{K}_t^{\text{CI}}$ will not be shown since from testing it on synthetic data it always coincided with $\hat{K}^{\text{post}}$. When looking at Figure 11 we see similar behavior of $\hat{K}^{\text{post}}$ as in Figure 10. We again notice that if T_n is small enough then $\frac{|\mathcal{E}^{[t]}|}{n} = 1 - \alpha_1 \approx 0.578$. This is logical as if $\hat{K}^{\text{post}} = 1$ then there is no cluster improving to do as there are no other clusters to move states to. Now again if T_n grows we see that $\frac{|\mathcal{E}^{[t]}|}{n}$ decreases meaning the performance of the Cluster Improvement Algorithm is increasing. Now an important observation to make is that this decrease is a lot faster than in Figure 10. This implies that if T_n is not that large with respect to n combining the Spectral Clustering Algorithm with unknown K with the Cluster Improvement Algorithm has a big positive effect on the performance. However, if $\epsilon > 0$ and T_n is large enough then again ϵ is close to $1 - \frac{1}{n}$. This is to be expected as if the path is long enough than $\hat{K}^{\text{pre}}$ and $\hat{K}^{\text{post}}$ go to n. Thus meaning that the underlying model can be seen as a BMC with n clusters. Thus the Cluster Improvement Algorithm will not have an affect on the performance if T_n is large enough.

4.5 Codon pairing

As discussed in Section 3.4 we will analyze a DNA sample of a gene. The data given by [11] can be fed in to our implementation of Algorithm 3 which can give us a understanding of how well the Spectral Clustering Algorithm with unknown K performs on real data. We will look at the amount of clusters the algorithm estimates this data contains. However, we will vary the length of the DNA sample which is given to our algorithm. This results in Figure 12 in which $\hat{K}^{\text{post}}$ is the estimated value of K.

Firstly, we will look at the behavior of the estimator $\hat{K}^{\text{pre}}$ in Figure 12. Notice that it first seems to blow up and then it returns to a lower estimate. This is similar behavior that $\hat{K}^{\text{pre}}$ showed in Figure 9 which we explained by looking at the noise in $\hat{N}$ when the path length is small. If one looks at when T_n gets very large with respect to n, we notice that $\hat{K}^{\text{pre}}$ seems to go up again. This can be explained using the results we have obtained in Figure 9. If the sample path is long enough we expect that $\hat{N}_\Gamma$ is also of rank n because the data is most likely not a real BMC but some object with a transition matrix of rank n. Thus our $\hat{K}^{\text{pre}}$ which depends on the rank of $\hat{N}_\Gamma$ should go up to n.

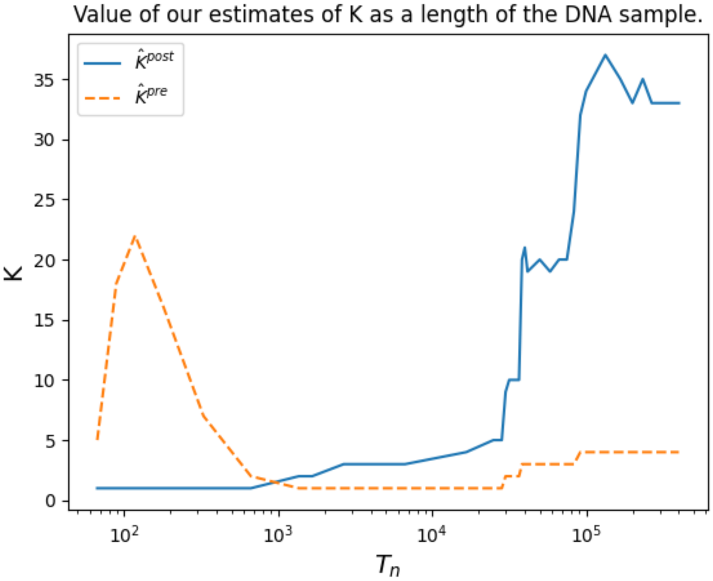

Figure 12: The estimated number or clusters in DNA gene as a function of the path length.

Secondly, we will look at the performance of the estimator $\hat{K}^{\text{post}}$ in Figure 12. A first observation one can make is that the estimated number of clusters increases as the sample path increases. Now in [17] they performed a similar experiment fixing $K = 5$ after an analysis of the matrix $\hat{N}_\Gamma$. With the fixing of $K = 5$ they rediscovered a biological phenomenon known as codon pair bias. Now note that in our case we see that our estimate for $\hat{K}^{\text{post}}$ does not converge to 5. Now a possible reason for this is most real world data is not a BMC. Thus, if the data is an object of order n then as T_n grows large, one expects estimated number of clusters to grow to n in which each cluster contains only contains one state. Now notice that we do see this behavior of $\hat{K}^{\text{post}}$ in Figure 12.

In order to validate the claim that this data is not a real BMC we have chosen the cluster transition matrix for Figures 8 and 9 such that they coincide with the empirical transition matrix obtained from the DNA samples with the clusters given in section 8.1 of [17]. Thus the results discussed in Section 4.4.1 is synthetic data from a perturbed BMC in which the transition matrix of the BMC is the same as the empirical transition matrix of the DNA sample. Thus it is no coincidence that for $\epsilon > 0$ we see that the estimators $\hat{K}^{\text{pre}}$ and $\hat{K}^{\text{post}}$ behave similar in Figures 8, 9 and 12. Thus this strongly suggests that the DNA sample does not follow a true BMC but follows some process which contains some rank n transition matrix.

5 Proof of the optimality of Algorithm 3

In this section we prove Theorem 1. First, some lemmas are introduced which will help us in the proof. We will introduce these lemmas by giving the intuition behind the model and explaining why we made certain choices. After this we will first prove that our algorithm remains optimal after which we will give a proof of why with high probability $\hat{K}^{\text{post}} = K$.

Let $N \triangleq \mathbb{E}[\hat{N}]$, $\alpha \in \Delta^{K-1}$ and $p \in \Delta^{(K-1)\times K}$, and introduce similar to [13] the quantity

$$D(\alpha, p) \triangleq \min_{a,b;a\neq b} \sum_{k=1}^{K} \left(\left(\frac{\pi_a p_{a,k}}{\alpha_k \alpha_a} - \frac{\pi_b p_{b,k}}{\alpha_k \alpha_b} \right)^2 + \left(\frac{\pi_k p_{k,a}}{\alpha_k \alpha_a} - \frac{\pi_k p_{k,a}}{\alpha_k \alpha_b} \right)^2 \right). \tag{17}$$

[13] concludes that when $I(\alpha, p) > 0$, this implies that $D(\alpha, p) > 0$. This implication will be needed in an upcoming proof.

5.1 Trimming and rank-$\hat{K}^{\text{pre}}$ approximation

After trimming $\hat{N}$ as explained in Section 3.1, we want to find a rank $\hat{K}^{\text{pre}}$ approximation of $\hat{N}_\Gamma$ such that $\hat{K}^{\text{pre}}$ is close to K. (10) gives the idea to find K such that $\sigma_K > O_{\mathbb{P}}\left(\sqrt{\frac{T_n}{n}}\right)$. Hence, we have to find a function such that our Kth singular value denoted by σ_K satisfies

$$\sigma_{\hat{K}^{\text{pre}}} \geq \sqrt{\frac{T_n}{n}} f_n \tag{18}$$

Here $f_n : \mathbb{N} \to \mathbb{R}$, may be any sequence that satisfies $\omega(1) = f_n = o\left(\sqrt{\frac{T_n}{n}}\right)$. In our case we will take

$$f_n \triangleq \ln \frac{T_n}{n}$$

.

This in turn means that if we obtain singular values of $\hat{N}_\Gamma$ in descending order $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$, then our $\hat{K}^{\mathsf{pre}}$ is given by

$$\hat{K}^{\mathsf{pre}} = \max\left\{k : \sigma_k \geq \sqrt{\frac{T_n}{n}} f_n\right\}. \tag{19}$$

From the definition above one also immediately understands why $\hat{K}^{\mathsf{pre}}$ is not a very precise estimator. There are a lot of different ways to choose the function f_n which can all be technically correct. Now when one realises that a choice of a lower f_n means one will receive in general a higher $\hat{K}^{\mathsf{pre}}$ which will in turn mean a higher amount of information will be passed on through. However, this could lead to a rank $\hat{K}^{\mathsf{pre}}$ approximation of $\hat{N}_\Gamma$ which is too noisy since too little information has been removed. On the contrary, if one has the bound too high, then a lot of information will be lost in the rank $\hat{K}^{\mathsf{pre}}$ approximation of $\hat{N}_\Gamma$. This leads us on to our first proposition and lemma in which we will use our choice of f_n.

Proposition 1. *Assume that $T_n = \omega(n)$ and $f_n = o\left(\sqrt{\frac{T_n}{n}}\right)$. When using Algorithm 3, $\hat{K}^{\mathsf{pre}} \leq K + 1$ with high probability.*

Proof. Observe that the singular values are in descending order. This implies that if we can show that $\sigma_{\hat{K}^{\mathsf{pre}}}(\hat{N}_\Gamma) \geq \sigma_{K+1}(\hat{N}_\Gamma)$ with high probability, then $\hat{K}^{\mathsf{pre}} \leq K + 1$ with high probability.

By construction of Algorithm 3, we have that $\sigma_{\hat{K}^{\mathsf{pre}}}(\hat{N}_\Gamma) \geq \sqrt{T_n/n} f_n$. From Section 2.4 we know that $\sigma_{K+1}(\hat{N}_\Gamma) = O_{\mathbb{P}}\left(\sqrt{T_n/n}\right)$ which implies that there exists a $C \in \mathbb{R}$ such that $\sigma_{K+1}(\hat{N}_\Gamma) \leq C\sqrt{T_n/n}$ with high probability for sufficiently large n. We conclude that for sufficiently large n and T_n such that $f_n > C$,

$$\sigma_{\hat{K}^{\mathsf{pre}}}(\hat{N}_\Gamma) \geq \sqrt{T_n/n} f_n \geq C\sqrt{T_n/n} \geq \sigma_{K+1}(\hat{N}_\Gamma) \quad \text{with high probability.}$$

□

Lemma 1. $\|\hat{R}^0 - N^0\|_F \leq \sqrt{8(2K+1)}\|\hat{N}_\Gamma - N\|$ *with high probability.*

Proof. The proof of this lemma is similar to the proof be found in SM 4.3 of [13]. Notice that for any matrix $A \in \mathbb{R}^{n \times n}$ that $\|A\|_F^2 = \sum_{i=1}^n \sigma_i^2(A)$ and for the spectral norm $\|A\| = \max_{i=1,\ldots,n}\{\sigma_i(A)\}$. Because $\hat{R}$ is of rank $\hat{K}^{\mathsf{pre}}$ and N is of rank K. We know with high probability using Proposition 1 that $\hat{R} - K$ is at most of rank $2K + 1$. Recall that $\hat{R}^0 = [\hat{R}, \hat{R}^T]$, after which we conclude that

$$\|\hat{R}^0 - N^0\|_F^2 = 2\|\hat{R} - N\|_F^2 \leq 2(2K+1)\|\hat{R} - N\|^2.$$

Using $(143) - (145)$ of [13] we can complete the proof. □

5.2 Bounding the neighborhoods

Before we start selecting the clusters as done in lines 6-15 of Algorithm 3, we calculate the neighborhood for each state $x \in V$. If we let h_n be the function which corresponds to the size of the neighborhood. Then the neighborhood of each state $x \in V$ is given by

$$\mathcal{N}_x \triangleq \left\{ y \in V \,\middle|\, \sqrt{\|\hat{R}_{x,\cdot} - \hat{R}_{y,\cdot}\|_2^2 + \|\hat{R}_{\cdot,x} - \hat{R}_{\cdot,y}\|_2^2} \leq h_n \right\}. \tag{20}$$

In order to make sure that the neighborhood of each state $x \in V$ contains the right states it is important to choose the bound h_n right. The same criteria as in the proof of Lemma 6 in [13] are used, which yields us the equality.

$$\omega\left(\frac{f_n^2}{n}\right) = h_n^2 = o\left(\frac{T_n^2}{n^3}\right). \tag{21}$$

Since $f_n = o\left(\sqrt{\frac{T_n}{n}}\right)$, we can conclude that we have to asymptotically let h_n^2 be between the functions $\frac{T_n}{n^2}$ and $\frac{T_n^2}{n^3}$. There are different options to choose for h_n; we choose

$$h_n \triangleq \sqrt{\frac{T_n^2}{n^3 \ln \frac{T_n}{n}}}. \tag{22}$$

In the following lemma the reason for this choice of h_n becomes apparent.

Lemma 2. *Let $x, y \in V$. Then if $\sigma(x) \neq \sigma(y)$, then*

$$\|N_{x,\cdot}^0 - N_{y,\cdot}^0\|_2 = \Omega\left(\frac{T_n\sqrt{D(\alpha,p)}}{n^{3/2}}\right)$$

Proof. The proof of Lemma 3 can be found in SM4.2 of [13]. □

5.3 Clustersize bound

In the situation that K is known one can sequentially select K centers and then the K clusters. However, in the case that K is unknown, a stopping criteria is needed in order to know when to stop generating new clusters. In our Spectral Clustering Algorithm with unknown K we therefore determined a lower bound ρ on the size of the ith approximate cluster $\hat{\mathcal{V}}_i$. Thus we select clusters in the same manner as in Algorithm 1, however, once the cardinality of the ith cluster drops below ρ, the process is stopped and $\hat{K}^{\text{post}} = i - 1$ is set. $\hat{K}^{\text{post}}$ will be our estimator of K, and $\rho > 0$ is given by

$$\rho \triangleq \frac{n^2 \ln \frac{T_n}{n}}{T_n}.$$

The fact that each cluster $\hat{\mathcal{V}}_k \geq \rho$ for all $k \in \{1, \ldots, \hat{K}^{\text{post}}\}$ is used in the proof of the following lemma.

Lemma 3. *If $\|\hat{N}_\Gamma - N\| = o_{\mathbb{P}}(f_n)$ for some sequence $f_n = o\left(\frac{T_n}{n}\right)$ and there exists a sequence h_n such that $\omega\left(\frac{f_n}{\sqrt{n}}\right) = h_n = o\left(\frac{T\sqrt{D(\alpha,p)}}{n^{3/2}}\right)$, then*

$$\|\hat{R}^0_{x,\cdot} - N^0_{x,\cdot}\|_2 = \Omega_{\mathbb{P}}\left(\frac{T_n\sqrt{D(\alpha,p)}}{n^{3/2}}\right) \text{ for any misclassified state } x \in \mathcal{E}$$

Proof. The proof of Lemma 3 will be similar to the proof presented in SM4.4 of [13].

Let $\bar{N}^0_k \triangleq (1/|\mathcal{V}_k|)\sum_{z\in\mathcal{V}} N^0_{z,\cdot}$ for $k = 1,\dots,K$. One can think of $\bar{N}^0_k$ as the underlying center of the kth cluster which in an ideal case should be close to $\hat{R}^0_{z^*_k}$ which is found in (13). Recall the definition of $\mathcal{N}_x$ for $x \in V$ in (20). We specified a specific function for h_n, however, in this proof we will assume that h_n satisfies (21).

The approach of the proof will be similar to the proof given in SM4.4 of [13]. We show that for any $0 < a < 1/2$ the recursive algorithm (13) will (for sufficiently large n, T_n) give centers $z^*_1,\dots,z^*_{\hat{K}^{\text{post}}}$ satisfying

$$\|\hat{R}^0_{z^*_k} - \bar{N}^0_{\gamma(k)}\|_2 < ah_n \quad \text{for} \quad k = 1,\dots,\hat{K}^{\text{post}} \tag{23}$$

for some permutation γ. Assuming (23) holds, we can finish the proof by checking two cases. Notice that if $x \in \mathcal{E}$ then x was not in the neighborhood of its cluster ($x \notin \mathcal{N}_{z^*_{\sigma(x)}}$). We distinguish two cases: either x was in the neighborhood of another cluster and was misclassified via (13) or x was not in any neighborhood and was misclassified via (15).
Case 1: If $x \in \mathcal{N}_{z^*_c}$ for some $c \neq \sigma(x)$, we have by (13) and (23)

$$\|\hat{R}^0_{x,\cdot} - \bar{N}^0_c\|_2 \leq \|\hat{R}^0_{x,\cdot} - \hat{R}^0_{z^*_c,\cdot}\|_2 + \|\hat{R}^0_{z^*_c,\cdot} - \bar{N}^0_c\|_2 \leq (1+a)h_n. \tag{24}$$

Notice that for some vectors $a, b, c \in \mathbb{R}^n$, we get using the triangle inequality that

$$\|a-b\|_2 \leq \|a-c\|_2 + \|c-b\|_2 \implies \|a-c\|_2 \geq \left|\|a-b\|_2 - \|c-b\|_2\right|. \tag{25}$$

Lemma 2 and (24) give the lower bound

$$\|\hat{R}^0_{x,\cdot} - \bar{N}^0_{\sigma(x)}\|_2 \overset{(25)}{\geq} \left|\|\bar{N}^0_{\sigma(x)} - \bar{N}^0_c\|_2 - \|\bar{N}^0_c - \bar{N}^0_{\sigma(x)}\|_2\right| \geq \frac{T_n\sqrt{D(\alpha,p)}}{n^{3/2}} - (1+a)h_n \tag{26}$$

Since $h_n = o(\frac{T_n}{n^{3/2}})$, Lemma 3 holds.
Case 2: Otherwise $x \in \left(\bigcup_{k=1}^{\hat{K}^{\text{post}}} \mathcal{N}_{z^*_k}\right)^c$ and by (15) there exist a center $z^*_c \in V$ such that $\|\hat{R}^0_{z^*_c,\cdot} - \hat{R}^0_{x,\cdot}\|_2 \leq \|\hat{R}^0_{z^*_{\sigma(x)},\cdot} - \hat{R}^0_{x,\cdot}\|_2$. Because (23) implies that each center z^*_k is ah_n close to its truth $\bar{N}^0_k$, and Lemma 2 implies that $\bar{N}^0_k$ and $\bar{N}^0_l$ are $\Omega\left(\frac{T_n}{n^{3/2}}\right)$ apart for any $k \neq l$, we conclude that $\|\hat{R}^0_{x,\cdot} - \bar{N}^0_{\sigma(x)}\|_2 = \Omega\left(\frac{T_n}{n^{3/2}}\right)$.

To prove (23), we will construct K disjoint set $C_1, \dots, C_K$ such that

$$\left|\mathcal{N}_z \backslash \bigcup_{l=1}^{k-1} \hat{\mathcal{V}}_l\right| \geq m_k, \quad \exists z \in \left(\bigcup_{k=1}^{K} C_k\right) \backslash \left(\bigcup_{l=1}^{k-1} \hat{\mathcal{V}}_l\right) \tag{27}$$

With m_k being the kth largest value of $\{|C_1|, \dots, |C_K|\}$. The existence of sets $C_1, \dots, C_K$ implies that for any one of the centers $z_1^*, \dots, z_{\hat{K}^{\text{post}}}^*$ provided by (13) it is impossible to be an outlier when n, T_n are sufficiently large. We define the sets of cores:

$$C_k \triangleq \left\{x \in \mathcal{V}_k \middle| \|\hat{R}^0_{x,\cdot} - \bar{N}^0_k\| < ah_n\right\} \quad \text{for } k = 1, \dots, K$$

and conversely the set of outliers $\mathcal{O}$ can be introduced

$$\mathcal{O} \triangleq \left\{x \in \mathcal{V}_k \middle| \|\hat{R}^0_{x,\cdot} - \bar{N}^0_k\| \geq bh_n, \, \forall k = 1, \dots, K\right\}$$

In 13 we give a visual representation of how one can interpret these concepts. W.l.o.g. we assume that the C_k are ordered based on their cardinality. We have to put some restrictions on a and b: assume that $0 < a < \frac{1}{2}$ and $b - a > 1$.

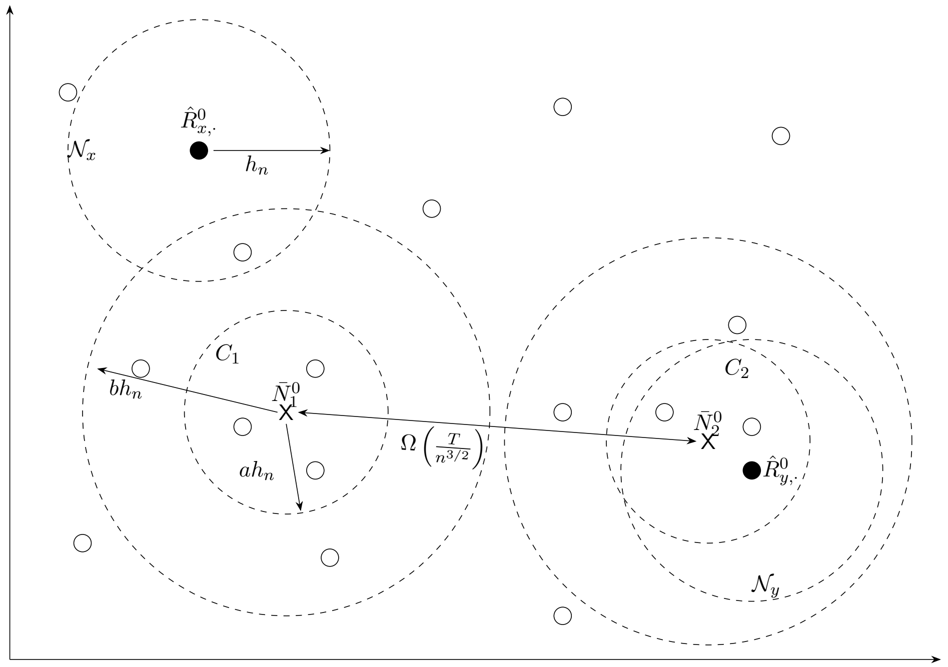

Figure 13: Schematic representation courtesy of [13] of the cores C_1 and C_2 for a situation in which $K = 2$ and for which $x \in \mathcal{O}$ and $y \in C_2$ with their neighborhoods $\mathcal{N}_x$ and $\mathcal{N}_y$ respectively.

Notice the following four properties of the cores and outliers:

1. $\left|\left(\bigcup_{k=1}^{K} C_k\right) \cap \mathcal{N}_x\right| = 0$ for all $x \in \mathcal{O}$.

   Proof. Recall that if $y \in \mathcal{N}_x$, $\|\hat{R}^0_{x,\cdot} - \hat{R}^0_{y,\cdot}\|_2 \leq h_n$. Then by the definition of an outlier we have that

   $$\|\hat{R}^0_{y,\cdot} - \bar{N}^0_k\|_2 \overset{(25)}{\geq} \left|\|\hat{R}^0_{x,\cdot} - \bar{N}^0_k\|_2 - \|\hat{R}^0_{x,\cdot} - \hat{R}^0_{y,\cdot}\|_2\right| \geq (b-1)h_n \geq ah_n. \tag{28}$$

   Thus $y \in \mathcal{N}_x$ can not be in any of the cores C_k. If $y \in C_k$ for some k then

   $$\|\hat{R}^0_{x,\cdot} - \hat{R}^0_{y,\cdot}\|_2 \overset{(25)}{\geq} \left|\|\hat{R}^0_{x,\cdot} - \bar{N}^0_k\|_2 - \|\hat{R}^0_{y,\cdot} - \bar{N}^0_k\|_2\right| \geq (b-a)h_n > h_n. \tag{29}$$

   This implies that $y \notin \mathcal{N}_x$ which concludes the proof. □

2. $\left|\left(\bigcup_{k=1}^{K} C_k\right)^c\right| = \frac{\|\hat{R}^0 - N^0\|_F^2}{\min_{x \in \left(\bigcup_{k=1}^{K} C_k\right)^c} \|\hat{R}^0_{x,\cdot} - N^0_{x,\cdot}\|_F^2} = \frac{O_{\mathbb{P}}\left(\frac{T_n}{n}\right)}{\Omega_{\mathbb{P}}\left(\frac{T_n^2 D(\alpha,p)}{n^3}\right)} = O_{\mathbb{P}}\left(\frac{n^2}{T_n}\right).$

   Proof. From Lemma 1 we know that

   $$8(2K+1)\|\hat{N}_\Gamma - N\|^2 \geq \|\hat{R}^0 - N^0\| = \sum_{x \in V} \|\hat{R}^0_{x,\cdot} - \bar{N}^0_{\sigma(x)}\|_2^2. \tag{30}$$

   Since $C_k \subseteq \mathcal{V}_k$ by construction we get that

   $$\sum_{x \in V} \|\hat{R}^0_{x,\cdot} - \bar{N}^0_{\sigma(x)}\|_2^2 \geq \left|\left(\bigcup_{k=1}^{K} C_k\right)^c\right| \min_{x \in \left(\bigcup_{k=1}^{K} C_k\right)^c} \{\|\hat{R}^0_{x,\cdot} - \bar{N}^0_{\sigma(x)}\|_2^2\} \tag{31}$$

   Notice that for every $x \in \left(\bigcup_{k=1}^{K} C_k\right)^c$, $\|\hat{R}^0_{x,\cdot} - \bar{N}^0_{\sigma(x)}\|_2 \geq ah_n$ by definition. We obtain that

   $$8(2K+1)\|\hat{N}_\Gamma - N\|^2 \geq \left|\left(\bigcup_{k=1}^{K} C_k\right)^c\right| \min_{x \in \left(\bigcup_{k=1}^{K} C_k\right)^c} \{\|\hat{R}^0_{x,\cdot} - \bar{N}^0_{\sigma(x)}\|_2^2\} \tag{32}$$

   $$\geq \left|\left(\bigcup_{k=1}^{K} C_k\right)^c\right| a^2 h_n^2 \tag{33}$$

   Rearrange to conclude that $\left|\left(\bigcup_{k=1}^{K} C_k\right)^c\right| = \frac{8(2K+1)}{a^2}\frac{\|\hat{N}_\Gamma - N\|^2}{h_n^2} = O_{\mathbb{P}}\left(\frac{f_n^2}{h_n^2}\right)$. By construction of f_n and h_n we find that $\left|\left(\bigcup_{k=1}^{K} C_k\right)^c\right| = O_{\mathbb{P}}\left(\frac{n^2}{T_n}\right)$. Note that by repeating $(28)-(31)$ we can conclude that $8(2K+1)\|\hat{N}_\Gamma - N\|^2 \geq a^2 h_n^2\, |C_k^c \cap \mathcal{V}|$, which implies that

   $$|C_k^c \cap \mathcal{V}| = O_{\mathbb{P}}\left(\frac{n^2}{T_n}\right). \tag{34}$$

   □

3. $C_{\sigma(x)} \subseteq \mathcal{N}_x$ for all $x \in \bigcup_{k=1}^K C_k$.

   Proof. Let $x \in \left(\bigcup_{k=1}^K C_k\right)$. Since the cores C_k are disjoint there exist $l \in \{1, \dots, K\}$ such that $x \in C_l$. Let $y \in C_l$ and recall $\|\hat{R}^0_{y,\cdot} - \bar{N}^0_l\|_2 < ah_n$. Since $2a < 1$ by construction, we conclude

   $$\|\hat{R}^0_{x,\cdot} - \hat{R}^0_{y,\cdot}\|_2 \leq \|\hat{R}^0_{x,\cdot} - \bar{N}^0_l\|_2 + \|\hat{R}^0_{y,\cdot} - \bar{N}^0_l\|_2 < 2ah_n < h_n.$$

   This proves that $y \in \mathcal{N}_x$. □

4. If $|\mathcal{N}_x \cap C_k| \geq 1$ and n, T_n are sufficiently large, then for any $l \neq k$ we have that $|\mathcal{N}_x \cap C_l| = 0$.

   Proof. Notice that it suffices to show that if $l \neq k$. Then $C_l \cap C_k = \varnothing$. We will argue by contradiction. Let $x \in C_l$ and $x \in C_k$. Then we have that

   $$\|\hat{R}^0_{x,\cdot} - \bar{N}^0_l\| < ah_n \text{ and } \|\hat{R}^0_{x,\cdot} - \bar{N}^0_k\| < ah_n$$

   However we know that $\|\bar{N}^0_k - \bar{N}^0_l\| = \Omega\left(\frac{T}{n^{3/2}}\right) \geq h_n$ by construction of h_n. Using the triangle inequality we find

   $$\begin{aligned}\|\bar{N}^0_k - \bar{N}^0_l\| &\leq \|\hat{R}^0_{x,\cdot} - \bar{N}^0_l\| + \|\hat{R}^0_{x,\cdot} - \bar{N}^0_k\| \\ &< 2ah_n < h_n.\end{aligned}$$

   This is a contradiction and thus x can't be in both clusters at the same time thus $C_l \cap C_k = \varnothing$. □

Observe that if $x \in \mathcal{O}$ and $y \in \bigcup_{k=1}^K C_k$ we have that $y \notin \mathcal{N}_x$. Hence $y \in \left(\bigcup_{k=1}^K C_k\right)^c$ is necessary for $y \in \mathcal{N}_x$. Then from properties $1, 2$ and the observation made just now we have that

$$|\mathcal{N}_x| = O_{\mathbb{P}}\left(\frac{n^2}{T_n}\right) \quad \forall x \in \mathcal{O} \tag{35}$$

Properties 2, 3 and 4 show that

$$\left|\mathcal{N}_z \backslash \bigcup_{l=1}^{k-1} \hat{\mathcal{V}}_l\right| \geq m_k, \quad \exists z \in \left(\bigcup_{k=1}^{K} C_k\right) \backslash \left(\bigcup_{l=1}^{k-1} \hat{\mathcal{V}}_l\right) \tag{36}$$

We also conclude that by definition of C_k that $|C_k| = |V_k| - |C_k^c \cap V_k|$, which implies that using (34)

$$|C_k| = |V_k| - |C_k^c \cap V_k| = n\alpha_k - O_{\mathbb{P}}\left(\frac{n^2}{T_n}\right) = n\alpha_k(1 - o_{\mathbb{P}}(1)). \tag{37}$$

□

5.4 Proving that the fraction of misclassified states are of $o_{\mathbb{P}}(1)$

From Lemma 3 we conclude that $\|\hat{R}^0 - N^0\|_F^2 \geq \sum_{x\in\mathcal{E}} \|\hat{R}^0_{x,\cdot} - N^0_{x,\cdot}\|_2^2 = |\mathcal{E}|\Omega_{\mathbb{P}}\left(\frac{T_n^2 D(\alpha,p)}{n^3}\right)$. Notice that since $I(\alpha,p) > 0$, $D(\alpha,p) > 0$ as concluded in [13]. Furthermore, if we have two families of random variables denoted by $\cup_{n=1}^{\infty}\{X_n\}_{n\geq 0}$ and $\cup_{n=1}^{\infty}\{Y_n\}_{n\geq 0}$ and two sequences x_n, y_n such that $X_n = O_{\mathbb{P}}(x_n)$ and $Y_n = \Theta_{\mathbb{P}}(y_n)$. Then, by Lemma 22 from [13] we conclude that

$$\frac{X_n}{Y_n} = O_{\mathbb{P}}\left(\frac{x_n}{y_n}\right). \tag{38}$$

By (38) and Lemmas 1 and 3 we conclude,

$$\frac{|\mathcal{E}|}{n} = \frac{O_{\mathbb{P}}\left(\|\hat{N}_\Gamma - N\|^2\right)}{\Omega_{\mathbb{P}}\left(\left(\frac{T_n\sqrt{D(\alpha,p)}}{n^{3/2}}\right)^2\right)} = O_{\mathbb{P}}\left(\frac{T_n/n}{(T_n/n)^2}\right) = O_{\mathbb{P}}\left(\frac{n}{T_n}\right) = o_{\mathbb{P}}(1).$$

5.5 Proving that $\hat{K}^{\text{post}} = K$ with high probability

Recall (36) and that we assumed the C_i are ordered in a descending way based on their cardinality, then after the initial cluster assignment

$$|\hat{\mathcal{V}}_K| \geq m_K = |\mathcal{C}_K| = |\mathcal{V}_K| - |\mathcal{C}_K^c \cap \mathcal{V}_K| = n\alpha_K(1 - o_{\mathbb{P}}(1)). \tag{39}$$

Since the cores are disjoint and using (39) we find

$$|\hat{\mathcal{V}}_{K+1}| \leq n - \sum_{k=1}^{K} |C_k| = n\left(1 - \sum_{k=1}^{K} \alpha_k(1 - o_{\mathbb{P}}(1))\right) = o_{\mathbb{P}}(n). \tag{40}$$

Recall after running Algorithm 3 the cluster assignment satisfies

$$\hat{\mathcal{V}}_{\hat{K}^{\text{post}}} = \Theta\left(\frac{n^2 \ln\frac{T_n}{n}}{T_n}\right) \quad \text{and} \quad \hat{\mathcal{V}}_{\hat{K}^{\text{post}}+1} = o\left(\frac{n^2 \ln\frac{T_n}{n}}{T_n}\right). \tag{41}$$

Then from (39), (40) and (41) we can conclude that $\hat{K}^{\text{post}} = K$ with high probability. This concludes Theorem 1.

6 Conclusion

When looking at Chapter 4, we first validated our model using synthetic data. For this we introduced the concept of normalized singular values of the matrix $\hat{N}_\Gamma$. We verified these normalized singular values by using the theory covered in Section 2.4. The results we obtained confirmed that the normalized singular values were in accordance with the theory. This gave some validation that the Spectral Clustering Algorithm with unknown K was working as expected and that our implementation was also correct. After this we looked at the estimators of K of Algorithm 3. We noticed that in sparse regimes $\hat{K}^{\textsf{pre}}$ blows up as n grows and attributed this to the noise in the trimmed empirical transition matrix $\hat{N}_\Gamma$. Now when looking at $\hat{K}^{\textsf{post}}$ we noticed that this estimator was a lot more accurate at detecting the number of clusters K. Furthermore, it did not blow up in a sparse regime of T_n. However, we did identify that our algorithm should be tested at bigger values of n since it could be that $\hat{K}^{\textsf{post}}$ in the regimes of $T_n = n(\ln n)^{1/2}$ and $T_n = n(\ln n)^{3/2}$ could still go to K. However, in general the results gathered from the tests on synthetic data pointed out that our algorithm is able to estimate the amount of clusters K. However, a requirement for an accurate estimation is that n is very large and that T_n is relatively big compared to n, e.g. in the case that T_n is dense.

When looking at the results of testing our algorithm on synthetic data from a perturbed BMC we obtained interesting results. Firstly, if the perturbation is small enough then there exists a T_n such that the original K clusters break up into smaller sub clusters which the algorithm predicts. However, this has a negative influence on the fraction of misclassified states since a lot of small clusters lead to a big amount of misclassified clusters. Secondly, we noticed that $\hat{K}^{\textsf{pre}}$ grew large as the perturbation level grew. Our reasoning for this was that $\hat{N}_\Gamma$ became more of a rank n matrix as ϵ grew, which implies that $\hat{K}^{\textsf{pre}}$ goes up to n. Thirdly, we noticed that the performance when K is known to the algorithm, the clustering performance improves significantly in the setting of a perturbed BMC. This is not a surprising result as it would be counter intuitive if the outcome of the model became worse when more information about the underlying structure of the BMC was known. A fourth conclusion we made was, if we have a perturbed BMC with $\epsilon > 0$, then at some point when T_n is large enough our estimator will converge to n. This was attributed to the fact that the transition matrix of a perturbed BMC is of rank n. Thus if one feeds enough information to the algorithm, the algorithm will differentiate each state into its own cluster, implying that $\hat{K}^{\textsf{post}}$ will go up to n. However, a good verification of our model was that in the case that $\epsilon = 0$ we saw that the $\hat{K}^{\textsf{post}}$ seemed to converge to K. Thus in the case of a real BMC, our model seems to benefit from long sample paths. This seems intuitive and strengthens the robustness of the model. Lastly, we looked at the clustering performance of the Spectral Clustering Algorithm with unknown K with and without using the Cluster Improvement Algorithm. We saw that if T_n was not too small and not too large that the clustering performance of the Spectral Clustering Algorithm with unknown K was better than putting all states in the first cluster. However, when using the Cluster Improvement Algorithm in sequence the clustering performance was greatly enhanced. However, if ϵ and T_n were large enough the Cluster Improvement Algorithm could not stop the fraction of misclassified states going to $1 - \frac{1}{n}$.

After all the synthetic data testing, we moved on to applying a real world data set. This data set consisted of a long sequence of codons of a gene. We observed that the performance of the model was similar to the case when the underlying model was a perturbed BMC with $\epsilon > 0$. Therefore, we concluded that one should be careful when applying this model to real world data. This is because most real world datasets do not follow true BMCs. As seen in Section 4.4.1, if the underlying transition matrix is of rank n, then the estimators for K will converge to n when the T_n is very long with respect to n.

In Theorem 1 we gave an upper bound on the fraction of misclassified states. We noticed that this upperbound is asymptotically equivalent to the lowerbound devised in [13] and discussed in Section 2.1. This in turn implies that our Spectral Cluster Algorithm with unknown K achieves asymptotically accurate detection whenever this is possible. A possible situation in which this is possible is $I(\alpha, p) > 0$ and $T_n = \omega(n)$. This shows that the Spectral Clustering Algorithm discussed in Section 2.2 remains asymptotically accurate even when the K is unknown. However, in general the performance of the Spectral Clustering Algorithm with unknown K is worse when compared to the Spectral Clustering Algorithm.

Acknowledgments

I would like to thank my father (James Cronk) for proofreading my final version which caught many of the grammar and spelling mistakes in this thesis.

References

[1] Emmanuel Abbe, Afonso S. Bandeira, and Georgina Hall. "Exact recovery in the stochastic block model". In: *IEEE Transactions on Information Theory* 62.1 (Jan. 2016), pp. 471–487. ISSN: 00189448. DOI: 10.1109/TIT.2015.2490670.

[2] Emmanuel Abbe and Colin Sandon. "Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic BP, and the information-computation gap". In: (Dec. 2015). URL: http://arxiv.org/abs/1512.09080.

[3] Emmanuel Abbe and Colin Sandon. "Recovering Communities in the General Stochastic Block Model Without Knowing the Parameters". In: (2005). URL: https://arxiv.org/abs/1506.03729.

[4] Sourav Chatterjee. "Matrix estimation by Universal Singular Value Thresholding". In: *Annals of Statistics* 43.1 (Feb. 2015), pp. 177–214. ISSN: 21688966. DOI: 10.1214/14-AOS1272.

[5] Uriel Feige and Eran Ofek. "Spectral techniques applied to sparse random graphs". In: *Random Structures and Algorithms* 27.2 (Sept. 2005), pp. 251–275. ISSN: 10429832. DOI: 10.1002/rsa.20089.

[6] Bruce Hajek, Yihong Wu, and Jiaming Xu. "Achieving exact cluster recovery threshold via semidefinite programming". In: *IEEE Transactions on Information Theory*. Vol. 62. 5. Institute of Electrical and Electronics Engineers Inc., May 2016, pp. 2788–2797. DOI: 10.1109/TIT.2016.2546280.

[7] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. "Stochastic blockmodels: First steps". In: *Social Networks* 5.2 (June 1983), pp. 109–137. ISSN: 0378-8733. DOI: 10.1016/0378-8733(83)90021-7.

[8] Brian Karrer and M. E.J. Newman. "Stochastic blockmodels and community structure in networks". In: *Physical Review E- Statistical, Nonlinear, and Soft Matter Physics* 83.1 (Jan. 2011). ISSN: 15393755. DOI: 10.1103/PhysRevE.83.016107.

[9] Pierre Latouche, Etienne Birmelé, and Christophe Ambroise. "Model selection in overlapping stochastic block models". In: *Electronic Journal of Statistics* 8.1 (2014), pp. 762–794. ISSN: 19357524. DOI: 10.1214/14-EJS903.

[10] Pierre Latouche, Etienne Birmelé, and Christophe Ambroise. "Overlapping stochastic block models with application to the French political blogosphere". In: *Annals of Applied Statistics* 5.1 (Mar. 2011), pp. 309–336. ISSN: 19326157. DOI: 10.1214/10-AOAS382.

[11] National Library of Medicine. *OCA2 melanosomal transmembrane protein homo sapiens (human)*. 2021. URL: https://www.ncbi.nlm.nih.gov/gene/4948.

[12] Lijun Peng and Luis Carvalho. "Bayesian degree-corrected stochastic blockmodels for community detection". In: *Electronic Journal of Statistics* 10.2 (2016), pp. 2746–2779. ISSN: 19357524. DOI: 10.1214/16-EJS1163.

[13] Jaron Sanders, Alexandre Proutière, and Se-Young Yun. "Clustering in Block Markov Chains". In: *The Annals of Statistics* 48.6 (Dec. 2020), pp. 3488–3512. URL: http://arxiv.org/abs/1712.09232.

[14] Jaron Sanders and Albert Senen–Cerda. "Spectral norm bounds for block Markov chain random matrices". In: *Stochastic Processes and their Applications* 158 (Apr. 2023), pp. 134–169. ISSN: 0304-4149. DOI: 10.1016/J.SPA.2022.12.004.

[15] Jaron Sanders and Alexander Van Werde. "Singular value distribution of dense random matrices with block Markovian dependence". In: *Stochastic Processes and their Applications* 158 (Apr. 2023), pp. 453–504. ISSN: 0304-4149. DOI: 10.1016/J.SPA.2023.01.001.

[16] Tom A. B. Snijders and Krysztof Nowicki. "Estimation and Prediction for Stochastic Blockmodels for Graphs with Latent Block structure". In: *Journal of Classification 14* (1997), pp. 75–100.

[17] Alexander Van Werde, Albert Senen-Cerda, Gianluca Kosmella, and Jaron Sanders. "Detection and Evaluation of Clusters within Sequential Data". In: (Oct. 2022). URL: http://arxiv.org/abs/2210.01679.

[18] Se-Young Yun and Alexandre Proutiere. "Community Detection via Random and Adaptive Sampling". In: *Proceedings of The 27th Conference on Learning Theory*. Ed. by Maria Florina Balcan, Vitaly Feldman, and Csaba Szepesvári. Barcelona, Spain: PMLR, June 2014, pp. 138–175. URL: `https://proceedings.mlr.press/v35/yun14.html`.

[19] Se-Young Yun and Alexandre Proutiere. "Optimal Cluster Recovery in the Labeled Stochastic Block Model". In: *NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems*. Ed. by Daniel D. Lee, Ulrike von Luxburg, Roman Garnett, Masashi Sugiyama, and Isabelle Guyon. Barcelona, Spain: Curran Associates Inc., Oct. 2016, pp. 973–981. ISBN: 978-1-5108-3881-9. URL: `https://proceedings.neurips.cc/paper/2016/file/a8849b052492b5106526b2331e526138-Paper.pdf`.

7 Appendix

7.1 Parameters of the perturbed BMC

The parameters of the BMC were chosen based on the DNA data covered in Section 3.4. From [17], we have an idea of how the underlying clusters of this data should look like. They take $K = 5$ and determine that the clusters are given by,

$$
\begin{aligned}
\mathcal{V}_1 = {} & \text{AAA, AAG, TGT, AGT, CCT, TCT, ACT, CAG, ATT, ATG,}\\
& \text{CAT, TAT, AAT, TTG, CTT, TGA, CTG, CAA, TGG, ATA,}\\
& \text{TTA, AGG, TAA, ACA, TCA, CCA, AGA}\\
\mathcal{V}_2 = {} & \text{CAC, GCC, CCC, TCC, ACC, GTC, CTC, TTC, ATC, TGC,}\\
& \text{AGC, TAC, AAC, GGC, TAG, CTA, GAC}\\
\mathcal{V}_3 = {} & \text{GTG, GAG, GGT, GCA, GAA, GTA, GGA, GAT, GGG, GTT,}\\
& \text{GCT}\\
\mathcal{V}_4 = {} & \text{CGA, CGC, ACG, TCG, CCG, GCG, CGT, CGG}\\
\mathcal{V}_5 = {} & \text{TTT}
\end{aligned}
$$

Thus from there we can try to recreate the transition matrix by looking at the data given in [11]. From this we can determine that the distribution of the $K = 5$ clusters of the $n = 64$ states is given by

$$\alpha = \begin{pmatrix} 0.421875 & 0.265625 & 0.171875 & 0.125 & 0.015625 \end{pmatrix}.$$

With a cluster probability matrix given by

$$p = \begin{pmatrix} 0.507585391405 & 0.226065716999 & 0.203055875860 & 0.020883008706 & 0.042410007027 \\ 0.621191367533 & 0.249240761531 & 0.059312727219 & 0.031463565194 & 0.038791578521 \\ 0.479504028898 & 0.231135534197 & 0.229255588613 & 0.025038887602 & 0.035065960687 \\ 0.457059595472 & 0.254847203459 & 0.216852061118 & 0.046628320640 & 0.024612819308 \\ 0.542982925813 & 0.206734487476 & 0.156225831399 & 0.011719911951 & 0.082336843357 \end{pmatrix}.$$