BICLUSTERING USING NONPARAMETRIC BAYESIAN METHODS

by

Safiye Çelik

B.S., Computer Engineering, Middle East Technical University, 2007

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2012

ACKNOWLEDGEMENTS

First and foremost, I would like to offer my gratitude to my thesis advisor Assist. Prof. Ali Taylan Cemgil. Throughout the work for this thesis he always allocated time to answer my questions and he gave me valuable remarks on how to improve the experiments or explain subjects more clearly. Without his guidance, this thesis would never be complete.

I would also like to thank my friends for their valuable and colorful friendship. Not only did they motivate me whenever I felt weak and tired, but they also reminded me of other, non-academic (and probably more important) things in life.

I dedicate this work to my parents and sisters. They have always been there with their continuous support. Without their encouragement, motivation, patience and love, I could never bear the burden.

ABSTRACT

BICLUSTERING USING NONPARAMETRIC BAYESIAN METHODS

Multiway clustering is a popular analysis method due to its several potential applications. Various techniques have been developed to cluster different entities of a data matrix simultaneously by taking relational entries into account. Many of those techniques assume that the number of clusters to be discovered is known prior to the clustering operation. However, in real-world problems we have limited knowledge about the number of clusters before discovering them. Nonparametric methods, on the other hand, perform biclustering and learn the number of clusters concurrently. In this thesis, we introduce two nonparametric Bayesian biclustering methods that are applicable on two-way data. In the first method we model the rows and columns of the two-way data using Dirichlet Process Mixture Models and cluster them simultaneously, whereas in the second one we cluster the entities separately after applying spectral matrix decomposition on the data. We apply the biclustering algorithms on four different datasets; a simulated dataset created by a generative Gaussian model, a dataset of animals and their attributes, a cross-national trade and diplomacy dataset with five different relational networks, and a biological dataset from a microarray study of lung cancer. Since there are few real world data annotated with ground truth biclusters, we generally utilize link prediction in order to evaluate biclustering performances. We randomly remove data entries and predict them based on the fact that the entries in the same bicluster are similar to each other. First biclustering method results in higher accuracy since it makes use of all relational information in the data while the spectral method reduces dimensionality of the data prior to the clustering operation. On the other hand, computational complexity of spectral method is far less due to the reduction in the data entries to process.

ÖZET

PARAMETRİK OLMAYAN BAYESCİ YÖNTEMLERLE İKİ İNDİS ÜZERİNDEN ÖBEKLEME

Çok indis üzerinden veri öbekleme, olası birçok uygulaması dolayısıyla popüler bir araştırma alanıdır. Bir data matrisinin farklı indislerini aynı anda öbekleyebilmek için farklı yöntemler geliştirilmiştir. Bu yöntemlerin çoğu, ortaya çıkarılacak öbek sayısının öbekleme işleminden önce bilindiğini varsaymaktadır. Ancak gerçek hayat problemlerinde, öbekler ortaya çıkarılmadan önce öbek sayısı hakkında eldeki bilgi sınırlıdır. Buna karşılık parametrik olmayan yöntemler ise iki indis üzerinden öbekleme işlemi ile eş zamanlı olarak öbek sayısını öğrenmektedir. Bu tezde, iki adet parametrik olmayan iki indis üzerinden Bayesci öbekleme yöntemi tanıtılmaktadır. İlk yöntemde iki indisli verinin satır ve sütunları Dirichlet Süreci Karışım Modelleri ile modellenip eş zamanlı olarak öbeklenirken, ikinci yöntemde ise satır ve sütunlar veri üzerinde İzgesel Matris Ayrıştırması uygulandıktan sonra ayrı ayrı öbeklenmektedir. İki indis üzerinden öbekleme yöntemleri farklı veri grupları üzerinde test edilmektedir. Bu veri grupları, üretici bir Gauss modeli ile oluşturulmuş simule bir veri grubu, çeşitli hayvanlar ve özelliklerini içeren bir veri grubu, ülkeler arası ticaret ve diplomasi ilişkilerini gösteren ve beş farklı ağdan oluşan bir veri grubu, ve akciğer kanseri üzerinde bir mikro cihaz çalışmasına ait biyolojik veri grubu olmak üzere dört tanedir. İki indisli verilerde gerçek öbekler genelde tanımsız olduğundan, algoritmaların öbekleme performanslarını değerlendirmek için bağlantı tahmini kullanılmaktadır. Veri noktalarının bir kısmı rastsal olarak seçilip kaldırılmakta, ve bu noktalar aynı öbekteki noktaların benzer olması gerektiği bilgisine dayanılarak tahmin edilmektedir. Tanıttığımız yöntemlerin ilkinde bilgi kaybı olmaksızın verinin tümü kullanıldığından ilk yöntem daha hassas sonuç vermektedir. Buna karşılık ikinci yöntemde veri noktası sayısı önsel olarak oldukça azaltıldığından ikinci yöntemin zaman ve bellek karmaşıklığı çok daha düşüktür.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A	Symmetric adjacency matrix
a	First shape parameter of Beta Distribution
b	Second shape parameter of Beta Distribution
$\mathcal{B}$	Beta Distribution
$\mathcal{BE}$	Bernoulli Distribution
B_k	Bicluster submatrices in the data
$c_{-n,k}$	Element count of cluster k except the nth element itself
D_c	Bottom-right block of diagonal weight matrix
Dir	Dirichlet Distribution
$Disc$	Discrete Distribution
D_r	Top-left block of diagonal weight matrix
F	Distribution of data
$F(x_n \mid \theta)$	Data distribution conditioned on cluster parameters
G	Distribution of one-way cluster parameters
H	Base distribution of the Dirichlet Process
K	One-way cluster count - Two-way row cluster count
L	Two-way column cluster count
$\mathcal{L}$	Symmetric Laplacian matrix
M	Data matrix
N	One-way data entry count
$\mathcal{N}$	Gaussian (Normal) Distribution
P	Two-way data row count
p_k	Mixing proportion of kth cluster
$p(\boldsymbol{x})$	Marginal likelihood of data
$p(\theta \mid \boldsymbol{x}, \lambda_1, \lambda_2)$	Posterior distribution of cluster parameters
$p(\theta \mid \lambda_1, \lambda_2)$	Prior distribution of cluster parameters
Q	Two-way data column count
W	Diagonal weight matrix

$\boldsymbol{x}_{k,-n}$	Data assigned to cluster k except the nth element itself
x_n	One-way data entry
$\boldsymbol{x_p}$	Data in pth row of two-way data
$\boldsymbol{x_q}$	Data in qth column of two-way data
x_{pq}	Two-way data entry count
z_n	One-way cluster indicator
z_{-n}	One-way cluster indicators other than n_{th}
α	Concentration parameter of Dirichlet Process
δ	Dirac delta function
θ_n	One-way cluster parameter
θ_{-n}	One-way cluster parameters other than n_{th}
$\theta_n \mid \theta_{-n}$	Conditional distribution of a cluster parameter given others
λ	Hyperparameter of cluster parameters
μ	Mean of Gaussian Distribution
σ^2	Variance of Gaussian Distribution

LIST OF ACRONYMS/ABBREVIATIONS

1D	One Dimensional
2D	Two Dimensional
AUC	Area Under the Curve
AUROC	Area Under Receiver Operating Characteristics
CGS	Collapsed Gibbs Sampling
CRP	Chinese Restaurant Process
DP	Dirichlet Process
DPMM	Dirichlet Process Mixture Model
GS	Gibbs Sampling
MCMC	Markov Chain Monte Carlo
NMF	Nonnegative matrix factorization
NRMSE	Normalized Root Mean Square Error
NTF	Nonnegative tensor factorization
ROC	Receiver Operating Characteristics
SVD	Singular Value Decomposition

1. INTRODUCTION

Clustering is the task of splitting data into groups so that the data items in the same group are more similar to each other than to those in other groups. The main aim of clustering is to acquire a high inter-group similarity and a low intra-group similarity. The problem of grouping data having a common pattern together arises as an important problem in real world situations for ages. For instance, a supermarket might need to group its customers according to their shopping behaviours, and offer different special advantages to the different groups in order to increase the amount of the supermarket's income.

Whereas clustering is a practical tool for one-way data, it turns out to be inadequate for multiway data. Clustering attempts to find relations between objects that hold under all conditions whereas the relations may depend on different conditions. Therefore, it leads to diffuse results resulting in an inaccurate grouping. For instance, customers' behaviors may be different than each other in different days of a week and the supermarket may need to find out the customer behavior patterns in a week. Then we have a two-way data where one way corresponds to the customers and the other corresponds to the days, and we need to cluster the customers and the days, namely the rows and columns of the data simultaneously.

Today, many real world domains are relational, capturing relations among objects in different entities [1]. Several methods have been developed in recent years to extract meaningful information from data consisting of a set of objects related to each other in complex ways [2–6]. Many of those methods introduce parametric techniques, namely they assume the number of clusters to be known. Parametric models are advantageous in that they reduce the problem of estimating the probability density function to estimating the values of a small number of parameters. However, it is usually not possible to exactly know the number of clusters prior to observation in real-world data. Therefore, the assumptions on the number of clusters may incur a large error [7]. If an overly complex model –one with too many adjustable parameters– is constructed with

insufficient amounts of data, then the model will not only extract regularities in the data, but also use its extra capacity to model the randomness in the data. Such a model overfits the data. On the other hand, a model with too few parameters will underfit because it will not have enough capacity to learn all the regularities in the data [8]. In this sense, nonparametric methods are required and results in more accurate models fitting the data.

In this thesis, we introduce two non-parametric two-way clustering (biclustering) algorithms in detail, then apply and present their results on simulated and real-world data. Though the models and derivations given throughout the thesis seem to be applicable to two-way data, multiway procedures are very similar in that the data has solely more entities to process. Since row and column partitioning processes are in some sense independent from each other, what more entities bring out is just the need for higher time and space complexity.

We devote Chapter 2 for recalling the main concepts this thesis is based on. We also review the related literature in Chapter 2. In Chapter 3, we investigate the DPMM biclustering algorithm where the rows and columns are clustered simultaneously. In the same chapter, we also demonstrate the results of DPMM biclustering experiments. Afterwards in Chapter 4, we introduce the second set of biclustering algorithms we deal with in this thesis, namely the spectral algorithms. In the same way with Chapter 3, we explain the experiments on spectral algorithms and demonstrate their results at the end of Chapter 4. We also compare the two methods in Chapter 4. Lastly, Chapter 5 summarizes what we have done throughout the thesis and concludes the discussion.

2. BACKGROUND

In this chapter we introduce the basic concepts that would be needed for more advanced techniques introduced in the subsequent chapters. Together with the review of those concepts, we mention the related research in the literature and particularly their correspondence to the biclustering task. The concepts we review in this chapter include Clustering, Biclustering, Link Prediction, Dimensionality Reduction, Dirichlet Processes and Nonparametric Bayesian Methods.

2.1. Clustering

Given a $1 \times n$ or an $n \times 1$ matrix, that's a row or column vector M, clustering is the unsupervised task of grouping data items in M together so that the data items in the same group are alike each other. The data used for a clustering task may be multi-dimensional. However, it is composed of a single entity. Figure 2.1 shows an example row vector to be used in a clustering task, and Figure 2.2 demonstrates an example clustering operation.

Figure 2.1. An example row vector to be used in a clustering task. Although the data is 2D, it is one-way being composed of only one entity.

The clustering problem has been addressed in many contexts and by researchers in many disciplines for decades. This reflects its broad appeal and usefulness as one of the steps in data analysis and mining [9]. Clustering algorithms can be broadly divided into two groups: hierarchical and partitional. Hierarchical algorithms [10, 11] either start with each data point in its own cluster and merge the most similar pair of clusters successively to form a cluster hierarchy or start with all the data points in one cluster and recursively divide each cluster into smaller clusters. On the other hand, partitional clustering algorithms [12–14] find all the clusters simultaneously as a partition of the data and do not impose a hierarchical structure. [15]. One of the

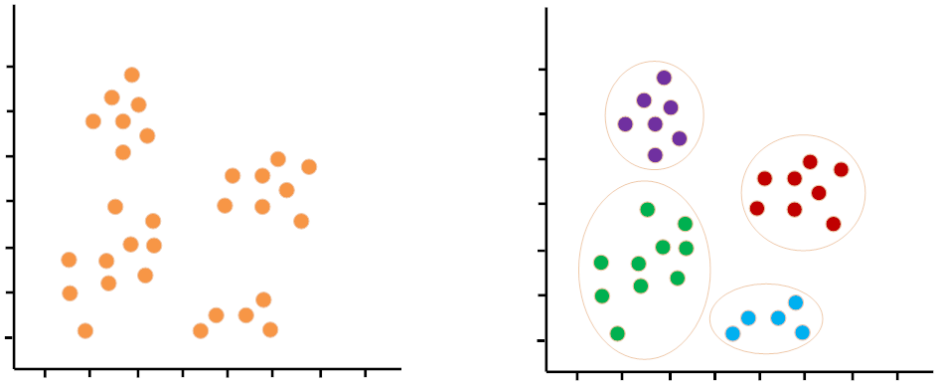


Figure 2.2. An example clustering task. Left: Unclustered 2D data. Right: Clustered data. There are four clusters in the data. Different colors (and circles) correspond to different clusters.

most well-known and frequently used clustering algorithm is K-means [13] which aims to cluster data into K groups in which each data item belongs to the group with the closest mean.

While clustering is a very useful tool for data analysis, it turns out to be inadequate when the observations to be clustered are composed of relational entries. Multiway clustering is the task of simultaneously cluster several entities of relational data. In the next section, we review biclustering, the task of clustering two-way data, for which we investigate two techniques throughout this thesis.

2.2. Biclustering

Given a $p \times q$ matrix M, biclustering (a.k.a. co-clustering or two-way clustering) is to cluster rows and columns of M simultaneously and find submatrices B_k such that the entries in each B_k are similar in some sense to one another. An optimal biclustering operation maximizes the mutual information between the clustered random variables subject to constraints on the row and column cluster counts [5]. The data to be used for a biclustering task is composed of two entities. However it would still be multi-dimensional. Figure 2.3 represents an example data matrix that would be used in a biclustering task.

(x_{11}, y_{11})	(x_{12}, y_{12})	.....	$(x_{1(q-1)}, y_{1(q-1)})$	(x_{1q}, y_{1q})
(x_{21}, y_{21})	(x_{22}, y_{22})	.....	$(x_{2(q-1)}, y_{2(q-1)})$	(x_{2q}, y_{2q})
.....	.....	.....	.....	.....
$(x_{(p-1)1}, y_{(p-1)1})$	$(x_{(p-1)2}, y_{(p-1)2})$	.....	$(x_{(p-1)(q-1)}, y_{(p-1)(q-1)})$	$(x_{(p-1)(q-1)}, y_{(p-1)(q-1)})$
(x_{p1}, y_{p1})	(x_{p2}, y_{p2})	.....	$(x_{p(q-1)}, y_{p(q-1)})$	(x_{pq}, y_{pq})

Figure 2.3. An example data matrix to be used in a biclustering task. The data is 2D, as well as it is two-way being composed of rows and columns.

There are different approaches for the arrangement of the bicluster submatrices in the base matrix. Where most of the work in biclustering literature assume each row or column belongs to a unique cluster [3, 16, 17], there are some work assuming that a matrix entry may belong to several different biclusters [6]. A different approach may be the existence of some background matrix entries that do not belong to any biclusters, and its alternative is that each matrix entry belongs to at least one cluster. Figure 2.4 represents some examples to those structures. [17] is a recent work utilizing stochastic block modeling and generates bicluster structures like the grid-base paintings of Piet Mondrian as in (c) in Figure 2.4. We, in this work, model the data assuming that each matrix entry belongs to exactly one cluster and the inference procedures result in a representation like (d) in Figure 2.4. That's, the biclusters form contiguous blocks in the base matrix which results in a checkerboard structure.

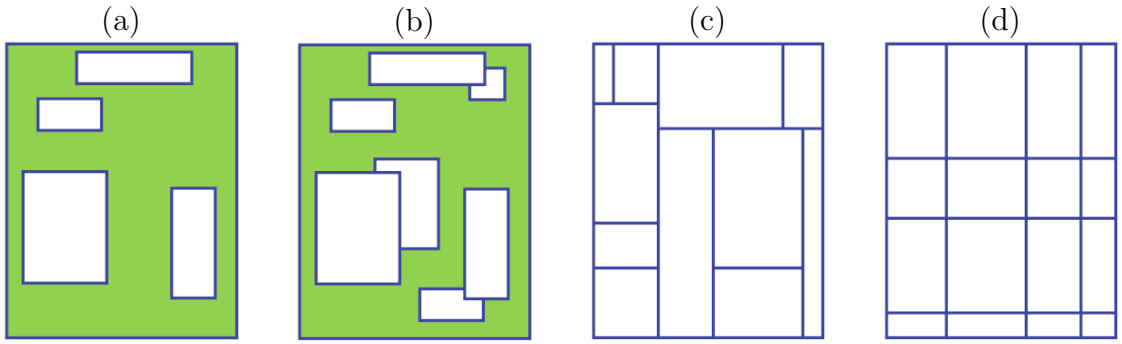


Figure 2.4. Different bicluster structures. Figure is based on [18].

Figure 2.5 shows an input data matrix and the one-way clustering results with respect to rows and columns and the biclustering result respectively. The rows and

columns are rearranged in the result so that the rows/columns in the each bicluster are contiguous and so the partitions are more visible. The difference in bicluster colors stand for difference in the magnitude of the corresponding bicluster parameter.

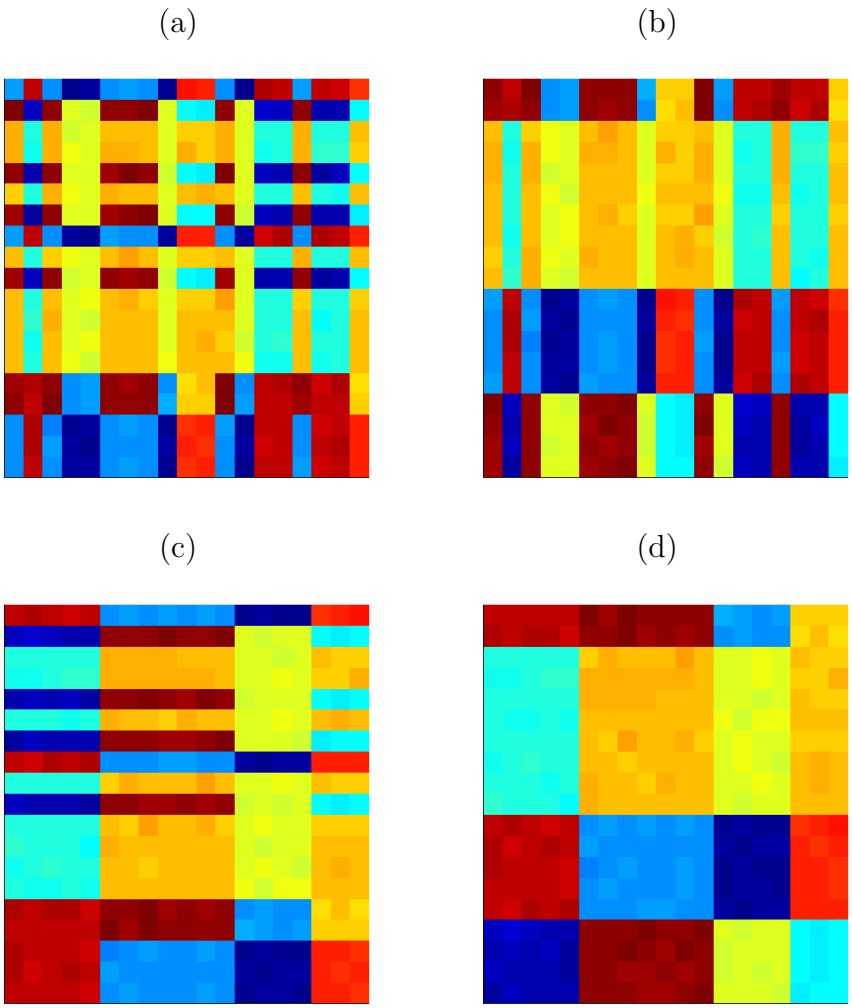


Figure 2.5. An example biclustering. **(a)** The original data. **(b)** Result of clustering in one-way with respect to rows. **(c)** Result of clustering in one-way with respect to columns. **(d)** Biclustering result. Rows and columns are clustered simultaneously.

The concept of simultaneous clustering of rows and columns of a data matrix was initially introduced in seventies by Hartigan [19]. Since then many different models and approaches have been developed for this frequent problem in machine learning. In [18] and very recently in [20], authors present a comprehensive survey of the models, methods and applications developed to solve the biclustering problem. Whereas the problem of biclustering arises in many different fields such as text mining, social network analysis, reommendation systems [2,21,22], most of the algorithms in the literature were

introduced in order to partition microarray data of genes and conditions [3, 23–27]. An entry M_{ij} of the microarray data correspond to the expression of the $gene_i$ under $condition_j$. The aim is then to find the co-regulated genes as well as the co-related conditions. For instance, genes in a bicluster may involve in the emerge of the lung cancer in some specific conditions whereas the genes in another bicluster may result in leukemia in some other conditions. Finding out those gene/condition groups has a significant role in discovery of causes and/or treatments of metabolic processes and/or diseases.

2.3. Link Prediction

Links correspond to the relationships between entities in a network. Not all of those links are observed in many cases, and predicting the existence of the missing links, in other words *completing* the links, is an interesting and fundamental problem in analysis of networks appearing in many different fields [28–30]. For instance, in a network of suspects, we might want to identify which of those individuals are probable to be involved in a certain terrorist activity where some of them are already known to be convict for the corresponding type of criminal activities. Figure 2.6 demonstrates the link prediction problem.

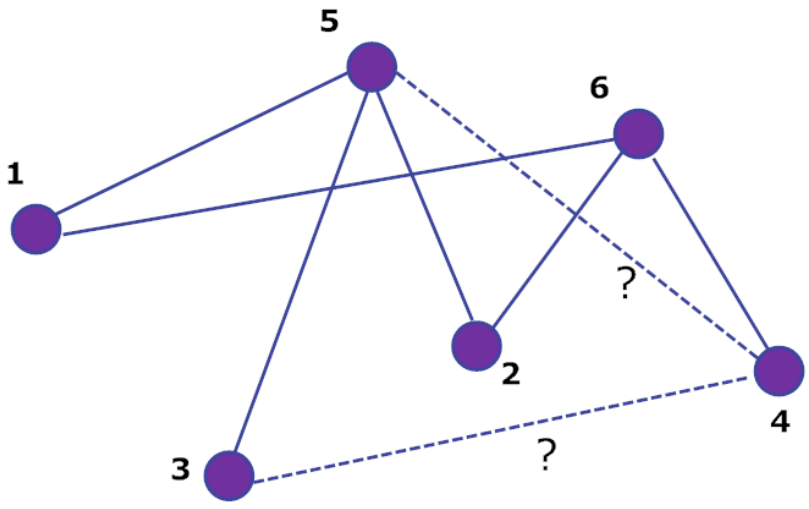


Figure 2.6. An example link prediction problem. The problem is the edges with question marks actually exist or not.

Link prediction problem is often viewed as a simple binary classification problem; for any two potentially linked objects o_i and o_j , predict whether l_{ij} is zero or one. One approach is to do the link prediction entirely based on structural properties of the network, whereas another one is to make use of attribute information for link prediction [31]. Biclustering-based link prediction is an example to the former approach. Since the data items in a partition are assumed to have similar properties, existence/non-existence of the missing links in a bicluster can easily be predicted based on the observed links in the same partition [32]. Figure 2.7 summarizes this approach.

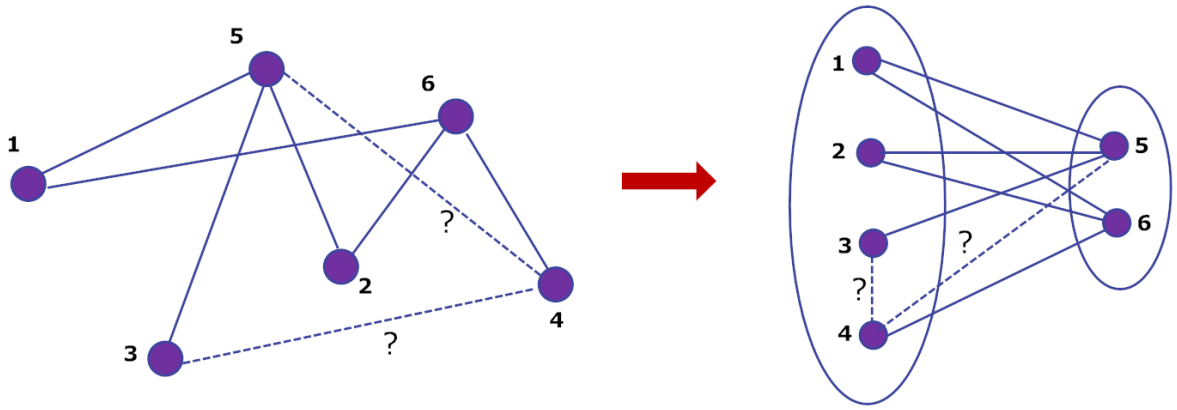


Figure 2.7. Solution of link prediction problem using bipartite graphs. The probability that there is an edge between nodes 3 and 4 is very low since they are in the same partition whereas it is very probable that there is an edge between nodes 4 and 5 since there are many edges between the two partitions they belong to.

Another common usage of link prediction, which we also utilize in this thesis, is to measure the performance of data partitioning methods [16]. For this purpose, some data entries are randomly selected and marked as missing, then they are predicted and the prediction performance is calculated using the complete data as the ground truth. Since, there are very few available real world datasets annotated with ground truth biclusters, this approach is a good means to evaluate the performance of biclustering techniques.

2.4. Dimensionality Reduction

In the field of linear algebra, the term *matrix decomposition* stands for the representation of a given matrix in some canonical form, which is in general involve product of newly introduced simpler matrices. By use of matrix decomposition, data can be represented in fewer dimensions with little or no loss of information. When data can be explained with fewer features, we get a better idea about the process that underlies the data, and create simpler and therefore more robust models for classification or regression purposes. Furthermore, dimensionality reduction decreases the time and space complexity of the inference algorithms [7].

Utilization of spectral decomposition for clustering multiway data is a frequent approach in machine learning literature [2, 3]. Figure 2.8 summarizes the method. In [2], the method is applied on word-document matrices. However, the authors here assume that there are the same number of clusters for row and column data. This being impractical in most of the real-world datasets, in [3], developed spectral biclustering algorithm is capable of clustering rows and columns of a gene-condition expression matrix into different number of clusters.

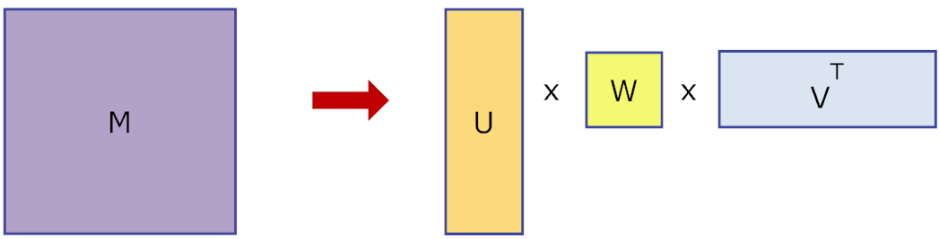


Figure 2.8. Dimensionality reduction operation. Left: The original matrix with high dimensionality. Right: The resulting simpler matrices with lower dimensions. Generally, U and V^T are used to cluster rows and columns of relational data respectively.

There are also methods in the machine learning literature utilizing NMF and NTF to retrieve clusters in different ways of a data matrix. [33–35]. Similar to the spectral methods, tensor-factorization based methods assume a symmetric affinity matrix which is to serve as the input to the process of assigning points to different clusters. Spectral

methods and NMF are discussed and proved to be equivalent in [36].

When the base matrix entries are all binary, we can think of biclustering as a bipartite graph partitioning problem. This approach is useful for spectral biclustering method which is introduced in Chapter 4. We give the computational details of spectral matrix decomposition operation in Chapter 4.

2.5. Dirichlet Process

Dirichlet Process (DP), which was firstly introduced by Ferguson in 1973 [37], is a non-parametric statistical method to better model complex and realistic data in real-world domains without the restriction of the *parameter count*. Similar to the other non-parametric methods, it favours models whose complexity increases as the dataset grows.

We can think of a Dirichlet Process as an infinite-dimensional generalization of the Dirichlet distribution, which is a distribution over distributions. Consider a probability distribution H and a partition $(A_1, ..., A_n)$ on the domain of this distribution. G is also a distribution that is distributed according to a Dirichlet Process with base distribution H and concentration parameter α if

$$(G(A_1), ..., G(A_n)) \ \sim \ Dir(\alpha H(A_1), ..., \alpha H(A_n)) \tag{2.1}$$

We show the above explanation formally as $G \sim DP(\alpha, H)$. Distributions drawn from a DP are discrete, but cannot be described using a finite number of parameters, thus the measure G is precisely discrete although the partitioned distribution H is continuous [38]. Most of the real-world applications are discrete, and what makes DPs convenient for many modelling purposes is the discrete measure it brings out. In fact, it would not be a good idea to model continuous observations with DPs since it would be too restrictive. Figure 2.9 demonstrates a base distribution H and a sample distribution G generated from a Dirichlet based on H.

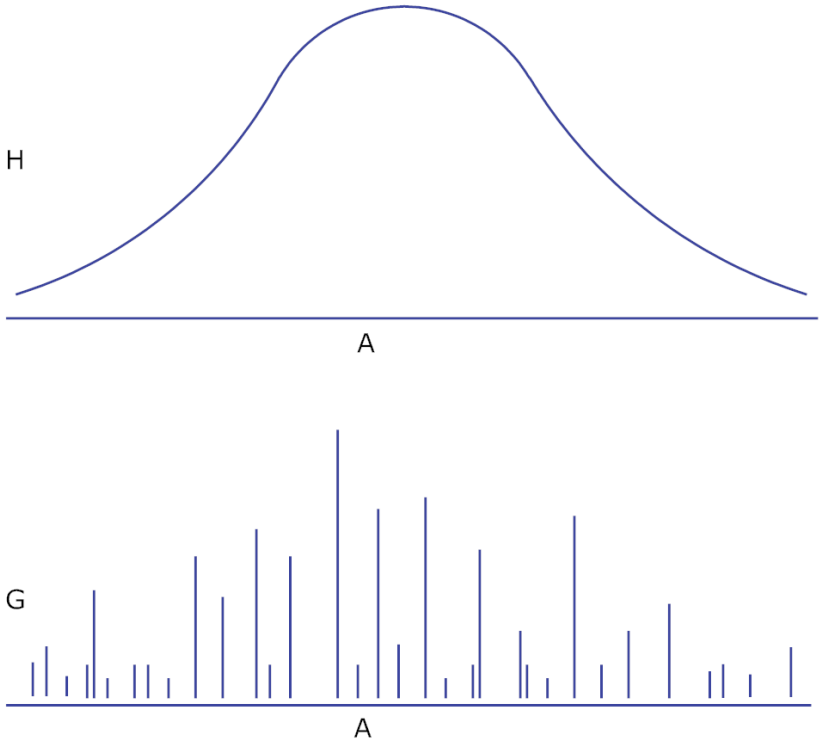


Figure 2.9. An example Dirichlet distribution. Above: The base distribution H in domain A. Below: A sample distribution G in the same domain A generated from a DP. Since DPs are distributions over distributions, a sample from a DP is itself a DP. Figure based on [39].

A nice analogy to discreteness of DP was introduced in [40]. In this article, the author states the *stick breaking construction*. Assume that we have a stick of unit length and split it into two pieces with respect to Beta distribution. Let the length of the first and second split is d_1 and d_2 respectively. In the second step, let we break the second split of the first step into two, and repeat this process infinitely. Then, the lengths d_i of created countably infinite splits are as if drawn from a DP.

In addition to being discrete, one of the most valuable characteristics of samples from a DP is that they exhibit a clustered structure. This scheme was introduced in [41]. Assume balls of different colors exist in an urn and we equiprobably draw a ball from that urn with replacement. And at each replacement, a new ball with the same color as the drawn one is added to the urn. Then the colors correspond to different clusters and the cluster probabilities are conditioned to one another. This

scheme creates the main idea behind DPMM that we introduce in Section 2.6.

The Pólya urn scheme is usually mentioned together with the Chinese Restaurant Process (CRP) which is frequently used as an analogy while explaining the DPs. Think of a restaurant where each customer is seated at a currently occupied table with a probability proportional to the number of customers n_k in that kth table, or at an empty table with a probability proportional to a constant α. Then, if there are countably infinite number of tables with countably infinite capacity, then there would be countably infinite number of customers seated, and the tables and customers are considered to be the mixture clusters and samples respectively. Moreover, the constant α correspond to the concentration parameter in DP representation in Equation 2.1. Figure 2.10 exemplifies the Chinese Restaurant Process.

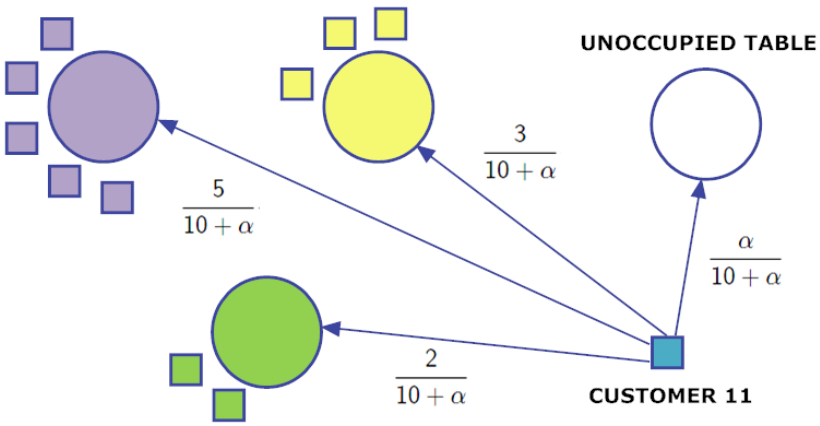


Figure 2.10. An example Chinese Restaurant Process. The probabilities of the 11th customer to be seated on one of the three occupied tables or an unoccupied table is shown on the corresponding edges.

There are many extensions to DPs in the literature such as hierarchical DPs [42], Pólya trees [43], Dirichlet diffusion trees [44], Indian Buffet Processes [45] and Pitman Yor Processes [46].

Apparently, the concentration parameter is an important factor affecting the number of clusters making up the DP samples. In [47], this is investigated in detail. We also consider and test the correlation in our experiments on real datasets.

2.6. Nonparametric Bayesian Methods

Assume we are trying to model people's patterns of movie preferences. Believing there are groups of people with similar preferences, it is reasonable to use a mixture model. A Bayesian approach may be putting a prior on some parameters defining the clusters, and calculate the posterior of those parameters using Bayesian inference rule while we observe real movie preference data. Bot how to select the number of priors, that's the number of clusters? We are possibly unlikely to know number of possible movie preference groups prior to observing tha data. Even if there are a few well defined groups, how to select the distribution of the parameters we measure is still problematic. To model complicated distributions for example, we might need many Gaussians for each cluster rather than one for each. That's, Bayesian methods based on finite parameters of prior belief is inadequate in such a case [39].

A nonparametric model is in fact a parametric model where the number of parameters increases with data. Therefore, they are free of a hard restriction over data, namely the restriction over the number of parameters. They let data to speak for itself [48].

Due to their countably infinite, discrete and clustered nature, DPs that we reviewed in Section 2.1 are reasonable priors for mixture models. This idea was first introduced in [49] and have been used for flexible modeling of mixtures for decades.

2.6.1. Dirichlet Process Mixture Models

Consider an exchangeable dataset points $x_1, ..., x_N$, some of which can be grouped together creating a number of groups $<= N$. In order to figure out the groups in this kind of data, we can model the data points as if they come from a discrete distribution $F(\theta)$ and figure out the mixing parameters θ in order to decide which data point should be assigned to which group. Since we do not have an idea about how many clusters exist in the dataset, we cannot know the number of different values that could be assigned to parameters θ. However, there are always a finite number of groups in the

data. Therefore, we can model the mixing parameters θ to be drawn from a DP with a continuous base distribution H and concentration parameter α, which would result in a discrete distribution G of the parameters. Below is the formal representation of such a model where N corresponds to the number of the observation points.

$$\begin{aligned} x_1, .., x_N \mid \theta &\sim F(\theta_n) \\ \theta_1, .., \theta_N &\sim G \\ G &\sim DP(\alpha, H) \end{aligned} \tag{2.2}$$

When G is integrated out in Equation 2.2, the representation in Equation 2.3 is retrieved for the distribution of θ_n conditioned on the values of other parameters θ_{-n}, namely $(\theta_1, .., \theta_{n-1}, \theta_{n+1}, .., \theta_N)$.

$$\theta_n \mid \theta_{-n} \sim \frac{1}{N-1+\alpha} \sum_{i \neq n} \delta_{\theta_i \theta_n} + \frac{\alpha}{N-1+\alpha} H(\theta_n) \tag{2.3}$$

Using such a model, it is possible to update only one observation parameter at each iteration which increases the time of convergence. Another version of a DPMM may be formed as below to avoid this problem. Here in this model, new parameters, namely the latent cluster indicators and Dirichlet-distributed cluster assignment probabilities are introduced, and K corresponds to the number of clusters in the dataset.

$$\begin{aligned} x_1, .., x_N \mid z_1, .., z_N, \theta_1, .., \theta_K &\sim F(\theta_{z_n}) \\ z_1, .., z_N \mid p_1, .., p_K &\sim Disc(p_1, .., p_K) \\ p_1, .., p_K &\sim Dir(\alpha/K, .., \alpha/K) \\ \theta_1, .., \theta_K &\sim H \end{aligned} \tag{2.4}$$

Similar to the operation in the first model, one can integrate out the mixing proportions p_k in this second model. Then, below distribution of each cluster indicator conditioned on other cluster indicators can be retrieved, where $c_{-n,k}$ corresponds to

the count of the elements in the cluster number k, except the element x_n itself.

$$z_n = k \mid z_{-n} \;\sim\; \frac{c_{-n,k} + \frac{\alpha}{K}}{N - 1 + \alpha} \tag{2.5}$$

Because an infinite number of parameters is assumed in this non-parametric model, the variable K can be eliminated as one let $K \to \infty$. Then below limiting values will be retrieved for the distribution of cluster indicators.

$$\begin{aligned} P(z_n = k \mid z_{-n}) &\to \frac{c_{-n,k}}{N - 1 + \alpha} \\ P(z_n \neq any\ z_{-n} \mid z_{-n}) &\to \frac{\alpha}{N - 1 + \alpha} \end{aligned} \tag{2.6}$$

The distributions above in Equation 2.6 correspond to the one shown in Equation 2.3 when one take $\theta_n = \theta_{c_k}$. So the two mentioned models are equivalent in that $K \to \infty$ in the latter. This limiting behaviour of Dirichlet Process Mixture Models (a.k.a Infinite Mixture Models)results in discreteness and clustered structure of the DP samples which we explain with analogies in the literature in Section 2.5.

2.6.2. Inference Methods for DPMM

In order to make an inference on the parameters, what we have to do is to compute the posterior expectation of the parameters at each observed data point. This computation is usually intractable with regular inference methods. However, Monte Carlo methods, which are used for sampling from the posterior when exact posterior is infeasible to compute, is useful in this context.

Especially when conjugate priors are used, Gibbs Sampling methods is an appropriate and efficient way to infer the parameters from DPMM. In [50], the authors introduce three Gibbs Sampling methods which are ordered in the article from the less to most efficient. We summarize them followingly. The most efficient algorithm is presented to be the Collapsed Gibbs Sampling, which we use in our applications and

experiments.

Using the first model given in Equation 2.2, when the prior given in Equation 2.3 is combined with the likelihood $F(x_n \mid \theta)$ of the data, below posterior distribution is retrieved for the parameters. A parameter value will be sampled from this posterior on each observed data point.

$$\theta_n \mid \theta_{-n} \;\propto\; \sum_{i \neq n} F(x_n \mid \theta_i)\; \delta_{\theta_i \theta_n} + \alpha\; p(\theta_n \mid x_n) \int_{\theta} F(x_n \mid \theta) H(\theta) d\theta \tag{2.7}$$

Proportionality operator is used in Equation 2.7 because the common denominators $N - 1 + \alpha$ in Equation 2.3 are removed and also the likelihood introductions have most probably impaired the discrete distribution of the probabilities. So, normalization is required on the posterior distribution before sampling.

Using the second model given in Equation 2.4, when the limiting prior probabilities as $K \to \infty$ given with Equation 2.6 are combined with the likelihood $F(x_n \mid \theta)$ of the data, below posterior distributions are retrieved for the parameters:

$$\begin{aligned} P(z_n = k \mid z_{-n}, x_n, \theta) &\quad\propto\quad c_{-n,k} F(x_n \mid \theta_k) \\ P(z_n \neq any\ z_{-n} \mid z_{-n}, x_n, \theta) &\quad\propto\quad \alpha \textstyle\int_{\theta} F(x_n \mid \theta) H(\theta) d\theta \end{aligned} \tag{2.8}$$

Above in Equation 2.8, again proportionality operators are used because the common denominators $N - 1 + \alpha$ in Equation 2.6 are removed and also the likelihood introductions have most probably impaired the discrete distribution of the probabilities.

Going a step further, we can only draw samples from the conditionals of the parameters that we are interested, and omit others from the conditionals. This is called Collapsed Gibbs Sampling.

In DPMM with latent indicators, we are only interested in the information that

which data point is assigned to which cluster, that's, what we want to know is the values of $\boldsymbol{z}$. Therefore, applying CGS to the model above requires to omit θ from the posteriors.

In the second part of Equation 2.8, we have already omitted θ by taking the integral of joint distribution over θ. What we have to deal with is the case in which we assign a data point to an already existing cluster. Here, we need the probability $p_{k,-n}(\theta) = p(\theta \mid \boldsymbol{x}_{k,-n})$ as the prior of θ and combine it with the likelihood. So, Equation 2.8 turns into;

$$\begin{aligned} P(z_n = k \mid z_{-n}, x_n, \theta) &\propto c_{k,-n} \textstyle\int_\theta F(x_n \mid \theta) p(\theta \mid \boldsymbol{x}_{k,-n}) d\theta \\ P(z_n \neq any\ z_{-n} \mid z_{-n}, x_n, \theta) &\propto \alpha \textstyle\int_\theta F(x_n \mid \theta) H(\theta) d\theta \end{aligned} \tag{2.9}$$

Now, we need to compute the two integrals in Equation 2.9, the CGS inference.

The second one is simpler. It requires direct integration of the likelihood and the posterior. And if they are chosen as conjugate distributions, the result would be easily calculated. The first integral similarly requires integration of product of two distributions. But this time, the second item of this product is not the pure prior, but it is actually the posterior of θ after observing the data $\boldsymbol{x}_{k,-n}$. Again, conjugate priors make it easy to calculate this product.

Integration of product of two distributions simply corresponds to the *evidence* in Bayesian formula. So, first, the posterior would be estimated. Then, the evidence would be acquired by calculating the normalization constant, namely by dividing the product of likelihood and prior to the posterior density.

In the next section, those steps are demonstrated based on exponential family and conjugate distribution concepts.

2.6.3. Obtaining Predictive Likelihoods with Conjugate Priors

Let the observation likelihood is modelled by the below exponential family:

$$p(x_n \mid \theta) = \exp(f(\theta)t(x_n) - h_l(x_n) - a_l(\theta)) \tag{2.10}$$

and model parameter prior is modelled by its conjugate exponential family, that's:

$$p(\theta \mid \lambda_1, \lambda_2) = \exp(\lambda_1 f(\theta) + \lambda_2(-a_l(\theta)) - a_c(\lambda_1, \lambda_2)) \tag{2.11}$$

Here, the hyperparameter $\lambda = (\lambda_1, \lambda_2)$ is of dimension $\dim(\theta) + 1$ and the sufficient statistics are $(f(\theta), -a_l(\theta))$. Then, using Bayesian inference, it can be derived that the posterior of the parameter θ given the observations $\boldsymbol{x} = (x_1, .., x_N)$ is from the same exponential family as the prior where

$$\begin{aligned} \hat{\lambda}_1 &= \lambda_1 + \textstyle\sum_{n=1}^{N} t(x_n) \\ \hat{\lambda}_2 &= \lambda_2 + N \end{aligned} \tag{2.12}$$

Therefore;

$$p(\theta \mid \boldsymbol{x}, \lambda_1, \lambda_2) = \exp(\hat{\lambda}_1 f(\theta) + \hat{\lambda}_2(-a_l(\theta)) - a_c(\hat{\lambda}_1, \hat{\lambda}_2)) = p(\theta \mid \boldsymbol{x}, \hat{\lambda}_1, \hat{\lambda}_2) \tag{2.13}$$

Now, what we have to do is to compute the integral in the first part of Equation 2.9 by calculating the evidence, that's, the marginal likelihood in the Bayesian formula. Adapting Bayesian formula in our problem results in:

$$p(\boldsymbol{x}) = \frac{(\prod_n p(x_n \mid \theta))p(\theta \mid \lambda_1, \lambda_2)}{p(\theta \mid \boldsymbol{x}, \lambda_1, \lambda_2)} \tag{2.14}$$

Substituting Equations 2.10, 2.11 and 2.13 into Equation 2.14, what we get for the

marginal likelihood of $\boldsymbol{x}$ is:

$$\begin{aligned} p(\boldsymbol{x}) &= \exp(a_c(\hat{\lambda}_1, \hat{\lambda}_2) - a_c(\lambda_1, \lambda_2) - \textstyle\sum_n h_l(x_n)) \\ &= \exp\left(a_c(\lambda_1 + \textstyle\sum_{n=1}^{N} t(x_n), \lambda_2 + N) - a_c(\lambda_1, \lambda_2) - \textstyle\sum_n h_l(x_n))\right) \end{aligned} \tag{2.15}$$

and

$$\begin{aligned} p(x_n) &= \exp(a_c(\hat{\lambda}_1, \hat{\lambda}_2) - a_c(\lambda_1, \lambda_2) - h_l(x_n)) \\ &= \exp\left(a_c(\lambda_1 + t(x_n), \lambda_2 + 1) - a_c(\lambda_1, \lambda_2) - h_l(x_n))\right) \end{aligned} \tag{2.16}$$

While Equation 2.15 is being calculated, it can be seen that the first term inside of exp is due to $p(\theta \mid \boldsymbol{x}, \lambda_1, \lambda_2)$, the second term is due to the prior $p(\theta \mid \lambda_1, \lambda_2)$, and lastly the third term is due to $\prod_n p(x_n \mid \theta)$. Considering this, the posterior prediction of x_n given $x_{k,-n}$ can be found out to be as below:

$$p(x_n \mid \boldsymbol{x_{k,-n}}) = \exp(a_c(\hat{\lambda}_1 + t(x_n), \hat{\lambda}_2 + 1) - a_c(\hat{\lambda}_1, \hat{\lambda}_2) - h_l(x_n)) \tag{2.17}$$

since

$$p(x_n \mid \boldsymbol{x}_{k,-n}) = \int_{\theta_k} p(x_n \mid \theta_k) p(\theta_k \mid \boldsymbol{x}_{k,-n}) d\theta_k = \frac{(x_n \mid \theta_k) p(\theta_k \mid \boldsymbol{x}_{k,-n})}{p(\theta_k \mid \boldsymbol{x}_{k,-n}, x_n)} \tag{2.18}$$

Here, the posterior $p(\theta_k \mid \boldsymbol{x}_{k,-n})$ calculated in Equation 2.13 becomes the new prior.

3. NONPARAMETRIC BAYESIAN BICLUSTERING

In this chapter, we explain how to retrieve the biclusters in the data by making use of DPMM and CGS directly on the row and column entities of the original data matrix. We also demonstrate the results of the experiments we performed to test the method.

3.1. DPMM for Biclustering

In Section 2.6, we presented DPMM for one way data. Now, we extend the idea to biclustering and create a compact model to do inference in the original two dimensional space of the data.

The aim of a biclustering or coclustering task is to cluster rows and columns simultaneously, where the parameter matrix is 2D. Below is the DP mixture model for Bayesian Biclustering. We base it on the one dimensional model we introduced in Equation 2.9. It contains the cluster indicator vectors $\boldsymbol{g}$ and $\boldsymbol{c}$, and mixing proportions $\boldsymbol{p}$ and $\boldsymbol{q}$ for row and columns respectively. In addition, the bicluster parameters that we think of as infinite in the nonparametic model are in $\boldsymbol{\theta}$. $\boldsymbol{\theta}$ is a 2D matrix in this case and each entry contains the cluster parameter for a bicluster. α and β are the Dirichlet Process parameters for row and column processes respectively.

$$\begin{aligned} x_{pq} \mid g_p, c_q, \theta_{kl} &\sim F(\theta_{g_p c_q}) \\ g_p \mid p_k &\sim Disc(p_1, .., p_K) \\ c_q \mid p_l &\sim Disc(p_1, .., p_L) \\ p_k &\sim Dir(\alpha/K, .., \alpha/K) \\ p_l &\sim Dir(\beta/L, .., \beta/L) \\ \theta_{kl} &\sim H \end{aligned} \tag{3.1}$$

One can integrate out the mixing proportions p_k and p_l in this model. Then,

below distribution of each cluster indicator conditioned on other cluster indicators can be retrieved for rows and columns, where $n_{-p,k}$ correspond to the count of all entries in the rows with cluster number k except the row x_{pq} itself exists, and $n_{-q,l}$ to the count of all entries in the columns with cluster number l except the column x_{pq} itself exists. P and Q are the row and column counts, K and L are row and column cluster counts respectively.

$$\begin{aligned} g_p = k \mid g_{-p} \quad &\sim \quad \frac{n_{-p,k} + \frac{\alpha}{K}}{P - 1 + \alpha} \\ c_q = l \mid c_{-q} \quad &\sim \quad \frac{n_{-q,l} + \frac{\alpha}{L}}{Q - 1 + \alpha} \end{aligned} \tag{3.2}$$

Because an infinite number of parameters is assumed in this non-parametric model, the variables K and L can be eliminated as one let $K \to \infty$ and $L \to \infty$. Then below limiting values will be retrieved for the distribution of row and column cluster indicators.

$$\begin{aligned} P(g_p = k \mid g_{-p}) \quad &\to \quad \frac{n_{-p,k}}{P - 1 + \alpha} \\ P(g_p \neq any\ g_{-p} \mid g_{-p}) \quad &\to \quad \frac{\alpha}{P - 1 + \alpha} \\ P(c_q = l \mid c_{-q}) \quad &\to \quad \frac{n_{-q,l}}{Q - 1 + \beta} \\ P(c_q \neq any\ c_{-q} \mid c_{-q}) \quad &\to \quad \frac{\beta}{Q - 1 + \beta} \end{aligned} \tag{3.3}$$

3.2. Inference Methods

The inference methods for Bayesian statistics can mainly be grouped into two. First group is MCMC inference algorithms which aim to estimate the real value of the model parameters based on sampling strategies, and the second one is the variational inference methods [51, 52] that formulate the marginal or conditional probability of

data in terms of an optimization problem.

MCMC methods provide a systematic approach to the computation of likelihoods and posterior distributions and permit the deployment of Bayesian methods in a rapidly growing number of applied problems [53]. In this thesis, we utilize MCMC methods. Followingly the MCMC inference for biclustering DPMM is formulated.

When the limiting prior probabilities as $K \to \infty$ and $L \to \infty$ given in Equation 3.3 are combined with the likelihood $F(x_{pq} \mid \theta)$ of the data an then integrated over bicluster parameters θ, we get rid of the requirement for sampling θ in each MCMC iteration. This type of GS perspective is mentioned in Section 3.5, and called Collapsed Gibbs Sampling (CGS). In Equation 3.4, the update equations for the row and column indicators are shown. The integrals are the posteriors from which samples are drawn at each iteration, and due to the integration over θ, they are in fact independent from θ.

$$\begin{aligned} P(g_p = k \mid g_{-p}, x_{pq}, \theta) &\propto n_{-p,k} \textstyle\prod_q \int_\theta F(x_{pq} \mid \theta) p(\theta \mid \boldsymbol{x}_{pq,-p}) d\theta \\ P(g_p \neq any\ g_{-p} \mid g_{-p}, x_{pq}, \theta) &\propto \alpha \textstyle\prod_q \int_\theta F(x_{pq} \mid \theta) H(\theta) d\theta \\ P(c_q = l \mid c_{-q}, x_{pq}, \theta) &\propto n_{-q,l} \textstyle\prod_p \int_\theta F(x_{pq} \mid \theta) p(\theta \mid \boldsymbol{x}_{pq,-q}) d\theta \\ P(c_q \neq any\ c_{-q} \mid c_{-q}, x_{pq}, \theta) &\propto \beta \textstyle\prod_p \int_\theta F(x_{pq} \mid \theta) H(\theta) d\theta \end{aligned} \tag{3.4}$$

Now, we need to compute the predictive and posterior predictive likelihoods of data based on row and column indicators given in Equation 3.4.

Integration of product of two distributions simply corresponds to the *evidence* in Bayesian formula. So, first, the posterior is to be estimated. Then, the evidence is to be acquired by calculating the normalization constant, namely by dividing the product of likelihood and prior to the posterior density. When we select data likelihood and parameter prior distributions from the set of exponential family distributions and conjugate to each other, then the calculations become simpler.

Let the observation likelihood is modelled by the below exponential family:

$$p(x_{pq} \mid \theta) = \exp(f(\theta)t(x_{pq}) - h_l(x_{pq}) - a_l(\theta)) \tag{3.5}$$

and model parameter prior is modelled by its conjugate exponential family, that's:

$$p(\theta \mid \lambda_1, \lambda_2) = \exp(\lambda_1 f(\theta) + \lambda_2(-a_l(\theta)) - a_c(\lambda_1, \lambda_2)) \tag{3.6}$$

Then, the inference steps are similar to the 1D case except that we have the restriction that elements in a row are not independent from each other and all have to be assigned to the same row cluster. Similarly, all data entries in the same column have to be in the same column cluster. In order not to repeat the derivations, based on Equation 2.16, newly added row cluster probabilities are

$$\begin{aligned} p(\boldsymbol{x}_p) &= \textstyle\prod_q \exp\left(a_c(\hat{\lambda}_1, \hat{\lambda}_2) - a_c(\lambda_1, \lambda_2) - h_l(x_{pq})\right) \\ &= \textstyle\prod_q \exp\left(a_c(\lambda_1 + t(x_{pq}), \lambda_2 + 1) - a_c(\lambda_1, \lambda_2) - h_l(x_{pq}))\right) \end{aligned} \tag{3.7}$$

and newly added column cluster probabilities are

$$\begin{aligned} p(\boldsymbol{x}_q) &= \textstyle\prod_p \exp\left(a_c(\hat{\lambda}_1, \hat{\lambda}_2) - a_c(\lambda_1, \lambda_2) - h_l(x_{pq})\right) \\ &= \textstyle\prod_p \exp\left(a_c(\lambda_1 + t(x_{pq}), \lambda_2 + 1) - a_c(\lambda_1, \lambda_2) - h_l(x_{pq}))\right) \end{aligned} \tag{3.8}$$

Similarly, if we use Equation 2.17 and apply our restriction mentioned above, we get the probabilities for existing row clusters as

$$p(\boldsymbol{x}_p \mid \boldsymbol{x}_{k,-p}) = \prod_q \exp(a_c(\hat{\lambda}_1 + t(x_{pq}), \hat{\lambda}_2 + 1) - a_c(\hat{\lambda}_1, \hat{\lambda}_2) - h_l(x_{pq})) \tag{3.9}$$

and for existing column clusters as

$$p(\boldsymbol{x}_q \mid \boldsymbol{x}_{k,-q}) = \prod_p \exp(a_c(\hat{\lambda}_1 + t(x_{pq}), \hat{\lambda}_2 + 1) - a_c(\hat{\lambda}_1, \hat{\lambda}_2) - h_l(x_{pq})) \tag{3.10}$$

. In Equations 3.9 and 3.10, the posterior $p(\theta_k \mid \boldsymbol{x}_{k,-p})$ becomes the new prior.

3.3. Experiments

In order to evaluate the performance of the DPMM Biclustering algorithm discussed, we do experiments on two simulated and two real datasets. We model the datasets with different statistical models using conjugate exponential family distributions.

3.3.1. Materials and Methods

We test the DPMM algorithm on four different datasets. We model two of them as Gaussian and the other two as Bernoulli. There are one simulated (toy) dataset and one real dataset in both Gaussian and Bernoulli groups.

First dataset is a Gaussian toy dataset we generated using the parameters displayed in Table 3.1.

Table 3.1. Toy data generation parameters.

Parameter	**Value**
Distribution	Gaussian
Dimensions	160×160
${\sigma_{data}}^2$	1
${\sigma_{clusters}}^2$	10
$\lambda_{clusters}$	0

Second dataset we utilize is a binary relational dataset containing different animals and their attributes. It is a toy dataset created by the authors of [54]. We binarized the data and removed attributes and samples unrelated to the animals. The resulting dataset contains 33 different animals and 15 different attributes and the resulting 33×15 data are displayed in Table 3.2.

The third group of data we make use of is *Countries* dataset from the network

Table 3.2. Animals and their attributes. A "1" corresponds to the belongingness of the attribute to the corresponding animal.

	Leg ≥ 4	**Carnivore**	**Feather**	**Wings**	**Domes**	**Eaten**	**>100k**	**>2m**	**BrthUnderWtr**	**Extinct**	**Danger**	**Life>20**	**Beak**	**WalkOn2Lgs**	**Spd≥20**
Giraffe	1	0	0	0	0	0	1	1	0	0	0	1	0	0	1
Cow	1	0	0	0	1	1	1	1	0	0	0	0	0	0	1
Lion	1	1	0	0	0	0	1	0	0	0	1	0	0	0	1
Gorilla	1	0	0	0	0	0	1	0	0	0	1	1	0	1	1
Fly	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Spider	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Shark	0	1	0	0	0	0	1	0	1	0	1	1	0	0	1
Horse	1	0	0	0	1	1	1	1	0	0	0	0	0	0	1
Elephant	1	0	0	0	0	0	1	1	0	0	0	1	0	0	1
Mammoth	1	0	0	0	0	0	1	1	0	1	0	1	0	0	1
Sabre Tiger	1	1	0	0	0	0	1	0	0	1	1	0	0	0	1
Pig	1	0	0	0	1	1	1	0	0	0	0	1	0	0	0
Cod	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0
Eel	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1
Jellyfish	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Dolphin	0	1	0	0	0	0	1	1	1	0	0	1	0	0	1
Nemo	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Shrimp	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
Dog	1	1	0	0	1	0	0	0	0	0	0	0	0	0	1
Cat	1	1	0	0	1	0	0	0	0	0	0	1	0	0	1
Fox	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
Wolf	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1
Rabbit	1	0	0	0	1	1	0	0	0	0	0	0	0	0	1
Chicken	0	0	1	1	1	1	0	0	0	0	0	0	1	1	0
Eagle	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1
Seagull	0	1	1	1	0	0	0	0	0	0	0	0	1	1	1
Blackbird	0	1	1	1	0	0	0	0	0	0	0	0	1	1	1
Bat	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0
T. Rex.	1	1	0	0	0	0	1	1	0	1	1	1	0	1	1
Neanderthal	1	1	0	0	0	0	0	0	0	1	0	1	0	1	0
Triceratops	1	1	0	0	0	0	1	1	0	1	1	1	0	0	0
Man	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1
Penguin	0	1	1	1	0	0	0	0	0	0	0	0	1	1	1

analysis literature [55]. This dataset declares five relational attributes of trade and diplomacy between 24 countries in 1984. Relations are binary, that's all data entries are either zero or one. Black squares in the data correspond to the existence of links (or edges), whereas an actual non-edge is represented with a white square. The countries used for creating the dataset are Algeria, Argentina, Brazil, China, Czech Republic, Ecuador, Egypt, Ethiopia, Finland, Honduras, Indonesia, Israel, Japan, Liberia, Madagascar, New Zealand, Pakistan, Spain, Switzerland, Syria, Thailand, United Kingdom (UK), United States of America (USA) and Yugoslavia. The relations are trade in food and live animals, crude materials, minerals and fuels, basic manufactured goods, and the exchange of diplomats. Each data matrix is of size 24×24 where rows correspond to the exporters and columns to the importers. Five different networks in Countries dataset are displayed in Figure 3.1. As we roughly know about the locations, living standards, etc., of the countries in the world, we can subjectively evaluate how good the DPMM Biclustering algorithm we have introduced is on grouping the importer and exporter countries. In addition to this subjective evaluation, we calculate AUROC score of the link prediction procedure we perform so that we can evaluate the results in an objective manner.

As we mentioned before, discovering biological partitions are the most frequent usage area of biclustering algorithms in the literature. Therefore, lastly we test the two nonparametric biclustering algorithms on a biological data from a microarray study of lung cancer. Samples were gathered from 56 lung cancer patients exhibiting four different histological types. Samples 1-20 are pulmonary carcinoid samples, 21-33 are colon cancer metastasis samples, 34-50 are normal lung samples, and 51-56 are small cell carcinoma samples. The dataset is described in detail in [56] and was used in different works [54,57–59]. Researchers are interested in identifying groups of co-regulated genes for different histological types of lung cancer. We use a random subset of 100 genes from the original data of 12625 genes. Therefore, the resulting data matrix is of size 100×56. It is displayed in Figure 3.2.

Monte Carlo based algorithms do not aim to provide us with an exact partitioning result but instead they intend to give us an idea about real marginal likelihood of the

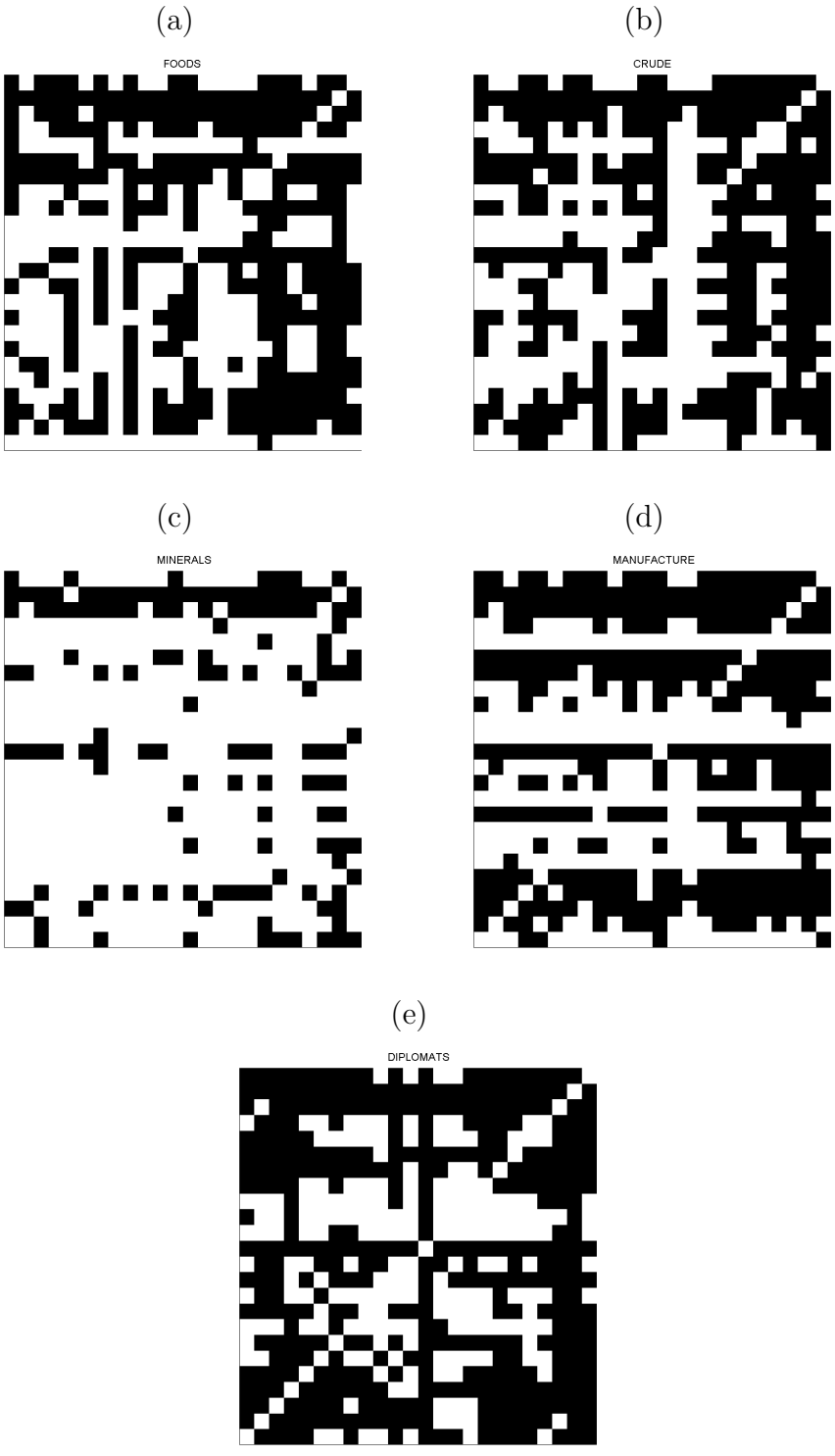


Figure 3.1. Trade and diplomacy relations between 24 countries in 1984. **(a)** Foods **(b)** Crude **(c)** Minerals **(d)** Manufacture **(e)**Diplomats. Rows and columns correspond to exporters and importers respectively. A black square indicates existence of the corresponding relationship.

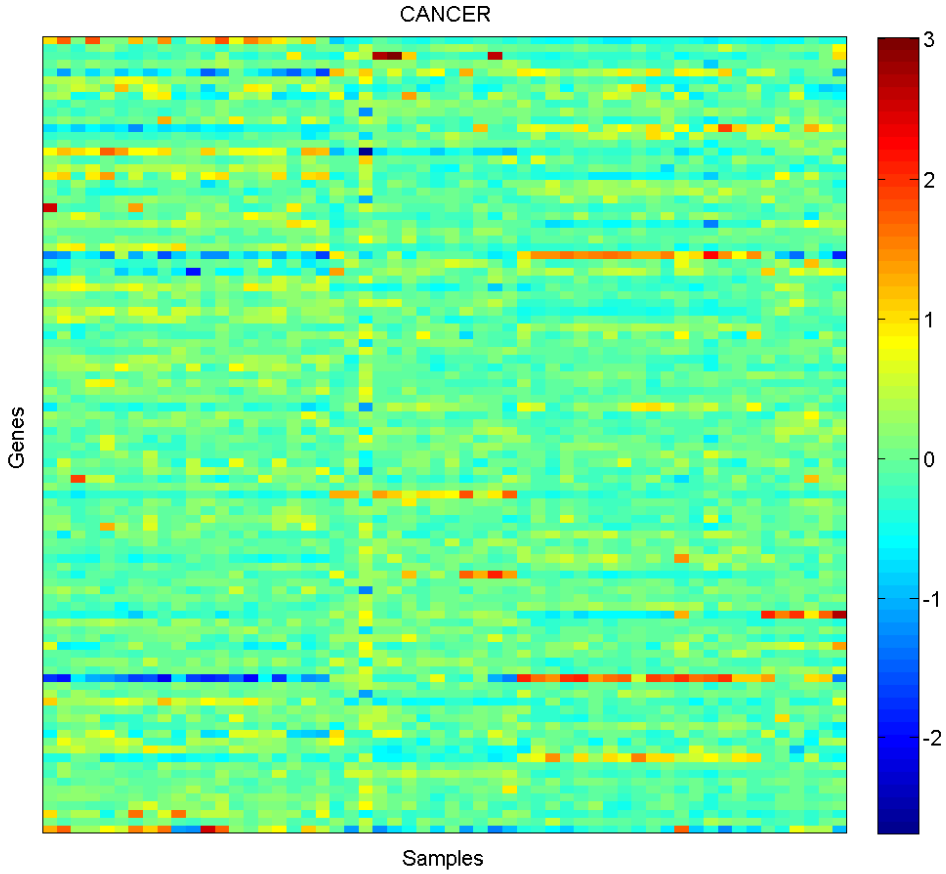


Figure 3.2. Lung cancer data matrix. Rows correspond to genes and columns correspond to patients exhibiting different histological types. Patients 1-20 are of type pulmonary carcinoid, patients 21-33 are of type colon cancer metastasis, patients 34-50 are normal, and patients 51-56 are of type small cell carcinoma.

data roaming around it after the burn-in period. That's, we do not find a single partitioning, but instead a distribution over partitionings. Therefore, for the purpose of imputation, we take care of all the likelihoods roamed after the burn-in period and average them to get the imputed cluster values.

At each Monte Carlo iteration, we perform one CGS scan for rows and columns each, and then apply five random split-merge proposals for both as introduced in [60]. Split-merge operations provide the necessary jumps that are needed when CGS scans stuck in local minima. In this way, it is probable to explore MCMC state space and approach to the global minima.

We test the performance of the algorithm by means of link prediction as mentioned in Section 2.3. For this purpose, we first apply DPMM Biclustering procedure on complete data. Then we randomly remove 50% of the data entries, and estimate missing values based on the results of the partitioning we have just performed. This approach uses the fact that the data values in the same cluster would be very close to each other. Since ground truth knowledge for biclustering is rare, it provides an objective metric for evaluating the biclustering performance. It is such that we use the original data as the ground truth.

While there are many metrics to evaluate the link prediction performance, a standard metric for quantifying the accuracy of prediction algorithms, commonly used in the medical sciences and machine learning communities, is the AUROC (a.k.a AUC) statistic, which is equivalent to the area under the ROC curve [61]. It measures how good the algorithm is at predicting positive (actually existing) links in the dataset. We use AUROC score in our experiments to evaluate the performance of algorithms for binary relational data. For evaluating imputation results for Gaussian data, we use NRMSE score which is a frequently used measure of the accuracy of the values predicted by an estimator based on the difference of the estimated values and the values actually observed.

As another evaluation method for DPMM clustering, we carry out the biclustering algorithms on the sparse data and test how well we can partition the data with missing items. For each network, we have removed 10%, 30%, 50%, 70%, 90% and 99% of the data entries uniformly at random in such a way that 10% of the original data entries are removed first, and are kept removed when we remove 30% next, etc. We respectively check how good DPMM Biclustering algorithm is at imputing the missing values for Gaussian data and predicting the missing actual edges for Bernoulli Countries data. Different than the biclustering experiments This type of evaluation is more challenging than the previous one because biclustering is also done based on the sparse data.

Since five relation matrices in the trade and diplomacy relations dataset contain links for the same group of countries, as a third performance metric for evaluating the

DPMM Biclustering, we remove all edges in one of the five data matrices, and use the other four to predict the partitioning structure the missing data exhibits. That's, we do not have any information about the food import/export relations for example, then we use minerals, manufacture, and crude materials trade to extract structure for food trading between the countries. We again use cumulative edge removing strategy for the compound network data we use as the source for biclustering.

At each iteration of DPMM Biclustering, we perform one CGS scan for row and column indicators, and then apply five split-merge proposals for both.

DPMM experimental setup parameters we use for the datasets are displayed in Tables 3.3, 3.4, 3.5 and 3.6.

Table 3.3. DPMM Biclustering parameters for Gaussian toy dataset.

Parameter	**Value**
Likelihood distribution	Gaussian
Prior distribution	Gaussian
DP prior α	0.01, 0.1, 1, 10, 50
${\sigma_{data}}^2$	1
${\sigma_{clusters}}^2$	10
$\lambda_{clusters}$	0
Run count	10
MCMC Epoch	1000
MCMC Burnin	100

3.3.2. Models and Inference

First type of model we use is 1D Gaussian for the toy data, and the second one the Beta-Bernoulli model we use for Countries trade and diplomacy relations dataset. In this section, we give details for the derivations of CGS update equations for these models.

Table 3.4. DPMM Biclustering parameters for Animals dataset.

Parameter	**Value**
Likelihood distribution	Bernoulli
Prior distribution	Beta
DP prior α	0.01, 0.1, 1, 10, 50
$\alpha_{clusters}$	0.2
$\beta_{clusters}$	0.2
MCMC Epoch	1000
MCMC Burnin	100

Table 3.5. DPMM Biclustering parameters for Countries dataset.

Parameter	**Value**
Likelihood distribution	Bernoulli
Prior distribution	Beta
DP prior α	0.01, 0.1, 1, 10, 50
$\alpha_{clusters}$	0.3
$\beta_{clusters}$	0.3
Run count	10
MCMC Epoch	1000
MCMC Burnin	100

Table 3.6. DPMM Bicluster parameters for Lung Cancer dataset.

Parameter	**Value**
Likelihood distribution	Gaussian
Prior distribution	Gaussian
DP prior α	0.01, 0.1, 1, 10, 50
${\sigma_{data}}^2$	0.1
${\sigma_{clusters}}^2$	1
$\lambda_{clusters}$	0
MCMC Epoch	100
MCMC Burnin	10

3.3.2.1. Gaussian-Gaussian Model. We model the generated toy dataset by 1D Gaussian distribution and we select the parameter prior again as 1D Gaussian model, as Gaussians are conjugate to each other. We set priors only on the mean of the data distribution. That's, we assume priorly that the mean of the biclusters is distributed by Gaussian. The Gaussian data likelihood in exponential family representation is:

$$p(x_{pq}|\theta) = \mathcal{N}(x_{pq};\theta,1) = \exp(\theta x_{pq} - \frac{\theta^2}{2} - (\frac{x_{pq}^2}{2} + \log(\sqrt{2\pi}))) \tag{3.11}$$

First, let's calculate the first integral which is to be used for the probability of the data point to be in one of the existing clusters. Adapting the equation of the definition of $p(x_{pq}|\mu)$ to the exponential family definition in the previous section:

$$\begin{aligned} f(\theta) &= \theta \\ t(x_{pq}) &= x_{pq} \\ a_l(\theta) &= \tfrac{\theta^2}{2} \\ h_l(x_{pq}) &= \tfrac{x_{pq}^2}{2} + \log\sqrt{2\pi} \end{aligned} \tag{3.12}$$

Using the statistics in Equation 3.12, we get the prior as:

$$p(\theta \mid \lambda_1, \lambda_2) = \exp\left(\lambda_1\theta + \lambda_2(\frac{-\theta^2}{2}) - a_c(\lambda_1,\lambda_2)\right) \tag{3.13}$$

We can see here that the sufficient statistics of above equation are θ and θ^2, which also shows that the prior is a Gaussian.

Now we should find λ_1 and λ_2 in terms of the mean and variance of a univariate Gaussian.

$$\begin{aligned} \mathcal{N}(\theta;\mu_0,\sigma_0^2) &= \frac{1}{\sqrt{2\pi\sigma_0^2}}\exp\left(\frac{-1}{2\sigma_0^2}(\theta^2+\mu_0^2-2\theta\mu_0)\right) \\ &= \exp\left(\frac{\theta\mu_0}{\sigma_0^2} - \frac{\theta^2}{2\sigma_0^2} - \left(\frac{\mu_0^2}{2\sigma_0^2} + \log(\sqrt{2\pi\sigma_0^2})\right)\right) \end{aligned} \tag{3.14}$$

Then

$$\begin{aligned} \lambda_1 &= \frac{\mu_0}{\sigma_0^2} \\ \lambda_2 &= \frac{1}{\sigma_0^2} \end{aligned} \tag{3.15}$$

and

$$a_c(\lambda_1, \lambda_2) = \frac{\mu_0^2}{2\sigma_0^2} + \log\sqrt{2\pi\sigma_0^2} = \frac{\lambda_1^2}{2\lambda_2} + \log\sqrt{\frac{2\pi}{\lambda_2}} \tag{3.16}$$

The hyperparameters $\hat{\lambda}_1$ and $\hat{\lambda}_2$ of Gaussian posterior $p(\theta_k \mid \boldsymbol{x}_{k,-p}, \lambda_1, \lambda_2)$ are then:

$$\begin{aligned} \hat{\lambda}_1 &= \frac{\mu_0}{\sigma_0^2} + \textstyle\sum_{k,-p}\sum_l x_{pq} \\ \hat{\lambda}_2 &= \frac{1}{\sigma_0^2} + N_{k,-p} \end{aligned} \tag{3.17}$$

Then the variance of $p(\theta_k \mid \boldsymbol{x}_{k,-p}, \lambda_1, \lambda_2)$ is:

$$\sigma^2 = \frac{1}{\hat{\lambda}_2} = \frac{1}{\frac{1}{\sigma_0^2} + N_{k,-p}} \tag{3.18}$$

Then the mean of $p(\theta_k \mid \boldsymbol{x}_{k,-p}, \lambda_1, \lambda_2)$ is:

$$\mu = \frac{\hat{\lambda}_1}{\hat{\lambda}_2} = \frac{\frac{\mu_0}{\sigma_0^2} + \sum_{k,-p}\sum_l x_{pq}}{\frac{1}{\sigma_0^2} + N_{k,-p}} \tag{3.19}$$

In order to find the predictive likelihood of x_{pq}, we should substitute values in Equation 3.17 into Equation 3.16. Then;

$$\begin{aligned} p(x_{pq} \mid \boldsymbol{x}_{k,-p}) &= \exp\left(\frac{(\hat{\lambda}_1 + x_{pq})^2}{2(\hat{\lambda}_2 + 1)} + \log\sqrt{\frac{2\pi}{\hat{\lambda}_2 + 1}} - \frac{\hat{\lambda}_1^2}{2\hat{\lambda}_2} - \log\sqrt{\frac{2\pi}{\hat{\lambda}_2}} - \frac{x_{pq}^2}{2} - \log\sqrt{2\pi}\right) \\ &= \sqrt{\frac{\hat{\lambda}_2}{2\pi(\hat{\lambda}_2 + 1)}}\exp\left(\frac{1}{2}\left(\frac{(\hat{\lambda}_1 + x_{pq})^2}{\hat{\lambda}_2 + 1} - \frac{\hat{\lambda}_1^2}{\hat{\lambda}_2} - x_{pq}^2\right)\right) \end{aligned} \tag{3.20}$$

Now, let's calculate the marginal probability of x_{pq}, namely the easier integral, which is again can be calculated using marginalization formula attained in Section 2.6. Then;

$$\begin{aligned} p(x_{pq}) &= \exp\left(\frac{(\lambda_1 + x_{pq})^2}{2(\lambda_2+1)} + \log\sqrt{\frac{2\pi}{\lambda_2+1}} - \frac{\lambda_1^2}{2\lambda_2} - \log\sqrt{\frac{2\pi}{\lambda_2}} - \frac{x_{pq}^2}{2} - \log\sqrt{2\pi}\right) \\ &= \sqrt{\frac{\lambda_2}{2\pi(\lambda_2+1)}}\exp\left(\frac{1}{2}\left(\frac{(\lambda_1 + x_{pq})^2}{\lambda_2+1} - \frac{{\lambda_1}^2}{\lambda_2} - x_{pq}^2\right)\right) \end{aligned} \tag{3.21}$$

λ_1, λ_2, $\hat{\lambda}_1$, $\hat{\lambda}_2$ in terms of mean and variance of the Gaussian prior of the model parameter are given in Equations 3.15 and 3.17. While implementation, they should be substituted into Equations 3.20 and 3.21.

3.3.2.2. Beta-Bernoulli Model. We model the Countries dataset by the Bernoulli distribution and we select the parameter prior as the conjugate prior of Bernoulli distribution, namely the Beta distribution. That's, we assume priorly that the success ratio of the Bernoulli, that's the mean of the biclusters, is distributed by Beta. The Bernoulli data likelihood in exponential family representation is:

$$p(x_{pq}|\theta) = \mathcal{BE}(x_{pq};\theta) = \exp(x_{pq}\log\theta + \log(1-\theta) - x_{pq}\log(1-\theta)) \tag{3.22}$$

Then;

$$\begin{aligned} f(\theta) &= \theta \\ t(x_{pq}) &= x_{pq} \\ a_l(\theta) &= -\log(1-\theta) \\ h_l(x_{pq}) &= 0 \end{aligned} \tag{3.23}$$

Using the statistics in Equation 3.23, we get the Beta prior for θ as:

$$\begin{aligned} p(\theta \mid \lambda_1, \lambda_2) &= \exp\left(\lambda_1 \log\tfrac{\theta}{1-\theta} + \lambda_2\log(1-\theta) - a_c(\lambda_1,\lambda_2)\right) \\ &= \exp(\lambda_1\log\theta - (\lambda_1+\lambda_2)\log(1-\theta) - a_c(\lambda_1,\lambda_2)) \end{aligned} \tag{3.24}$$

Since

$$p(\theta \mid a, b) = \mathcal{B}(\theta; a, b) = \exp\left((a-1)\log\theta + (b-1)\log(1-\theta) - \log\frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}\right) \tag{3.25}$$

then; λ_1, λ_2 and $a_c(\lambda_1, \lambda_2)$ in terms of a and b:

$$\begin{aligned} \lambda_1 &= a - 1 \\ \lambda_2 &= a + b - 2 \\ a_c(\lambda_1, \lambda_2) &= \log\frac{\Gamma(\lambda_1+1)\Gamma(\lambda_2-\lambda_1+1)}{\Gamma(\lambda_2+2)} \end{aligned} \tag{3.26}$$

Then the hyperparameters of Beta posterior are:

$$\begin{aligned} \hat{\lambda}_1 &= a - 1 + \textstyle\sum_{k,-p}\sum_l x_{pq} \\ \hat{\lambda}_2 &= a + b - 2 + N_{k,-p} \end{aligned} \tag{3.27}$$

Now, we should find $p(x_{pq})$ and $p(x_{pq} \mid \boldsymbol{x}_{k,-p})$ using Equation 3.21 and Equation 3.20 respectively.

$$p(x_{pq}) = \frac{\Gamma(\lambda_1 + x_{pq} + 1)\Gamma(\lambda_2 + 1 - \lambda_1 - x_{pq} + 1)}{\Gamma(\lambda_2+3)} \frac{\Gamma(\lambda_2+2)}{\Gamma(\lambda_1+1)\Gamma(\lambda_2-\lambda_1+1)} \tag{3.28}$$

Because the data are distributed by Bernoulli, each data entry is either zero or one. If we handle those cases separately in order to simplify above equation (using the fact that $\Gamma(n) = (n-1)!$ for discrete distributions):

$$p(x_{pq}) = \begin{cases} \dfrac{\lambda_2 - \lambda_1 + 1}{\lambda_2 + 2} = \dfrac{b}{a+b}, & x_{pq} = 0 \\ \dfrac{\lambda_1 + 1}{\lambda_2 + 2} = \dfrac{a}{a+b}, & x_{pq} = 1 \end{cases} \tag{3.29}$$

Similarly;

$$p(x_{pq} \mid \boldsymbol{x}_{k,-p}) = \begin{cases} \dfrac{\hat{\lambda}_2 - \hat{\lambda}_1 + 1}{\hat{\lambda}_2 + 2}, & x_{pq} = 0 \\ \dfrac{\hat{\lambda}_1 + 1}{\hat{\lambda}_2 + 2}, & x_{pq} = 1 \end{cases} \tag{3.30}$$

3.3.3. Results

In this subsection, we explain and demonstrate the results of experiments we performed to check the performance of DPMM Biclustering.

3.3.3.1. Biclustering Result for Gaussian Toy Data. In this experiment, we apply DPMM Biclustering on the original data with no missing entries. Then we remove 50% of the entries and use the biclustering results for original data to predict the missing links in the sparse data. We check the biclustering performance via the Normalized Root Mean Square Error (NMRSE) metric. Figure 3.3 shows the resulting biclusters in matrix representation and Figure 3.4 demonstrates the NRMSE scores of DPMM Biclustering for different values of α.

(a) (b)

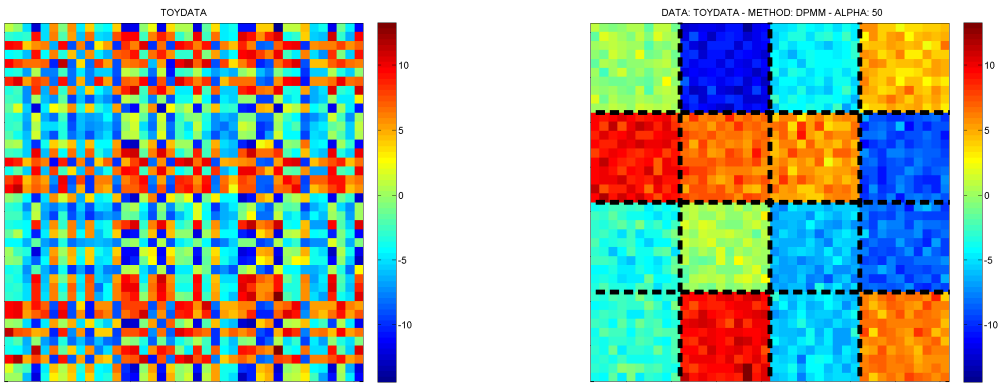


Figure 3.3. Generated toy data and DPMM Biclustering result. **(a)** Original Gaussian data. **(b)** Data rearranged according to biclustering results for maximum marginal likelihood estimation with DP prior $\alpha = 1$. The biclusters are nicely separated and they exhibit the same structure as the one we set while generating the data.

3.3.3.2. Imputation Result for Gaussian Toy Data. In this experiment, we remove 10%, 30%, 50%, 70%, 90% and 99% of the data entries respectively. At each of those six steps, the entries of the data missing at the previous step remain missing. Then we test how good we predict the values of the missing links. Different than the experiment in Section 3.3.3.1, we apply the biclustering algorithm on the sparse data here. Figure 3.5 represents the NRMSE scores of missing data imputation.

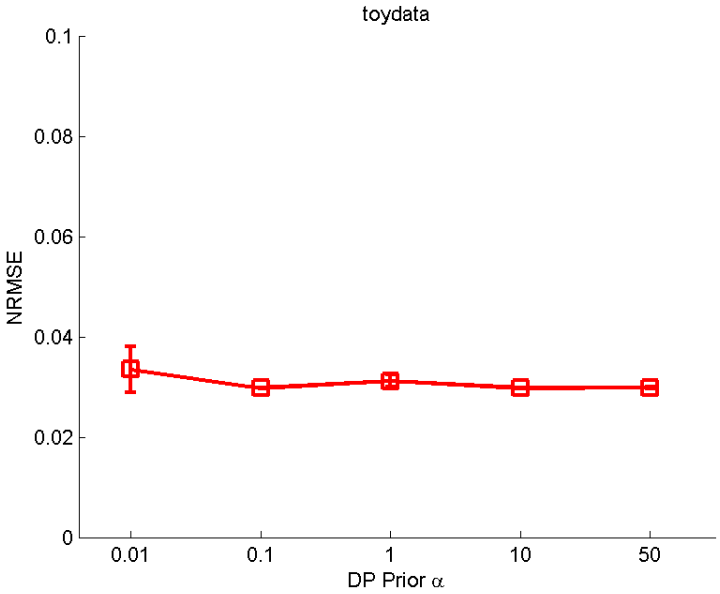


Figure 3.4. DPMM Biclustering performances for toy data with respect to different Dirichlet priors (α). We get quite small NRMSE for all tested values of α. That's, the same well-separated structure is achieved for all α with infinitesimal standard deviations.

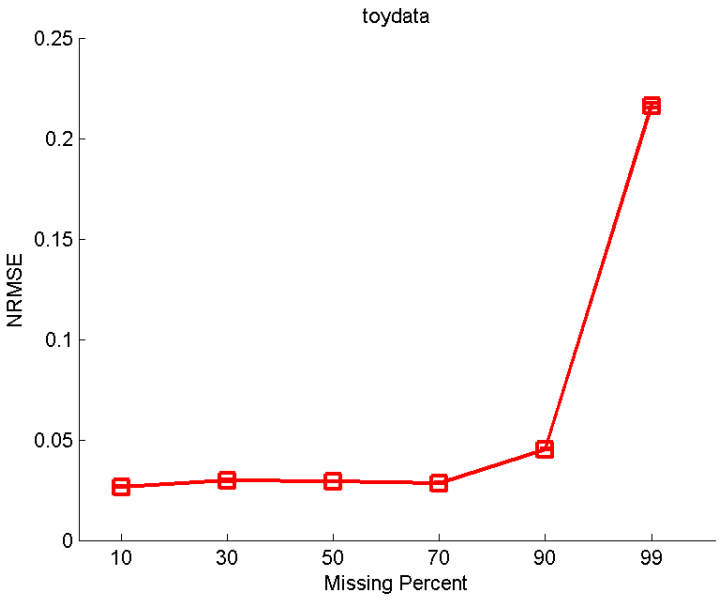


Figure 3.5. Imputation performances for toy data with respect to different missing ratios. We get the same small NRMSE for all missing percents except 99. That's, the same well-separated structure is achieved until almost all of the observed data items are removed.

3.3.3.3. Biclustering Result for Animals Data. In this experiment, we apply DPMM Biclustering algorithm on the animal-attribute data. Figure 3.6 shows the resulting animal-attribute biclusters.

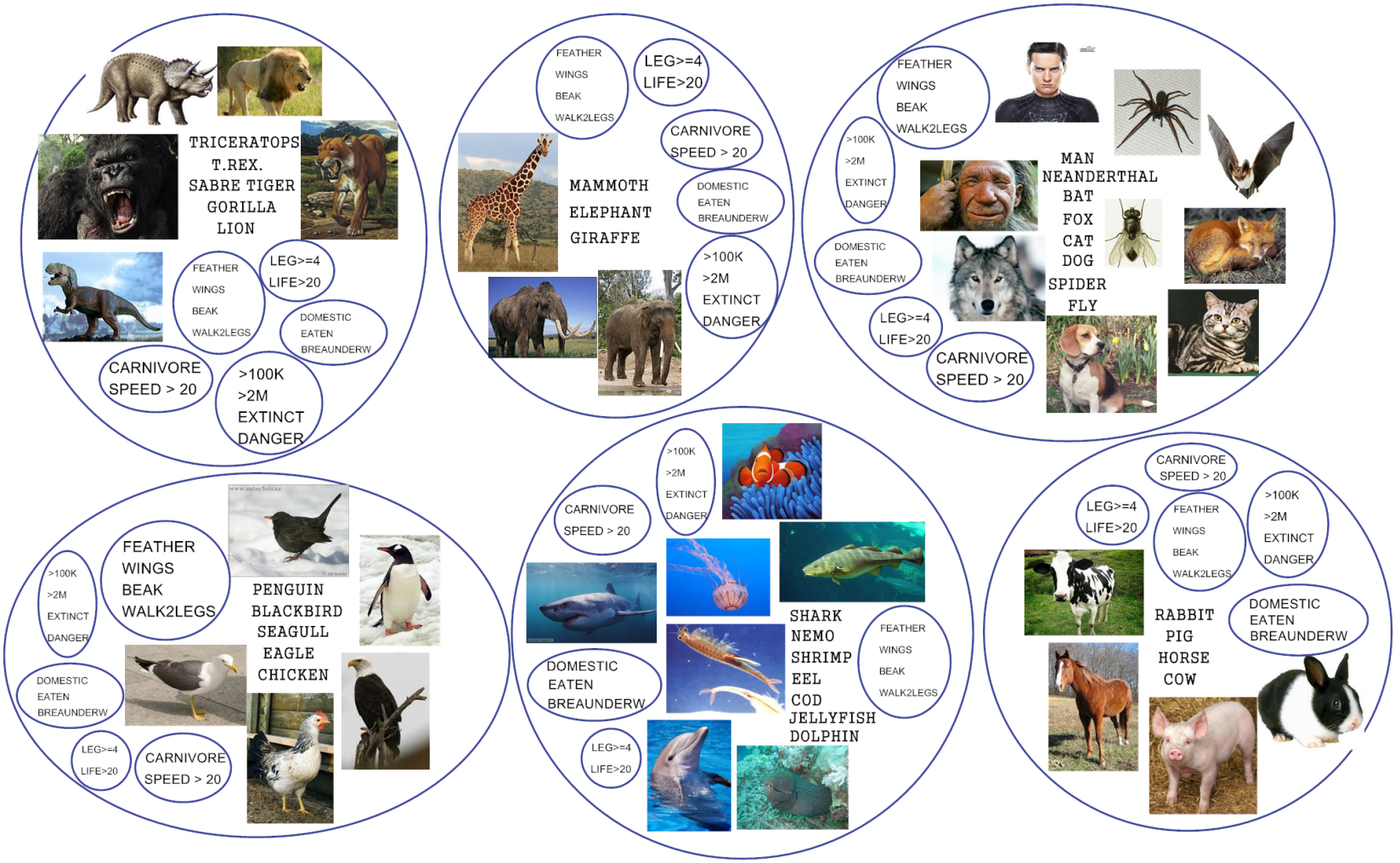


Figure 3.6. DPMM Biclustering result for Animals data. Six animal clusters and five attribute clusters exist. Font size of the attribute clusters (inner circles) demonstrates how much the attributes belong to the animals in the animal cluster (outer circle)

3.3.3.4. Biclustering Results for Countries Data. To evaluate the success of DPMM Biclustering on trade and diplomacy relations data, we apply biclustering algorithm on original data with no missing entries, them remove 50% of the entries and try to predict them using the partitioning results. We calculate AUROC scores for missing edge prediction. True positive outcome for each missing ratio is the proportion of the number of estimated actual edges (1s) to the number of missing actual edges (1s). On the other hand, false positive outcome is calculated as the ratio of estimated actual edges (1s) to the number of missing actual non-edges (0s). In Figure 3.7, DPMM Biclustering results for the trade of Foods, Crude Materials, Minerals, Manufactured Materials, and exchange of diplomats are demonstrated respectively in matrix representation. Original data are given in Figure 3.1. Moreover, in Figure 3.8, the biclustering performances in terms of AUROC are represented for different values of α.

In Figure 3.9, biclustering results for food trade are shown in a visually nicer representation. We see USA, Central Europian countries and Japan as great traders. They are leaders in both food import and export. Since they are first world countries, this is in fact an expected result. On the other hand, African countries have almost no food import or export tasks. Since they are poor and generally in famine, this also shows the success of the biclustering task. Far East countries go together with South American countries as food exporters with intermediate trade intensity.

In Figure 3.10, biclustering results for crude material trade are shown. We see USA, Central Europian countries and Japan as great Crude traders similar to the food trade. They are leaders in both crude material import and export. On the other hand, third world African countries have almost no crude material import or export tasks. They are already in a deep poverty. The second world Far East countries such as Thailand or New Zealand are also important crude material exporters and they export those material to first world countries where they are manufactured. They seem to import crude materials less than they export. Likewise the food trade, South American countries are within the same intermediate-intensity export cluster as the Far East. However, when import is in question, those third-world countries are with African Central African countries. North African countries, a.k.a the Middle East

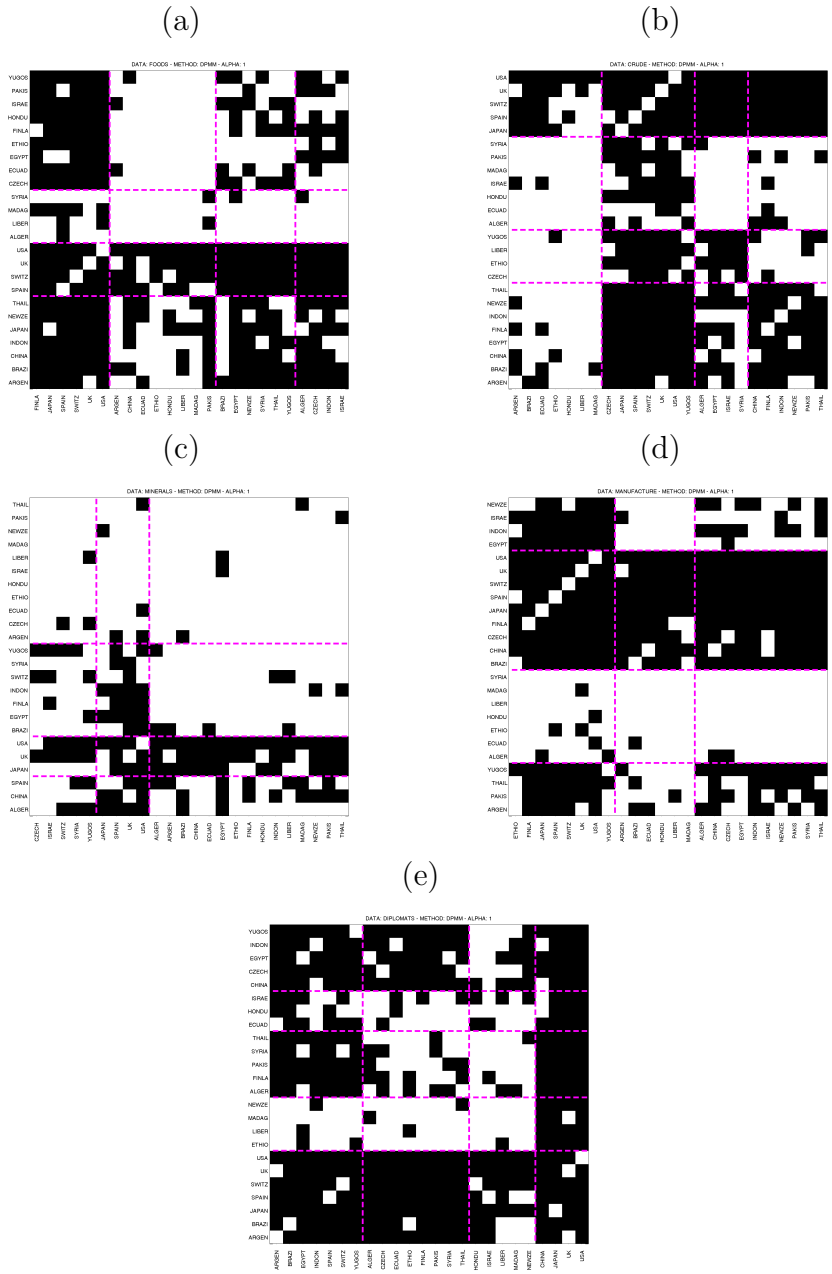


Figure 3.7. DPMM Biclustering results for Countries trade and diplomacy relations dataset. **(a)** Foods **(b)** Crude **(c)** Minerals **(d)** Manufacture **(e)** Diplomats. Dashed lines separate the partitions. Data are rearranged according to indicators bringing about the maximum marginal data likelihood with DP prior $\alpha = 1$. Original dataset is shown in Figure 3.1.

countries Algeria, Egypt, Israel and Syria separate themselves from Central African countries which are in a deeper suffering and create their own import clusters. Far East countries also seem to be crude material importers with intermediate trade density.

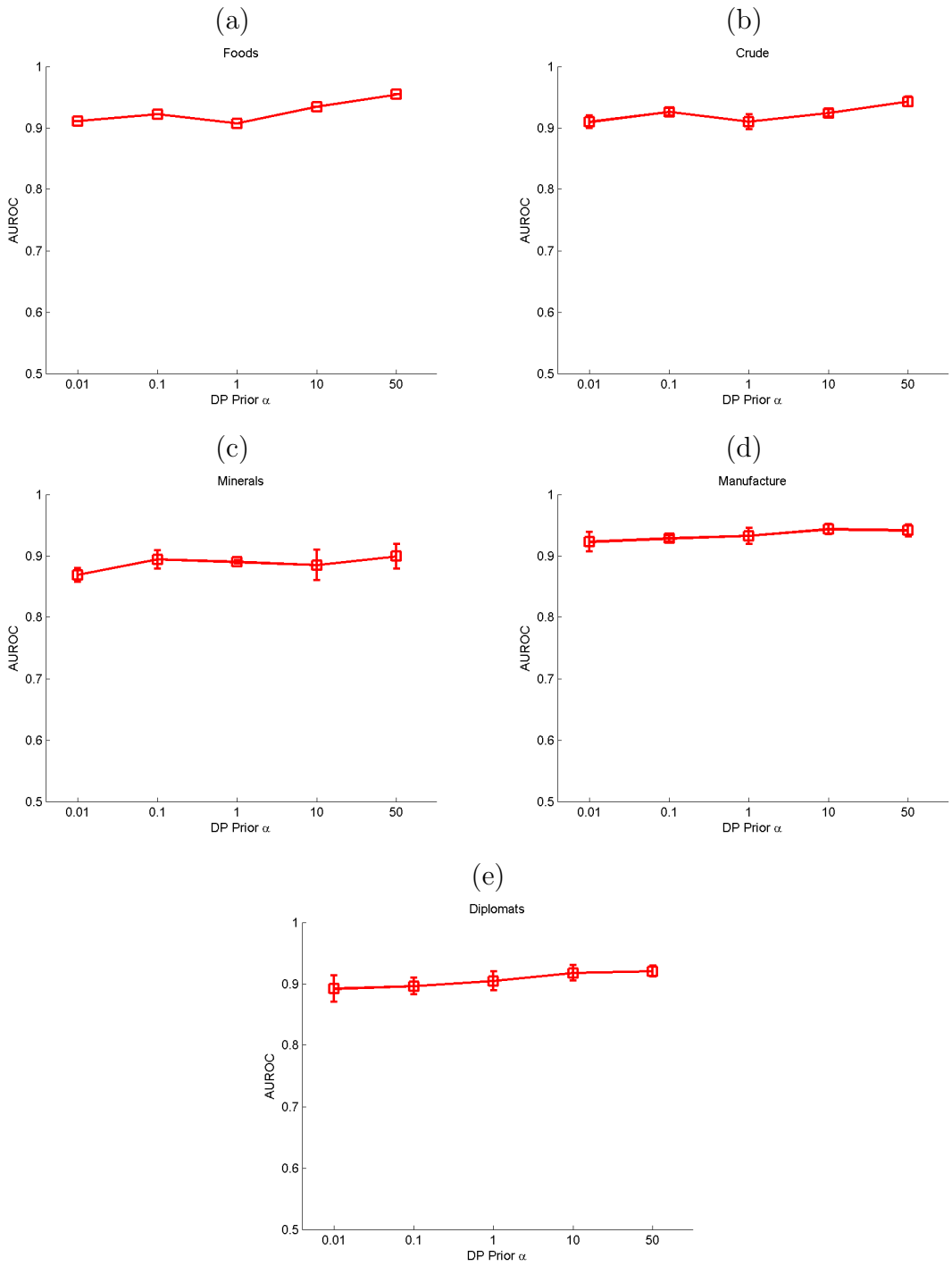


Figure 3.8. DPMM Biclustering performances for the trade and diplomacy dataset with respect to different Dirichlet priors (α). **(a)** Foods **(b)** Crude **(c)** Minerals **(d)** Manufacture **(e)** Diplomats. We get AUROC scores of around 0.9 for each of the tested values of α.

In Figure 3.11, biclustering results for mineral trade are demonstrated. Although mineral trade density is lower than trade density of other types of materials in general,

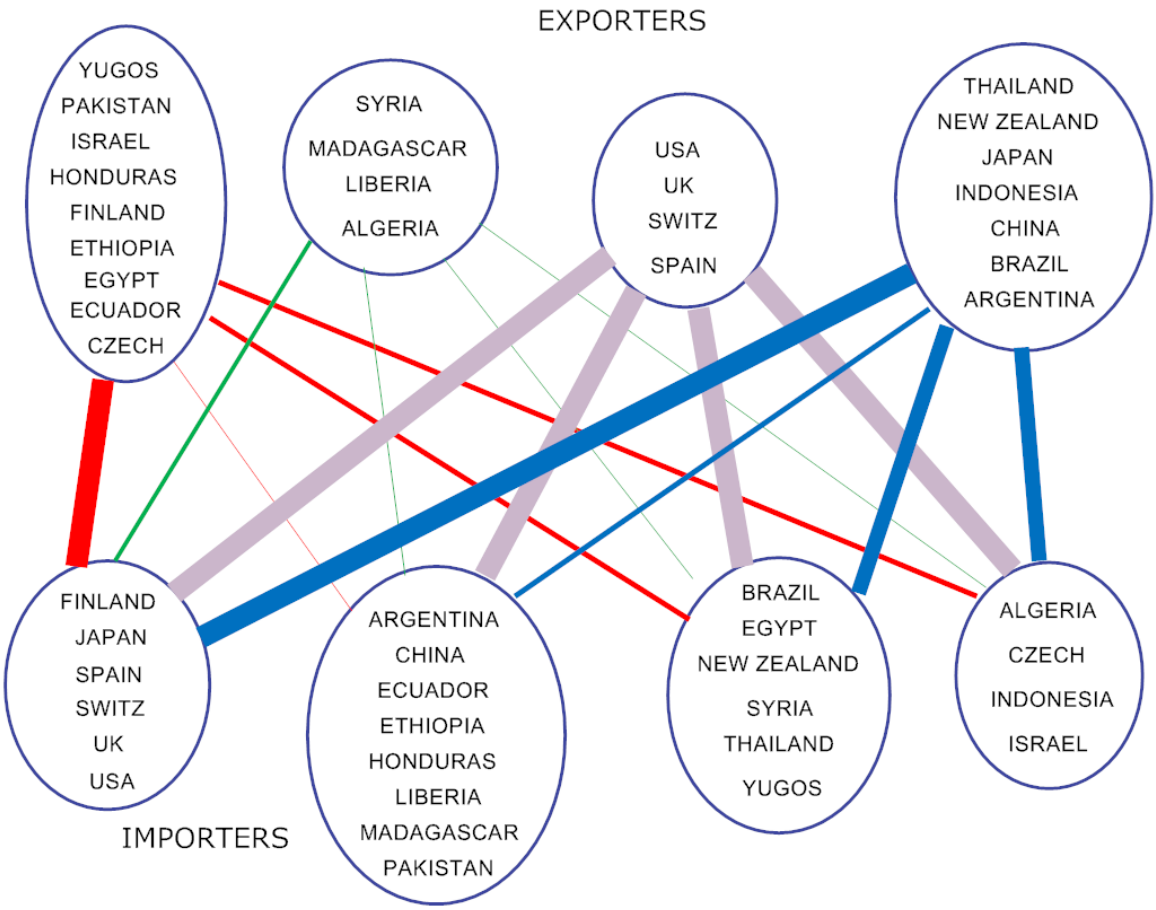


Figure 3.9. DPMM Biclustering results are represented as the relationship graph for Foods data. Thick lines between two groups of countries correspond to intense relationships between them whereas thin lines represent a lack or spareness of relation.

we still see first world countries as great mineral exporters. On the other hand, Far East or African countries have almost no mineral export tasks except China and Algeria. Those two behave much like European countries when mineral trade is in question. However, for the import case, they go together in the same cluster with almost all Far East and African countries. We see Easter European countries together with Israel and Syria as the most important cluster of mineral importers.

In Figure 3.12, biclustering results for manufactured material trade are represented. In addition to central European countries and USA, other European countries, Japan, China and Brazil go together with leading manufactured material exporter countries. Since we see them as biggest crude material importer group in Figure 3.10,

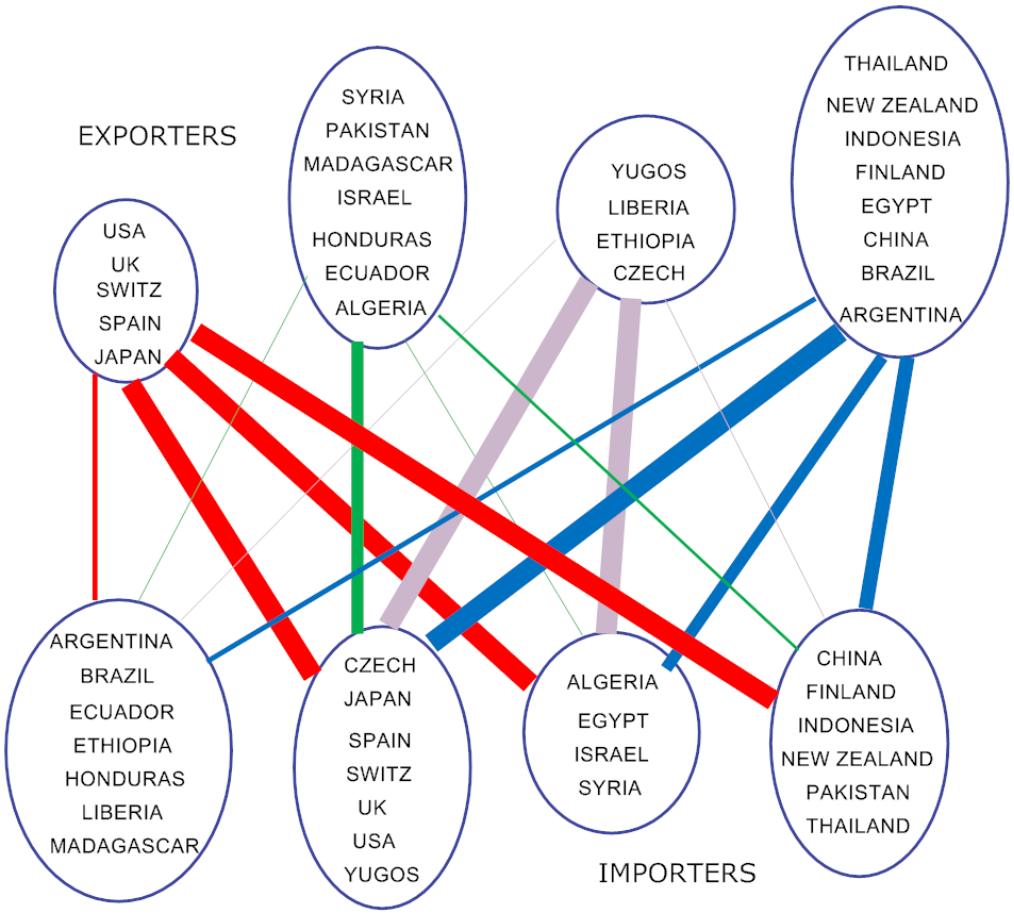


Figure 3.10. DPMM Biclustering results are represented as the relationship graph for Crude data. Thick lines between two groups of countries correspond to intense relationships between them whereas thin lines represent a lack or spareness of relation.

this seems reasonable. African countries except Israel and Egypt have almost no manufactured material export or import. North African, that's the Middle East countries behave similar to the Far East countries for both export and import tasks. East and Far East countries comprise importers. This is reasonable since we see them as crude material exporters in Figure 3.10.

In Figure 3.13, biclustering results for manufactured material trade are shown. Diplomacy relations are generally more intense and more symmetric than trade relations. Almost all of the countries exchange diplomats between one another. However, USA, UK and Japan are still the leaders in both export and import of diplomats. When import is in question, China goes together with those three. On the other hand,

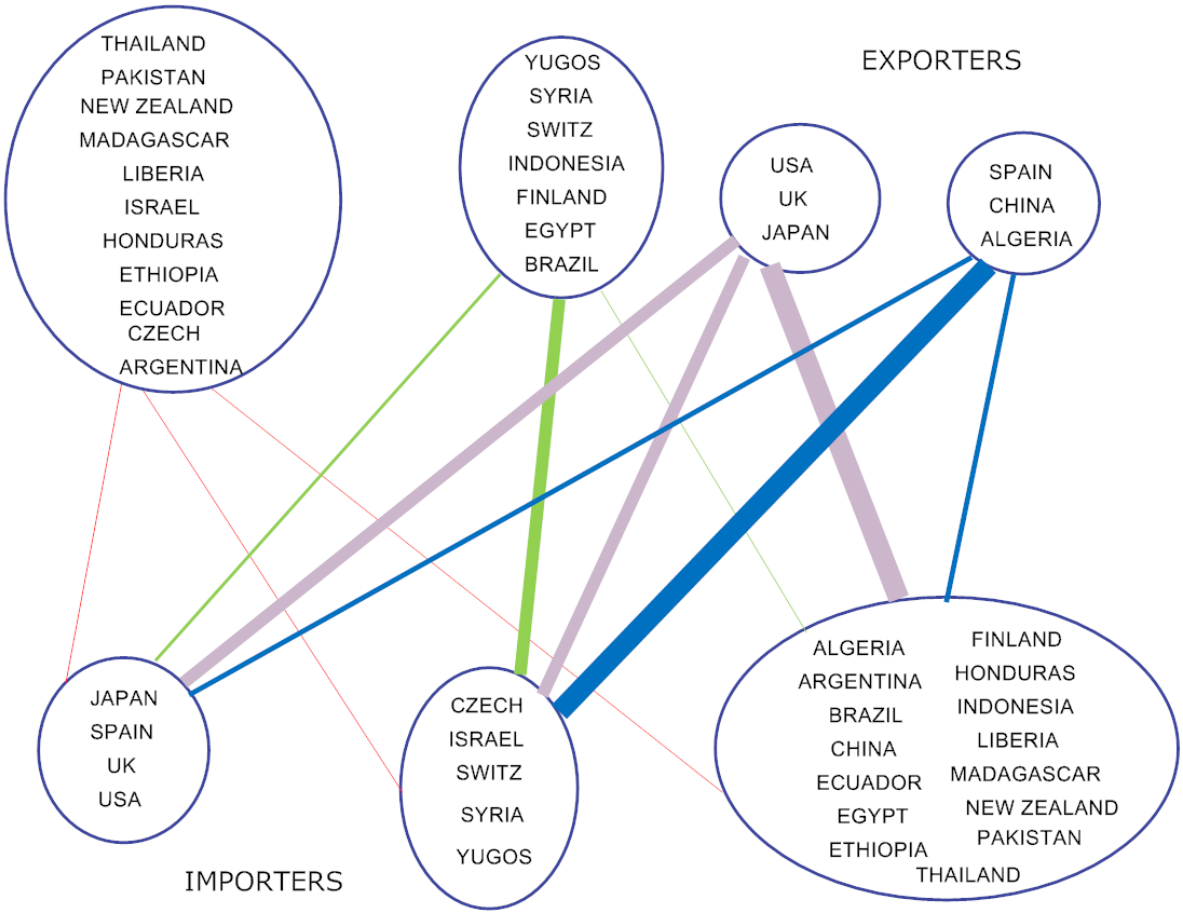


Figure 3.11. DPMM Biclustering results are represented as the relationship graph for Minerals data. Thick lines between two groups of countries correspond to intense relationships between them whereas thin lines represent a lack or spareness of relation.

China's diplomat export task is more similar to the ones of Easter European and Middle East. The group they compromise is a significant diplomacy exporter group. Another important issue we can read from the graph is that Asian countries Thailand, New Zealand, Indonesia or Pakistan are not in a the same group together as it was the case for other materials. Indonesia seems to be a country having denser diplomacy relations than other three, whereas New Zealand is the one in the last place among those four Asian countries in both diplomacy export and import tasks. Being the furthest from the rest of the world, it seems to be a reasonable result for diplomatic relations. It is not surprising that African countries are in the last place among all countries in exchange of diplomats.

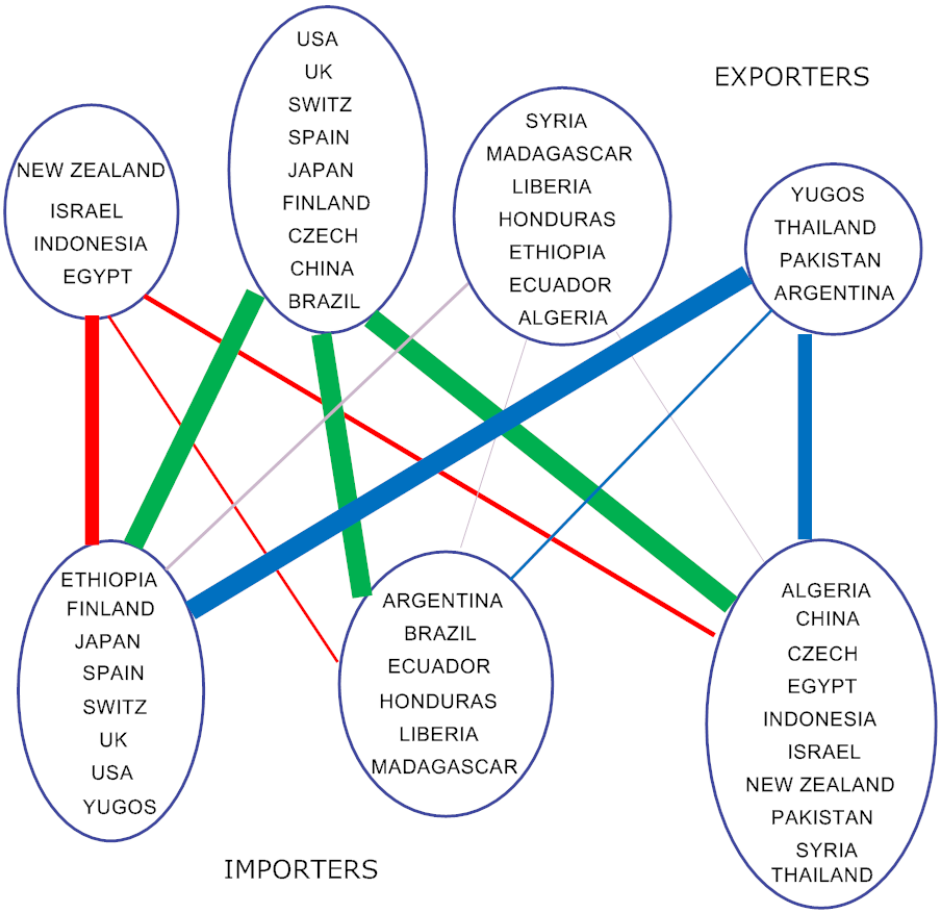


Figure 3.12. DPMM Biclustering results are represented as the relationship graph for Manufacture data. Thick lines between two groups of countries correspond to intense relationships between them whereas thin lines represent a lack or spareness of relation.

3.3.3.5. "Imputation by Self" Results for Countries Data. In this experiment, we apply biclustering on the five networks with missing percents 10, 30, 50, 70, 90 and 99 respectively, and then impute the missing entries using the result of the partitioning. The plots in Figure 3.14 demonstrate imputation results of this experiment in terms of AUROC scores for five different trade and diplomacy networks and for different missing data percents. The results of "Imputation by Complement" experiment in Section 3.3.3.6 are also shown in the same figure with separate plots for each network for the purpose of comparing two imputation experiments.

3.3.3.6. "Imputation by Complement" Results for Countries Data. In this second imputation experiment on the Countries trade and diplomacy dataset, we impute one of

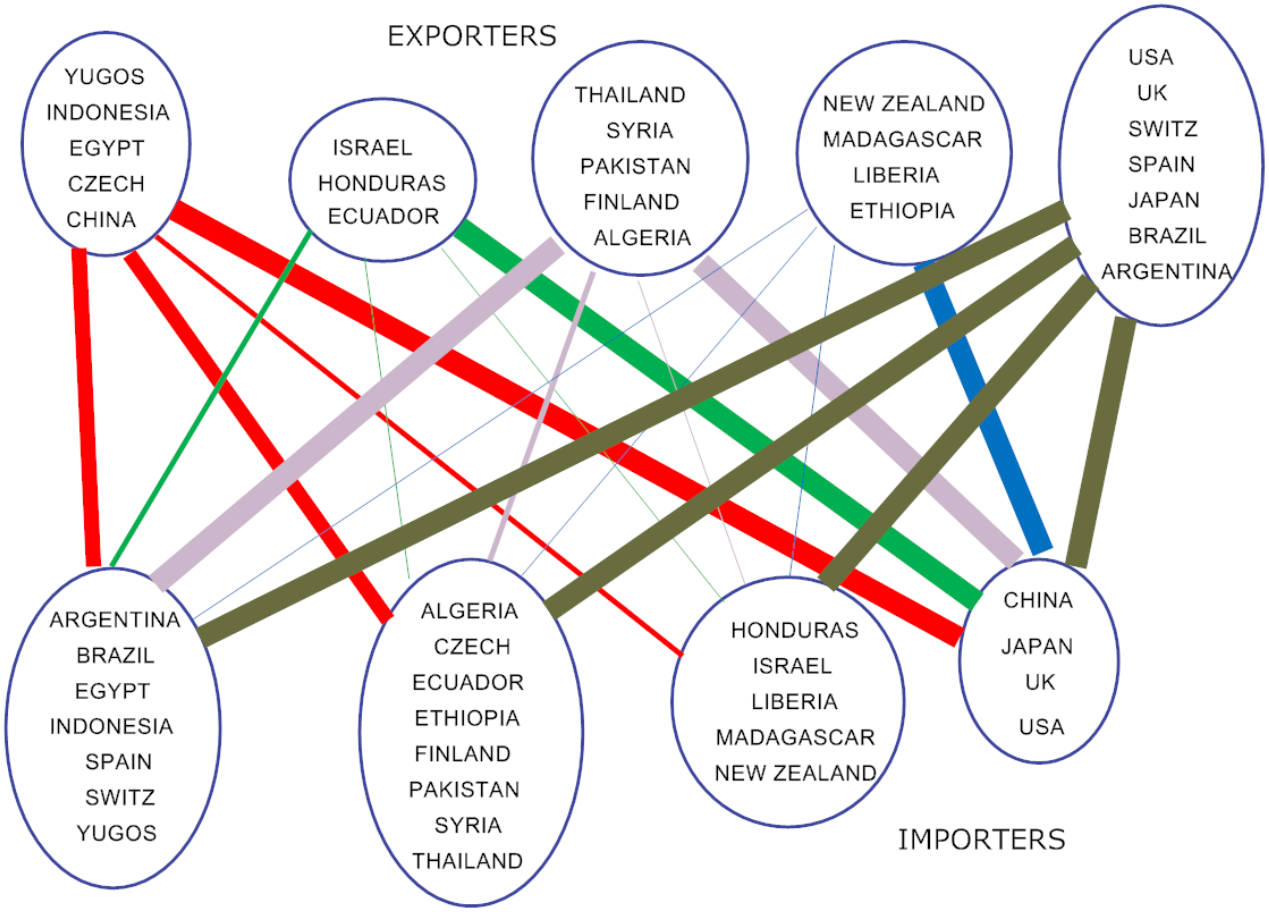


Figure 3.13. DPMM Biclustering results are represented as the relationship graph for Diplomats data. Thick lines between two groups of countries correspond to intense relationships between them whereas thin lines represent a lack or spareness of relation.

the five different networks using the results of DPMM Biclustering applied on the sparse four networks complement to the network to be imputed. For instance, we impute Foods data using the biclustering results for Crude, Minerals, Manufacture and Diplomats data. The complement data we use for biclustering is sparse with missing percents 10, 30, 50, 70, 90 and 99 respectively. The plots in Figure 3.14 demonstrate imputation results of this experiment in terms of AUROC scores for five different trade and diplomacy networks and for different missing data percents. The results of "Imputation by Self" experiment in Section 3.3.3.5 are also shown in the same figures with separate plots for each network for the purpose of comparing two imputation experiments. Whereas generally lower than the ones retrieved using self data, imputation AUROC scores retrieved using complement data are still fairly high around 0.7-0.8

level for low missing ratios. Those high values of sparse complement data strongly support the accuracy of our biclustering results stating that USA, central European countries and Japan are the trade leaders for all five datasets whereas African countries are generally in the last place and Far East or East European countries are in between.

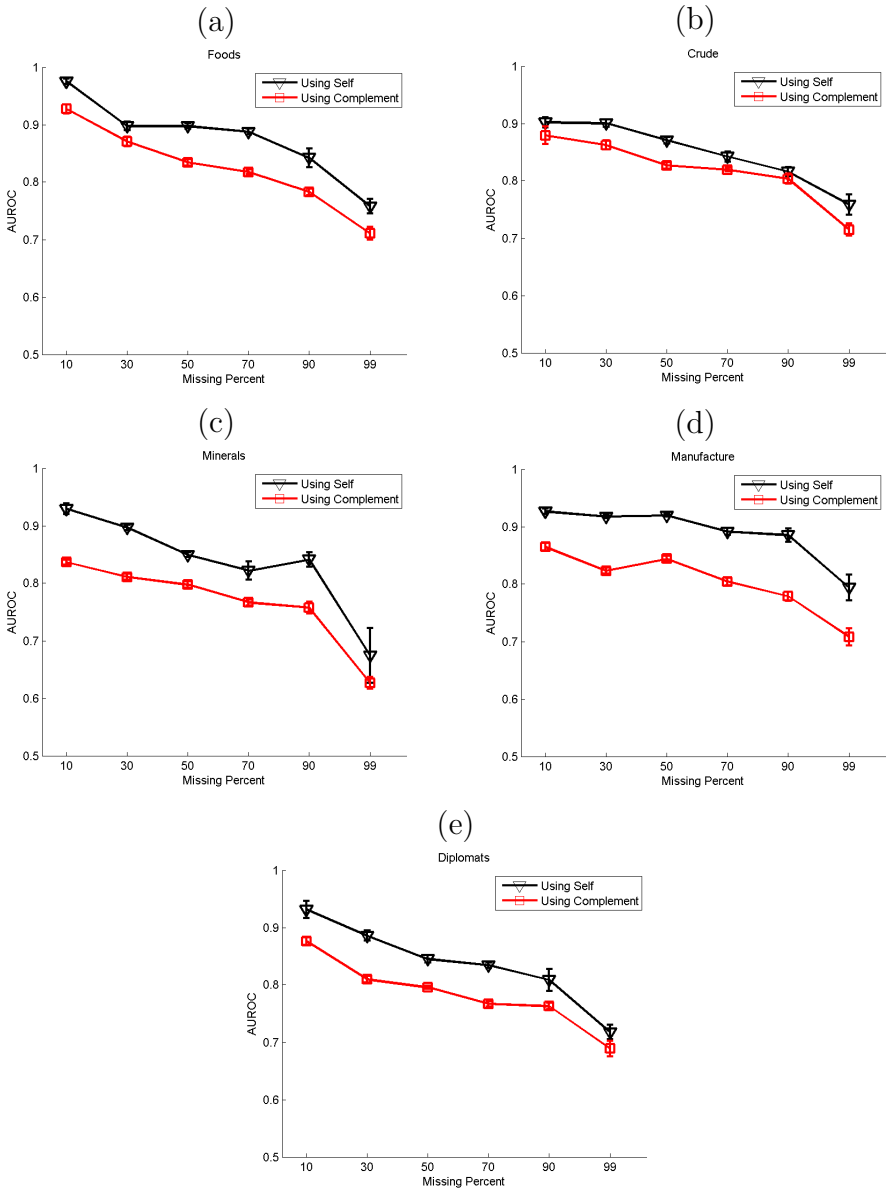


Figure 3.14. Missing value imputation results for Countries dataset for experiments "Imputation by Self" and "Imputation by Complement". **(a)** Foods **(b)** Crude **(c)** Minerals **(d)** Manufacture **(e)** Diplomats. Inference is done on sparse complement data based on all likelihoods after the burnin period of CGS.

3.3.3.7. Biclustering Result for Lung Cancer Data. In this experiment, we apply DPMM Biclustering on lung cancer data displayed in Figure 3.2. Then we remove 50% of the

entries and impute them based on the results of the partitioning operation. Figure 3.15 shows the biclustering result visually and compares the sample clusters with the ground truth. Furthermore, Figure 3.16 demonstrates the NRMSE scores of biclustering results for different values of α.

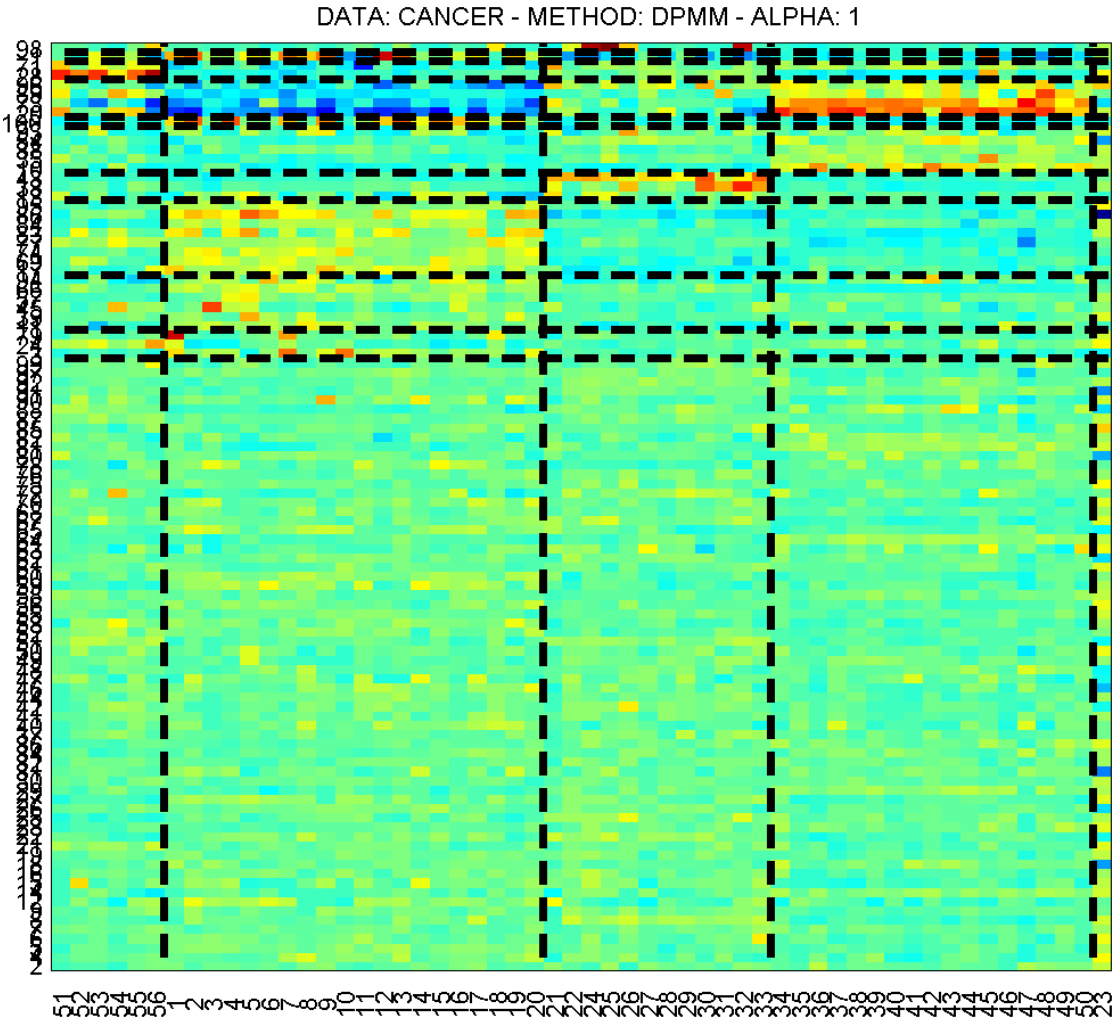


Figure 3.15. DPMM Biclustering results for Lung Cancer dataset. Only one sample (Sample 23) does not comply with the ground truth for samples. It is located in a separate sample cluster.

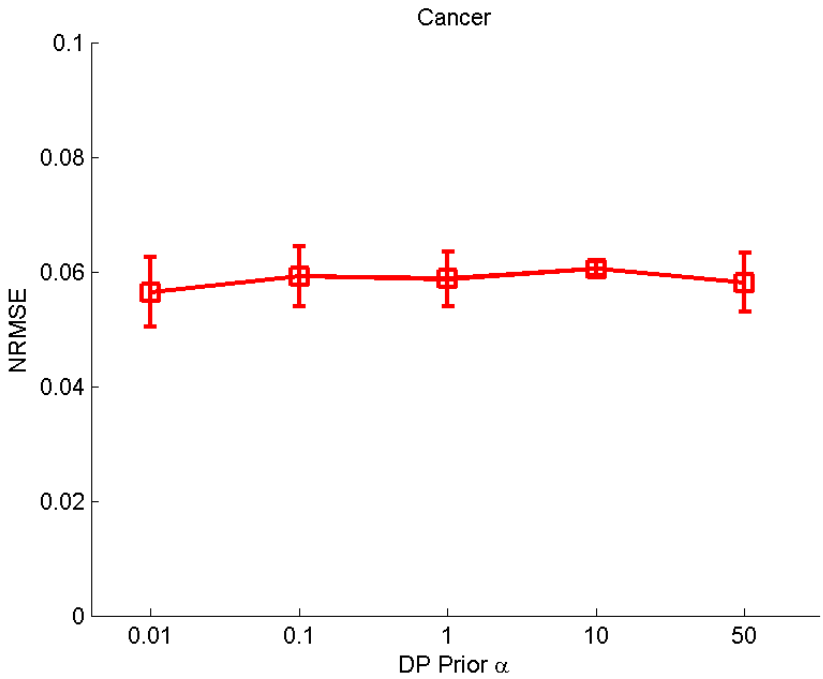


Figure 3.16. DPMM Biclustering performances for Lung Cancer dataset.

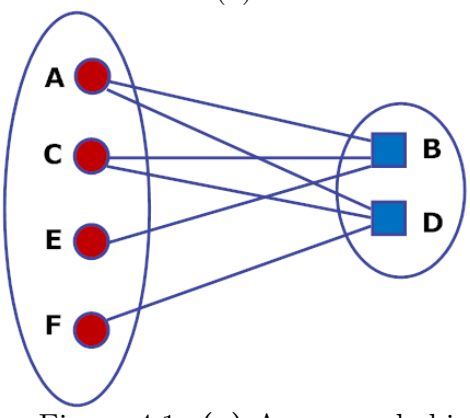
4. SPECTRAL BICLUSTERING

In Chapter 3, we mentioned how to retrieve the biclusters in the data by making use of DPMM and CGS directly on the rows and columns of the original data matrix. However, when the data matrix is of high dimensions, DPMM inference for both rows and columns is an inefficient and time consuming process. In this chapter, we deal with how to project the two dimensional data matrix onto the spectral space and then do non-parametric inference on resulting one-way data to retrieve the row and column clusters. That's, we integrate spectral matrix operations with nonparametric biclustering procedures. We also demonstrate the results of the experiments we performed to test the Spectral Biclustering method.

4.1. Spectral Graph Partitioning

A bipartite graph is a graph whose nodes constitute two independent groups such that each each edge is from one of the nodes in one of those groups to a node in the other group. That's, no edges exist between the nodes of the same group. Figure 4.1 shows an example unweighted bipartite graph and its adjacency matrix. However, the spectral partitioning graph mentioned next can be applied to partition both weighted and unweighted graphs.

(a)



(b)

	A	C	E	F	B	D
A	0	0	0	0	1	1
C	0	0	0	0	1	1
E	0	0	0	0	1	0
F	0	0	0	0	0	1
B	1	1	1	0	0	0
D	1	1	0	1	0	0

Figure 4.1. **(a)** An example bipartite graph. The two ellipses correspond to two bipartites. **(b)** The adjacency matrix of that bipartite graph.

Let A be a $n \times n$ square symmetric adjacency matrix of a bipartite graph G such that

$$A_{ij} = A_{ji}, \quad \forall i = 1, .., n, \quad \forall j = 1, .., n \tag{4.1}$$

where n corresponds to the number of nodes in the graph. Then the Laplacian matrix $\mathcal{L}$ of the graph G is an $n \times n$ symmetric matrix such that

$$L_{ij} = \begin{cases} \sum_k A_{ik}, & \text{if } i = j \\ -A_{ij}, & \text{if } i \neq j \text{ and there is an edge between } i \text{ and } j \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

Let's define a diagonal weight matrix W such that

$$W_{ij} = \begin{cases} \sum_k A_{ik}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

Alternatively, if W is shown as $\begin{pmatrix} D_r & 0 \\ 0 & D_c \end{pmatrix}$, then;

$$L = W - A = \begin{pmatrix} D_r & -B \\ -B^T & D_c \end{pmatrix} \tag{4.4}$$

where B and B^T correspond to upper-right and lower-left blocks of the symmetric adjacency matrix. This follows from the fact that an adjacency matrix is symmetric.

Every eigenvalue of a Laplacian matrix is non-negative. That is, it is positive semi-definite. Laplacian matrix is used in many graph algorithms based on spectral graph theory.

Where f and g are the clusterings for rows and columns respectively, our aim is to retrieve f and g providing the minimum between-cluster sum of weights and maximum within-cluster sum of weights. In [2], the authors suggest and prove that where $q = \begin{pmatrix} f \\ g \end{pmatrix}$, the second eigenvector of the generalized eigenvalue problem

$$Lq = \lambda W q \tag{4.5}$$

provides a real relaxation to this optimization problem. Substituting blocked forms of $\mathcal{L}$, W and q in Equation 4.5 results in

$$\begin{pmatrix} 0 & \tilde{B} \\ \tilde{B}^T & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \lambda \begin{pmatrix} u \\ v \end{pmatrix} \tag{4.6}$$

where

$$\tilde{B} = D_r^{-1/2} B D_c^{-1/2} \tag{4.7}$$

Then, u and v correspond to left and right singular vectors of $\tilde{B}$ and

$$\begin{pmatrix} u \\ v \end{pmatrix} = Dq = \begin{pmatrix} D_r^{-1/2} f \\ D_c^{-1/2} g \end{pmatrix} = z \tag{4.8}$$

. Since the eigenvectors of a symmetric matrix form an orthonormal base, z is an orthogonal matrix. After this point, clustering columns $(2, .., 2+k)$ of z would give the desired partitioning of the graph. Choice of k determines the eigenvector count to be used, and it is application-dependent. In [62], the authors assert the more eigenvectors result in better clusterings. However, in high dimensional data, performance matters put important restrictions on the value of k. Therefore, it is an issue of optimization to choose k.

Generally in the literature, K-means algorithm is used as the method to partition the columns of z. That's, the parameter, namely the number of the clusters is given

as the function argument. If this parameter is 2, then the procedure is called *bipartitioning* and if it is bigger than 2, then the task is called *multipartitioning*. Let vector c correspond to the resulting desired partitioning with entries corresponding to the node orders in the rows and columns of the adjacency matrix. Then, initial $\|u\|$ elements of c correspond to the row clusters and the remaining $\|v\|$ correspond to the column clusters respectively. In a graph partitioning problem, we have all nodes of the graph in both rows and columns of graph adjacency matrix A. So, we expect the clusterings of both rows and columns, namely u and to be the same. For instance, the desired bipartitioning indicator vector c corresponding to the graph in Figure 4.1 would be $(1\ 2\ 1\ 2\ 1\ 1\ 1\ 2\ 1\ 2\ 1\ 1)^T$.

In [2], the authors apply the procedure explained in the previous section to the word-document clustering. However, the procedure needs a symmetric matrix as the adjacency matrix whereas the word-document matrix is not. Therefore, the structure is a bit different here from the graph adjacency case. Let M be a $w \times d$ the matrix where the entry M_{ij} stands for the existence of word i in document j. Then they create the structure in Figure 4.2 in the background to apply the mentioned partitioning algorithm. So, the blocks of Equation 4.4 correspond to the original word-document matrix M and what calculated in Equation 4.7 correspond to $\tilde{M}$.

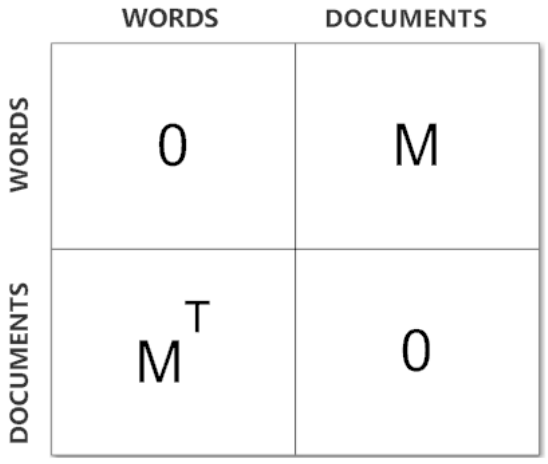


Figure 4.2. Word-document relation matrix is rearranged to be made symmetric to apply spectral biclustering procedure.

This approach has been considered as an important milestone in the spectral

biclustering subject and most of the existing algorithms are based on this perspective. However it has an important drawback. As a result of the mentioned procedure, the number of row and column clusters of the original relation matrix would be the same. This is because rows and columns are rearranged and put together in the rows and columns of a bigger symmetric matrix as in Figure 4.2 and the clustering is done using those rearranged rows and columns. Besides, as in the case of the graph adjacency matrix, we expect to get the same clustering for the rearranged rows and columns.

In [3], the authors suggest a solution to this problem. They apply their methods on microarray data of genes and conditions. Similar to the above procedure described in [2], they calculate $\tilde{B}$ in Equation 4.7. However, rather than combining left and right singular vectors and partitioning them together, they consider the left singular vector of Equation 4.7 u as the one to cluster columns, and they use Equation 4.9 to calculate the vector to partition the rows.

$$v = B * u; \tag{4.9}$$

Therefore, they are able to cluster rows and columns independently. Because it is often impractical to partition rows and columns into the same number of clusters, we also follow this approach.

4.2. Nonparametric Modeling and Inference in the Spectral Space

After the two-way base matrix is projected onto the spectral space, we come up with a k-dimensional spectral space with k eigenvectors describing this space. In most of our experiments, data points are projected onto the subspace of two eigenvectors. Now, what we have to do is to cluster the row and column eigenvector subspaces. As mentioned before, generally, K-means clustering is used for this purpose in the literature [2, 3]. Since we are dealing with nonparametric methods in this thesis and K-means needs the count of parameters, namely the number of clusters, it is not a way we can use for clustering the eigenvector pairs to find the corresponding row and column clusters.

Here, our approach is to create a statistical model for the row and column eigenvector pairs and partition them using DPMM for one-way data. To model this space in a nonparametric manner described in Section 2.6, we use 2D Gaussian likelihood with conjugate 2D Gaussian prior. Then we perform CGS iterations for inference purposes on this model. Although spectral partitioning is a frequently utilized method, there are very few examples in the literature that model the spectral space in a nonparametric manner [63].

4.3. Experiments

In order to evaluate the performance of the spectral biclustering algorithm discussed, we do experiments on the simulated and real datasets introduced in Section 3.3.1. Except for the animal-attribute dataset where we use three eigenvectors, we project the data onto the subspace of two eigenvectors by the dimensionality reduction task.

4.3.1. Materials and Methods

We test the spectral biclustering algorithm on the same four groups of data we used in DPMM Biclustering so that comparison of the results would be possible. Properties of datasets we use are explained in Section 3.3.1, and the biclustering parameters for the spectral space are displayed in Table 4.1.

Similar to the approach for DPMM Biclustering in Chapter 3, we randomly remove 50% of the data entries, and estimate missing values based on the results of the partitioning we have just performed and calculate NRMSE scores for Gaussian and AUROC scores for Bernoulli. We consider all the likelihoods roamed during MCMC iterations and average them to get the imputed cluster values.

We give details of DPMM inference for 1D Gaussian model for direct biclustering in Section 3.3.1. Therefore, we do not give the details of the calculations for 2D Gausian model for clustering the eigenvector space here in this chapter. The interested readers

Table 4.1. Spectral Biclustering parameters.

Parameter	**Value**
Spectral space likelihood distribution	2D Gaussian
Prior distribution	2D Gaussian
DP prior α	0.01, 0.1, 1, 10, 50
${\sigma_{data}}^2$	$0.0005\mathcal{I}$
${\sigma_{clusters}}^2$	$0.005\mathcal{I}$
$\lambda_{clusters}$	[0 0]
runCount	20
MCMC Epoch	1000
MCMC Burnin	100

would consider reading Section 3.2. The derivation of equations for DPMM for one-way 2D data would be trivial then.

Similar to the approach for DPMM Biclustering in Chapter 3, at each iteration in the spectral space, we perform one CGS scan for eigenvector pairs for row and columns each, and then apply five random split-merge proposals for both as introduced in [60].

4.3.2. Results

4.3.2.1. Biclustering Result for Gaussian Toy Data. Similar to the one in Section 3.3.3.1, in this experiment, we apply spectral biclustering on the original data with no missing entries, then we remove 50% of the entries and use the biclustering results to predict the missing links. Figure 4.3 demonstrates the biclustering results in matrix representation and Figure 4.4 demonstrates the NRMSE scores of DPMM Biclustering for different values of α.

4.3.2.2. Biclustering Result for Animals Data. In this experiment, we apply spectral biclustering algorithm on the animal-attribute data. Figure 3.6 shows the resulting animal-feature biclusters.

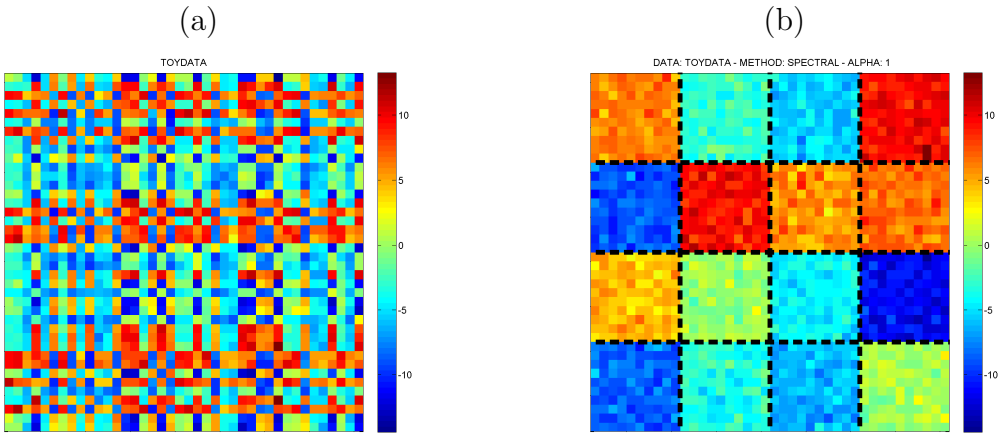


Figure 4.3. **(a)** Original Gaussian data. **(b)** Data rearranged according to biclustering results for maximum likelihood estimation with DP prior $\alpha = 0.1$. The biclusters are nicely separated, and they exhibit the same structure as the one we set while generating the data and as the DPMM Biclustering result in Figure 3.3.

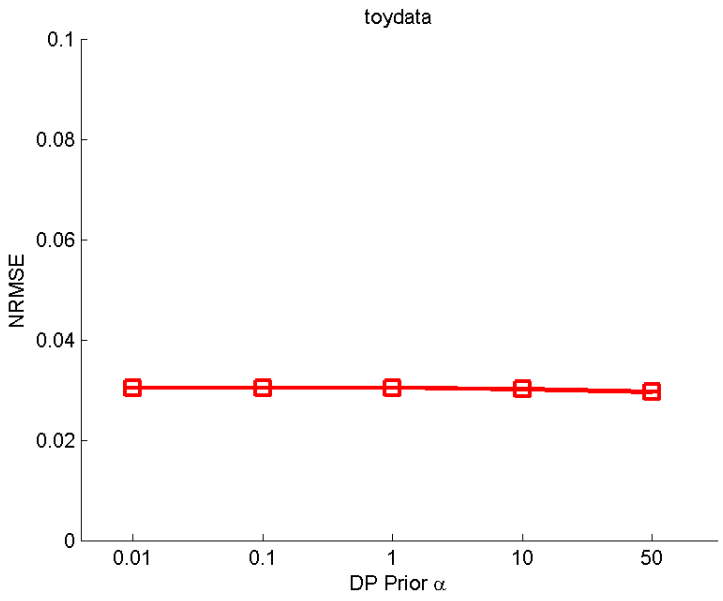


Figure 4.4. Spectral Biclustering performances for toy data with respect to different Dirichlet priors (α). We get the same small NRMSE for all tested values of α. That's, the same well-separated structure is achieved for all α with infinitesimal standard deviations.

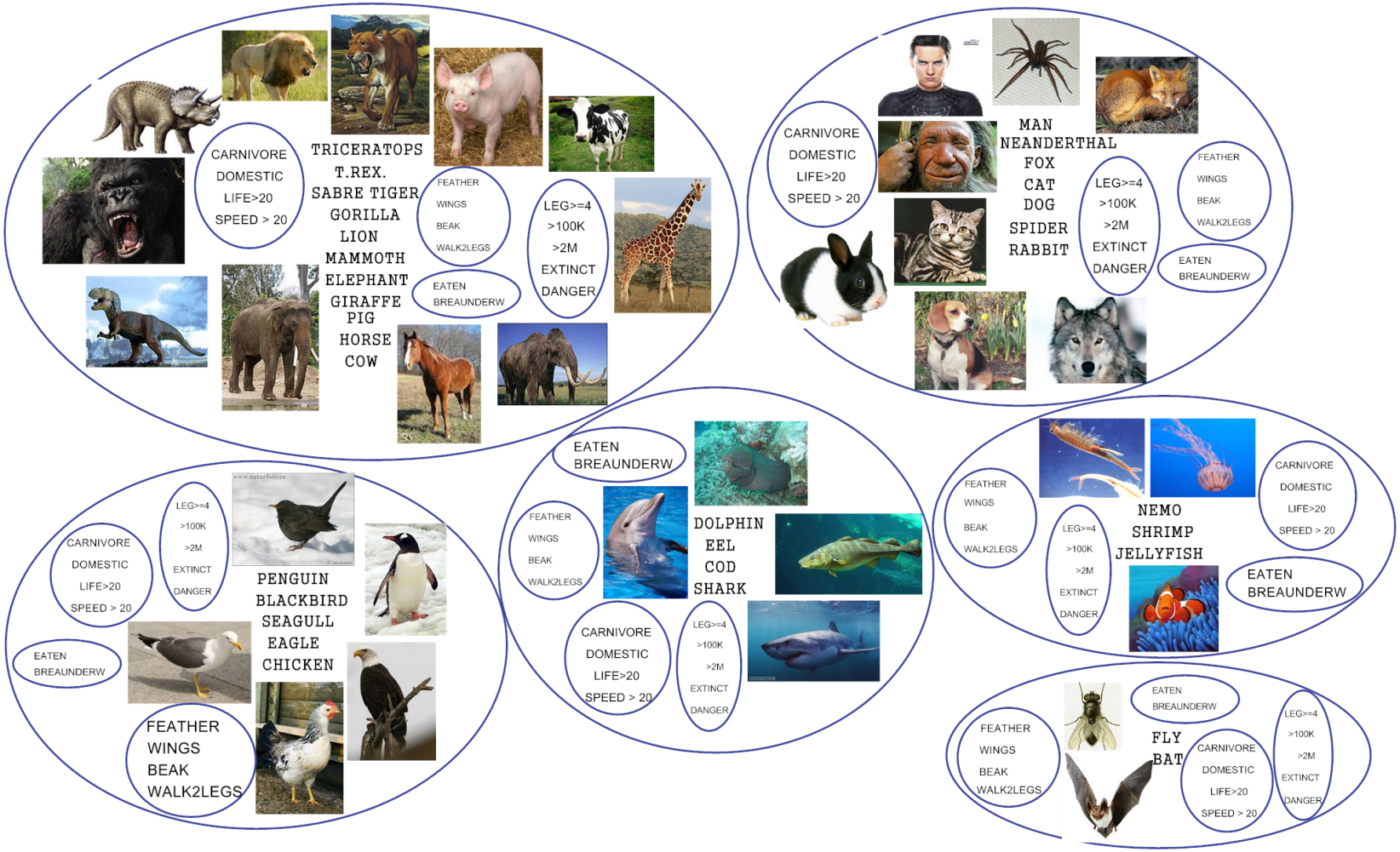


Figure 4.5. Spectral Biclustering result for Animals data. Different than other experiments, three eigenvectors are used in spectral decomposition. There are six animal clusters and five attribute clusters.

4.3.2.3. Biclustering Results for Countries Data. We check how good the spectral biclustering algorithm performs on five trade and diplomacy relations data using the same evaluation method as explained in Section 3.3.3.4. We apply biclustering on the original data, and remove 50% of the entries and fill them using the biclustering result retrieved in the previous step.

In Figure 4.6, spectral biclustering results for the trade of Foods, Crude Materials, Minerals, Manufactured Materials, and exchange of diplomats are demonstrated respectively in matrix representation. Original data are given in Figure 3.1 in Chapter 3. Moreover, in Figure 4.7, the biclustering performances in terms of AUROC are represented for different values of α.

In Figure 4.8, the biclustering results for Foods trade data are shown in a visually nicer representation. This graph is much similar to the Foods DPMM Biclustering result graph in Figure 3.9. Likewise in Figure 3.9, we see USA and Central Europian countries as great food traders. Japan behaves like those leader exporters although it was together with Far East countries in Figure 3.9. Moreover, different than in Figure 3.9, Madagascar is with other African countries for export task, which seems more reasonable. Third world African countries are still in the last place in food trade.

In Figure 4.9, the biclustering results are represented for Crude trade data. This graph has a high similarity to the one in Figure 3.10. However, there are three exporter groups in this graph whereas there are four in Figure 3.10. African countries Liberia and Ethiopia go together with African cluster, although they compromise a new cluster with East European countries Yugoslavia and Czech Republic in Figure 3.10. Furthermore, Czech Republic, Finland, Switzerland and Yugoslavia create a separate cluster rather than being together with first world countries as in Figure 3.10. Those two differences seem to be more plausible. On the other hand, Middle East countries go together with Far East countries while they were in a separate cluster in Figure 3.10, which may be more reasonable. We see USA, Central Europian countries and Japan as great Crude traders whereas third world African countries have almost no crude material import or export processes and second world Far East countries such as Thailand or China are

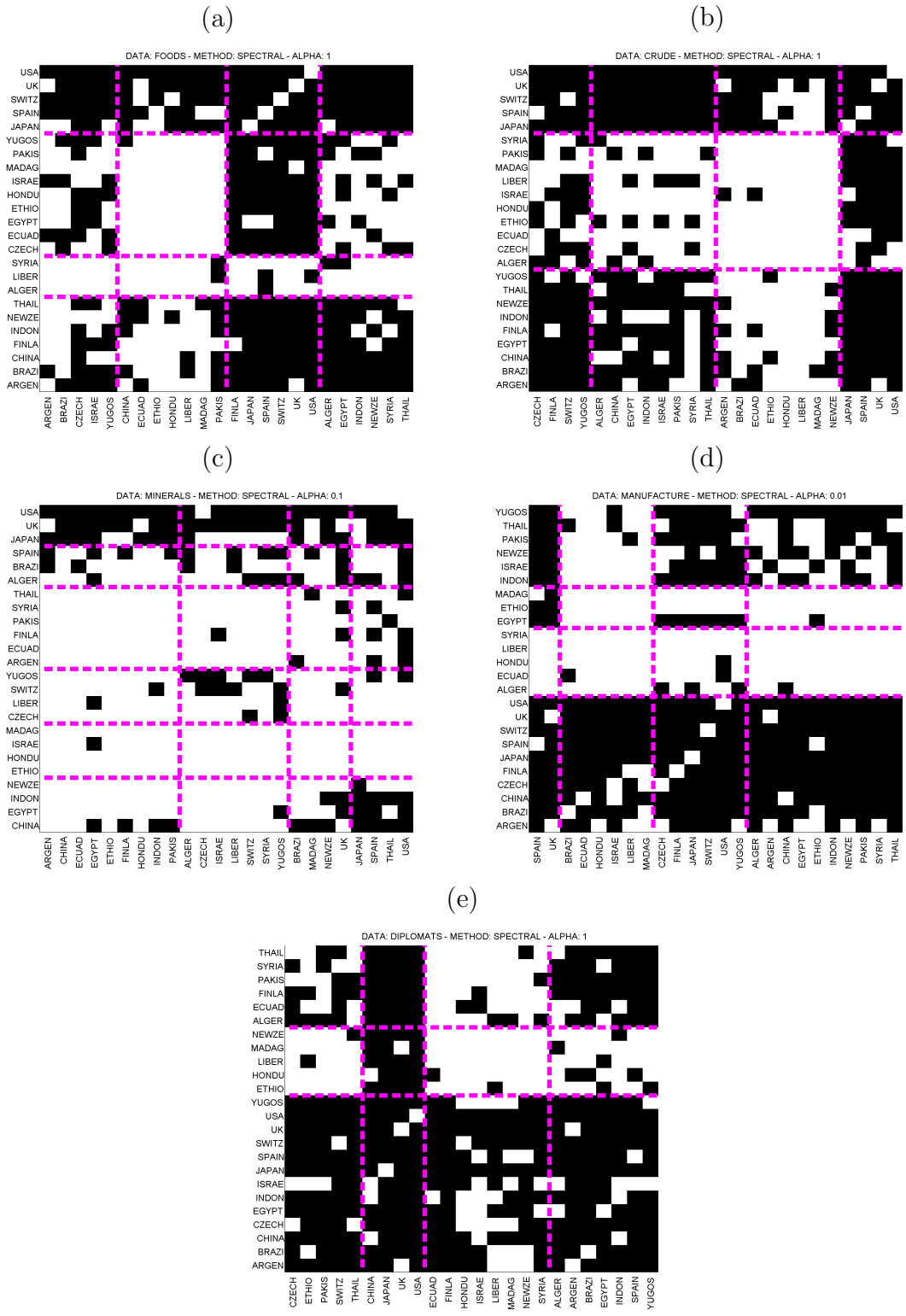


Figure 4.6. Spectral Biclustering results for Countries trade and diplomacy relations dataset. **(a)** Foods **(b)** Crude **(c)** Minerals **(d)** Manufacture **(e)** Diplomats. Dashed lines separate the partitions. Data are rearranged according to indicators bringing about the maximum marginal data likelihood. Original dataset is shown in Figure 3.1.

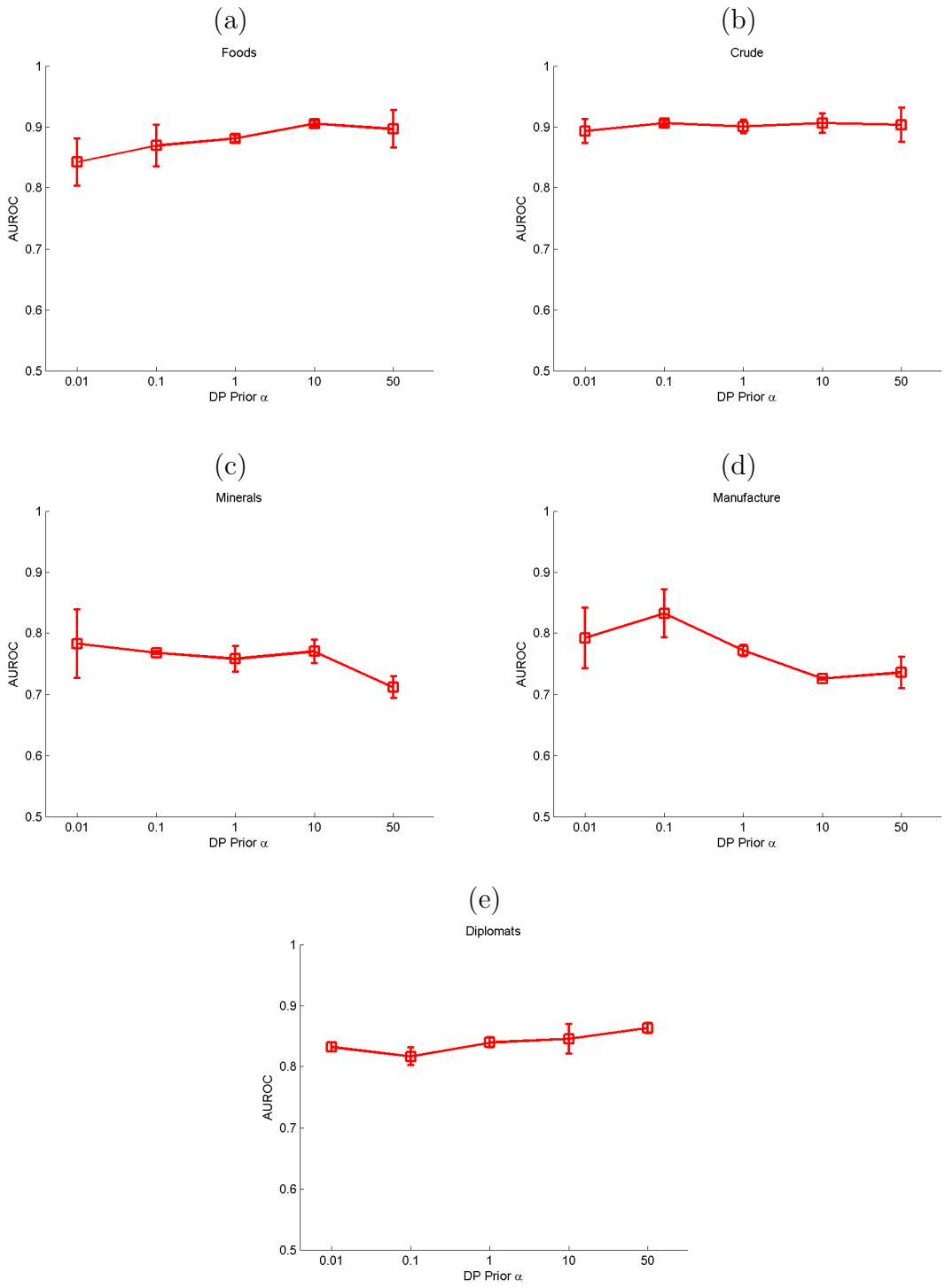


Figure 4.7. Spectral Biclustering performances for Countries dataset with respect to different Dirichlet priors (α). **(a)** Foods **(b)** Crude **(c)** Minerals **(d)** Manufacture **(e)** Diplomats. The AUROC scores for Minerals and Manufacture are relatively low since those data matrices have rows with all zeros and they have zero eigenvalues.

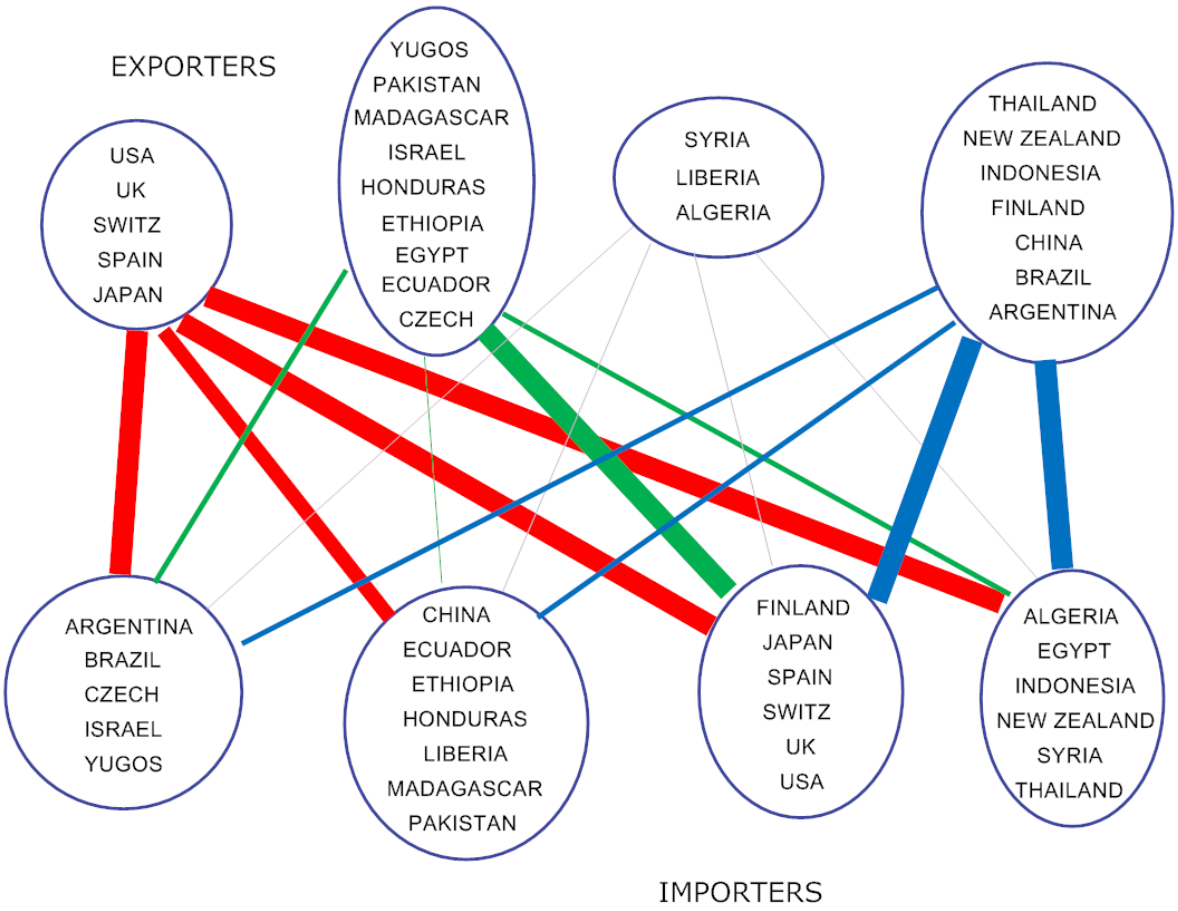


Figure 4.8. Spectral Biclustering results are represented as the relationship graph for Foods data. Thick lines between two groups of countries correspond to intense relationships between them whereas thin lines represent a lack or spareness of relation.

important crude material exporters.

In Figure 4.10, the biclustering results are represented for Crude trade data. Compared to DPMM clustering result shown in Figure 3.11, there are more exporter groups, however the results are still similar. Whereas two European countries Spain and Switzerland are generally in the same exporter or importer group for all other four types of trade, this is not the case for minerals trade. Spain seems to be a more active mineral exporter. Although USA and UK are always together in the same cluster in almost all of the biclustering result graphs together with Figure 3.11, they are in separate clusters in this graph. This may be the weak point of this spectral result.

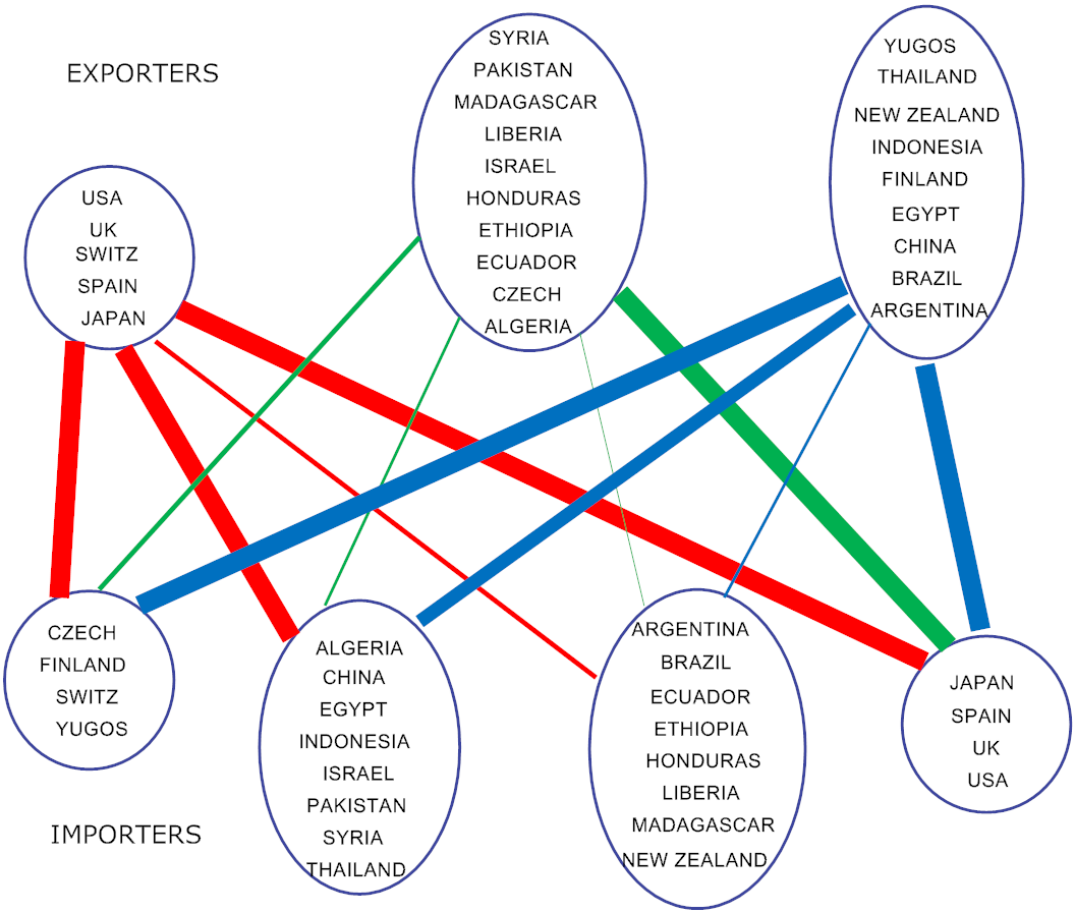


Figure 4.9. Spectral Biclustering results are represented as the relationship graph for Crude data. Thick lines between two groups of countries correspond to intense relationships between them whereas thin lines represent a lack or spareness of relation.

In Figure 4.11, the biclustering results are demonstrated for Crude trade data. In addition to central European countries and USA, other European countries, Japan, China and South American countries goes together with leading manufactured material exporter countries. Whereas Brazil and Argentina were in separate clusters in Figure 3.12, here those South American countries are reported to have similar export behaviors. Most of the African countries have almost no manufactured material export or import similar to DPMM result in Figure 3.12. Leader manufactured material exporters seem to be not much active in manufactured material import task since they are countries developed in industry and they fabricate the crude materials themselves.

In Figure 4.12, the biclustering results are demonstrated for Crude trade data.

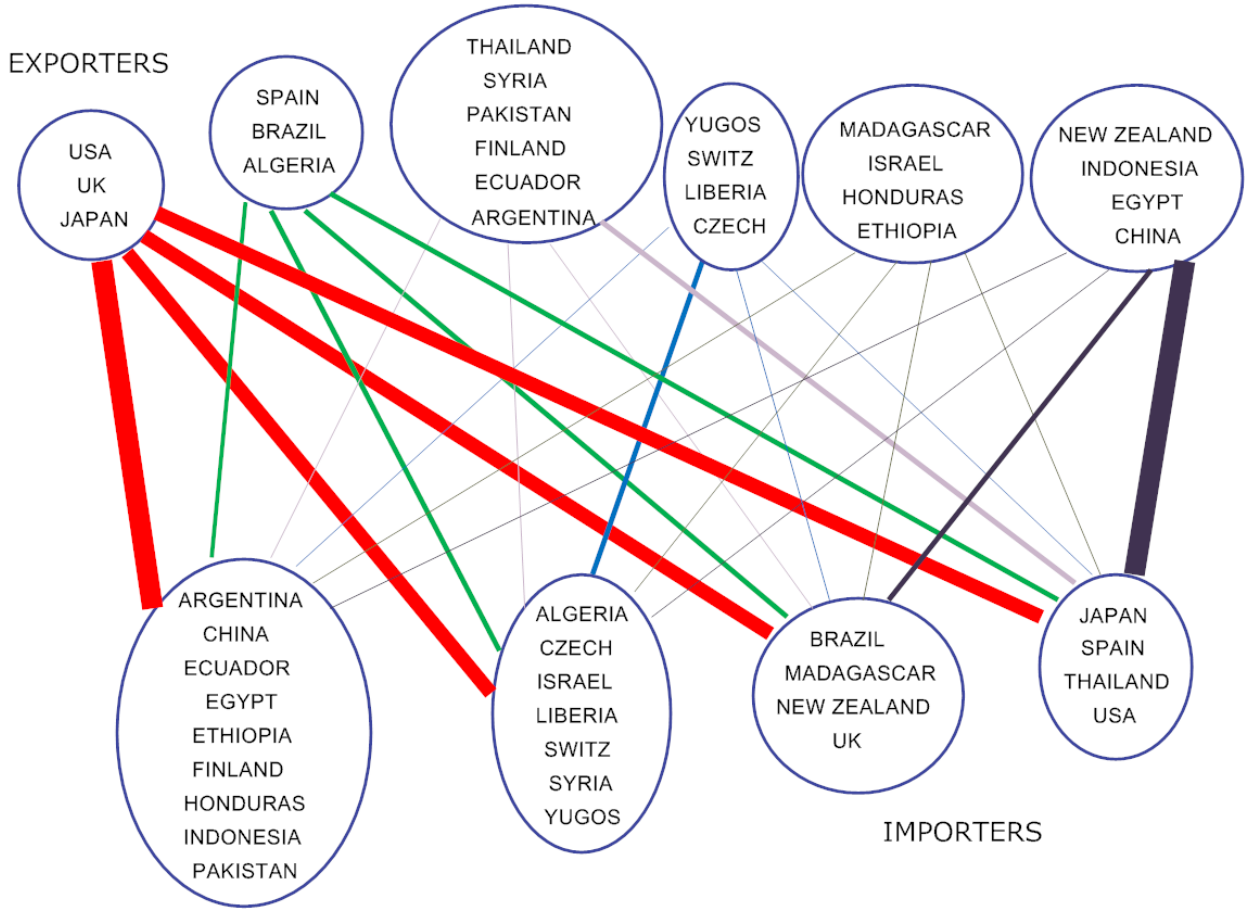


Figure 4.10. Spectral Biclustering results are represented as the relationship graph for Minerals data. Thick lines between two groups of countries correspond to intense relationships between them whereas thin lines represent a lack or spareness of relation.

Whereas this spectral result graph reports less export clusters than the DPMM result graph in Figure 3.13, likewise in Figure 3.13, China, Japan, UK and USA create a separate import cluster. Since they are countries having close relationships with the rest of the world, this result is much reasonable. Since diplomatic relations are much more intense and symmetric than trade relations, all European countries' including the Easter being in the same export cluster here in this graph is plausible. African countries are still in the last place in both export and import.

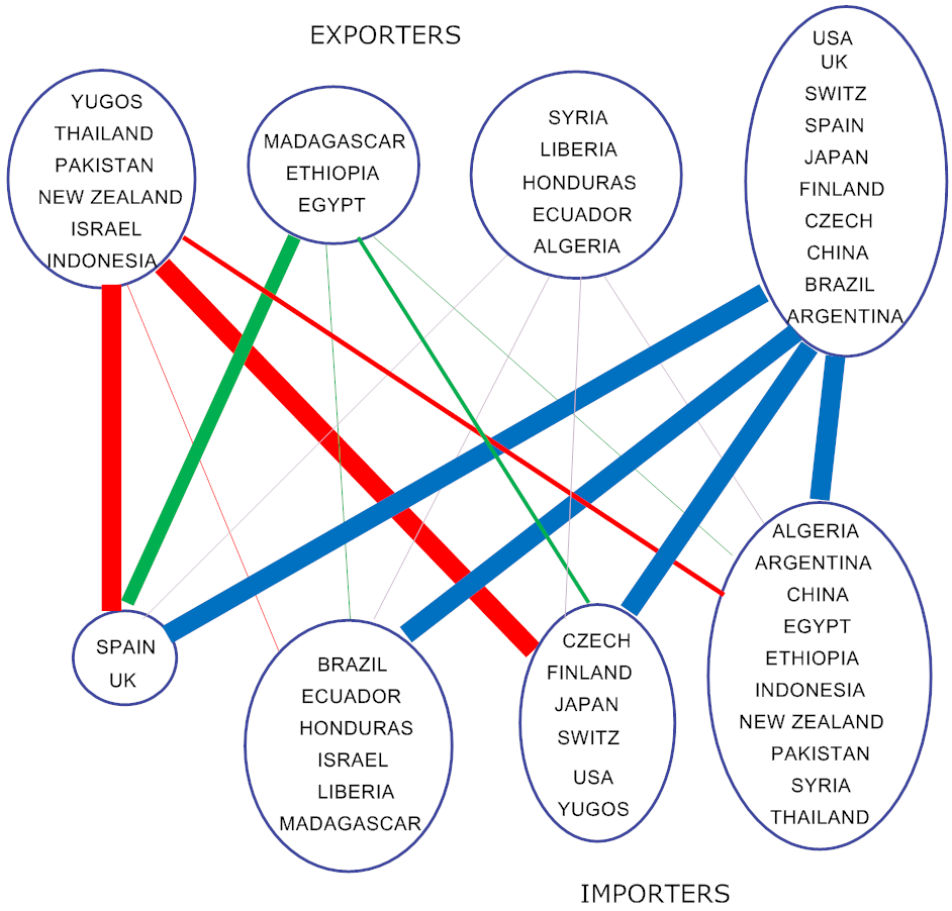


Figure 4.11. Spectral Biclustering results are represented as the relationship graph for Manufacture data. Thick lines between two groups of countries correspond to intense relationships between them whereas thin lines represent a lack or spareness of relation.

4.3.3. Comparison to DPMM Biclustering

Since we use the information inside the data representation completely, DPMM biclustering approach in Chapter 3 provides us with all a nonparametric mixture model can. However, since we have to run a number of simulations on both rows and columns at each iteration, MCMC sampling, which is already time-consuming, takes too long to get to the end. On the other hand, in case of the Spectral Biclustering, dimensionality of the data to process is highly reduced prior to the MCMC iterations, which fairly decreases the runtime of the algorithm. In Figures 4.13 and 4.14, the performances of two nonparametric methods are compared for different values of DP prior α for Countries and Lung Cancer datasets respectively.

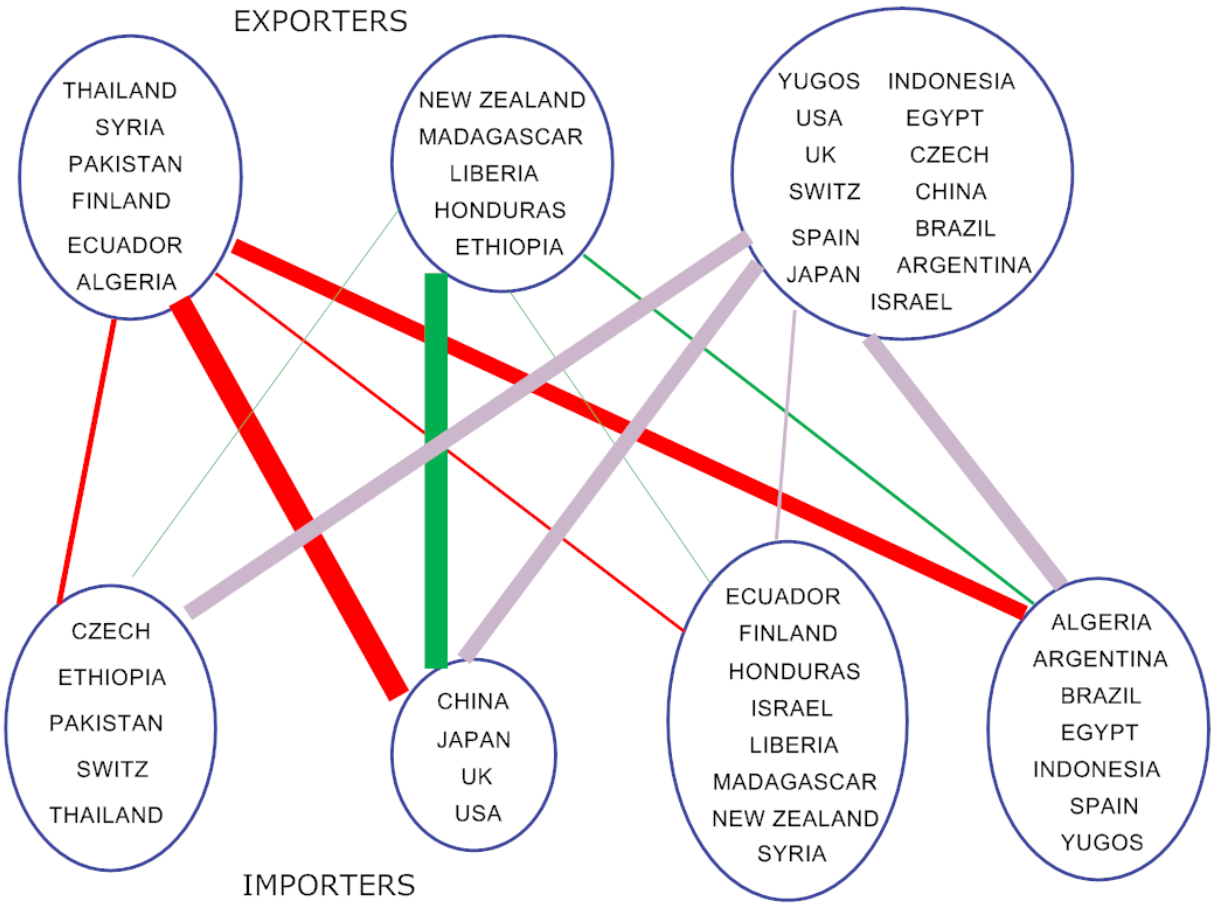


Figure 4.12. Spectral Biclustering results are represented as the relationship graph for Diplomats data. Thick lines between two groups of countries correspond to intense relationships between them whereas thin lines represent a lack or spareness of relation.

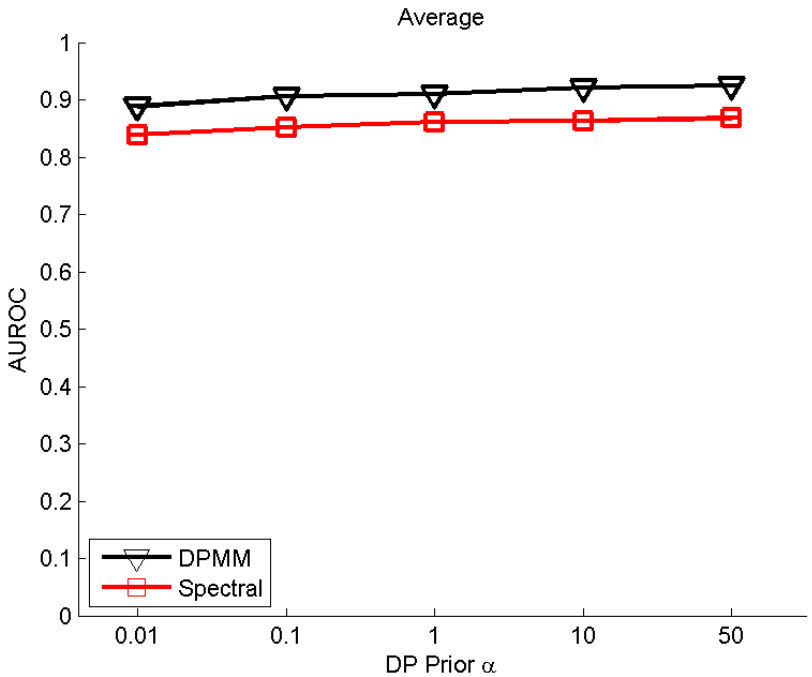


Figure 4.13. Average performances of DPMM and spectral biclustering methods for five different networks in Countries dataset. DPMM biclustering performs a bit better than spectral clustering for all values of α. Moreover, for all values of α, we get high biclustering performances, namely AUROC scores close to 1.

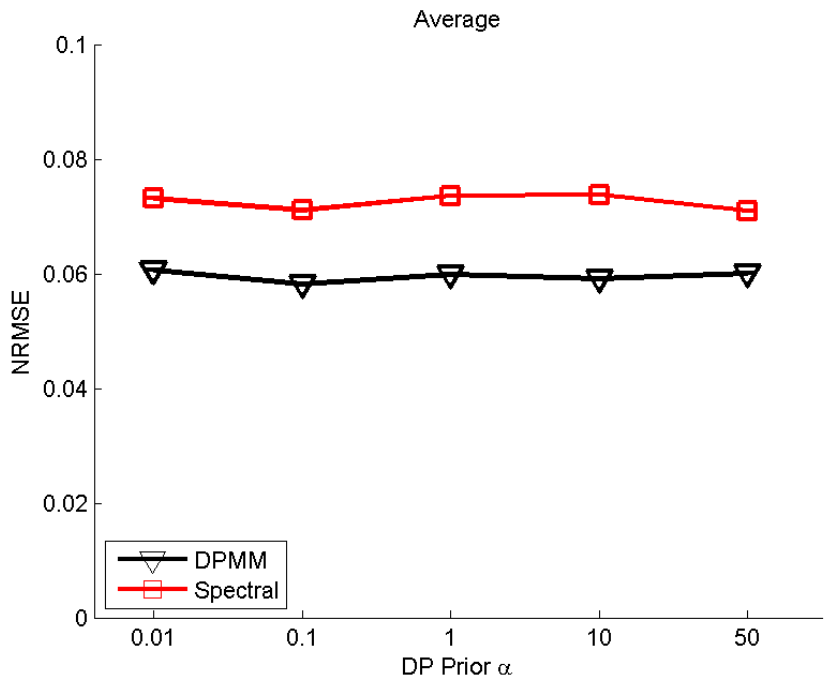


Figure 4.14. Average performances of DPMM and spectral biclustering methods for Lung Cancer dataset. DPMM biclustering performs a bit better than spectral clustering for all values of α. Moreover, for all values of α, we get high biclustering performances, namely NRMSE scores close to 0.

5. CONCLUSION

The aim of this thesis was to investigate nonparametric Bayesian methods for biclustering relational data. We mainly utilized DPs to obtain the nonparametric nature of those algorithms. Where it is hard to decide the number of clusters in the mixture data in advance, which is generally the case, DPs take to the stage. They are one's rescuers in that they remove the problem of underfitting or overfitting problem parametric model selection brings about. For inference purposes, we have run MCMC simulations which enable applying nonparametric Bayesian methods to a variety of data analysis problems. In our experiments, we evaluated perofrmances of algorithms in terms of two basic issues; biclustering and missing value imputation.

We applied two types of infinite models on two-way data. First approach is direct biclustering of data matrix using DPMMs, which we discussed in Chapter 3. For this approach, we modeled rows and columns with separate DPs and obtain the cluster parameters as the entry pairs using the latent row and column cluster indicators. Since what we need is the cluster assignments, that's, which rows/columns goes together in a cluster, we integrated out the cluster parameters and apply CGS.

Secondly, in Chapter 4, we combined spectral matrix operations with nonparametric Bayesian priors. Spectral decomposition of a matrix aims to reduce a dataset containing a large number of values to a dataset with significantly fewer values. However, the large fraction of the variability present in the original data is conserved in the resulting dataset, and this is the reason what makes spectral decomposition methods to be widely used in a variety of mathematical applications. In the spectral biclustering method, we firstly decomposed the data to its eigenvectors, and retrieved a few eigenvectors for both rows and columns each to use for clustering purposes in the next step. Then we modeled the 2D eigenvector space using DPs for both rows and columns, and retrieve the clusters via the combination of row/column indicators. Due to the dimensionality reduction operation prior to the DPMM modeling task, the second method, namely the Spectral Biclustering resulted in less time and space complexity whereas it

gives a bit less accurate biclustering output.

For both types of nonparametric methods, just after CGS at each iteration, we also performed five random split merge procedures on rows/columns. This was necessary to avoid getting stuck of CGS in local minimum dips. However, since MCMC methods are based on sampling and many iterations have to be performed, they result in high computational complexity. Performing split-merge procedures at each iteration add up to this complexity. Therefore code optimization turns out to be a very significant issue to consider while implementing MCMC-based nonparametric Bayesian biclustering techniques.

Mainly because of their simplicity and practicality, nonparametric Bayesian methods have many potential extensions and applications providing many future directions to follow. As a future work, we especially intend to test the performance of the nonparametric spectral biclustering algorithm, which has few applications in the literature on different datasets including biological time-series data.

We modelled two-way data in this thesis. However, the algorithms are easily extendible to multiway datasets. As a future direction, we intend to model and infer clusters in multiway data using the two methods we have discussed.

REFERENCES

1. Shan, H., *Probabilistic Models for Multi-relational Data Analysis*, Ph.D. Thesis, University of Minnesota, 2012.

2. Dhillon, I. S., "Co-clustering Documents and Words Using Bipartite Spectral Graph Partitioning", *7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 269–274, 2001.

3. Kluger, Y., Y. Basri, J. T. Chang, and M. Gerstein, "Spectral Biclustering of Microarray Cancer Data: Co-clustering Genes and Conditions", *Genome Research*, Vol. 13, p. 703, 2003.

4. Taskar, B., E. Segal, and D. Koller, "Probabilistic Classification and Clustering in Relational Data", *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 870–876, 2001.

5. Dhillon, I. S., S. Mallela, and D. S. Modha, "Information Theoretic Co-clustering", *9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 89–98, 2003.

6. Shan, H. and A. Banerjee, "Bayesian Co-clustering", *IEEE International Conference on Data Mining*, pp. 530–539, 2008.

7. Alpaydın, E., *Introduction to Machine Learning*, pp. 105–106, 153, The MIT Press, 2004.

8. Meeds, E., *Nonparametric Bayesian Methods for Extracting Structure From Data*, Ph.D. Thesis, University of Toronto, 2008.

9. Jain, A. K., M. N. Murty, and P. J. Flynn, "Data Clustering: A Review", *ACM Computing Surveys*, Vol. 31, 1999.

10. Sibson, R., "SLINK: An Optimally Efficient Algorithm for the Single-Link Cluster Method", *The Computer Journal (British Computer Society)*, Vol. 16, pp. 30–34, 1973.

11. Defays, D., "An Efficient Algorithm for a Complete Link Method", *The Computer Journal (British Computer Society)*, Vol. 20, p. 364–366, 1977.

12. Dempster, A. P., N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society, Series B (Methodological)*, Vol. 39, p. 1–38, 1977.

13. Lloyd, S., "Least Squares Quantization in PCM", *IEEE Transactions on Information Theory*, Vol. 28, pp. 129–137, 1982.

14. Kriegel, H.-P., P. Kröger, J. Sander, and A. Zimek, "Density-based Clustering", *WIREs Data Mining and Knowledge Discovery*, Vol. 1, p. 231–240, 2011.

15. Jain, A. K., "Data Clustering: 50 Years Beyond K-Means", *Pattern Recognition Letters*, Vol. 31, p. 651–666, 2010.

16. Meeds, E. and S. Roweis, "Nonparametric Bayesian Biclustering", Technical report, 2007.

17. Roy, D. M. and Y. W. Teh, "The Mondrian Process", *Advances in Neural Information Processing Systems*, Vol. 21, 2009.

18. Madeira, S. C. and A. L. Oliveira, "Biclustering Algorithms for Biological Data Analysis: A Survey", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 1, pp. 24–45, 2004.

19. Hartigan, J. A., "Direct Clustering of a Data Matrix", *Journal of the American Statistical Association*, Vol. 67, pp. 123–129, 1972.

20. Eren, K., M. Deveci, O. Küçüktunç, and U. V. Çatalyürek, "A Comparative Anal-

ysis of Biclustering Algorithms for Gene Expression Data", *Briefings in Bioinformatics*, Vol. 13, pp. 24–45, 2012.

21. George, T. and S. Merugu, "A Scalable Collaborative Filtering Framework Based on Co-clustering", *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM)*, pp. 625–628, 2005.

22. Peng, W. and T. Li, "Temporal Relation Co-clustering on Directional Social Network and Author-Topic Evolution", Vol. 26, pp. 467–486, 2011.

23. Cheng, Y. and G. M. Church, "Biclustering of Expression Data", *8th International Conference on Intelligent Systems for Molecular Biology*, pp. 93–103, 2000.

24. Getz, G., E. Levine, and E. Domany, "Coupled Two-way Clustering Analysis of Gene Microarray Data", *Proceedings of the National Academy of Sciences (PNAS)*, Vol. 97, 2000.

25. Tanay, A., R. Sharan, and R. Kupiec, M Shamir, "Revealing Modularity and Organization in the Yeast Molecular Network by Integrated Analysis of Highly Heterogeneous Genomewide Data", *Proceedings of the National Academy of Sciences (PNAS)*, Vol. 101, 2004.

26. Gasch, A. P. and M. B. Eisen, "Exploring the Conditional Coregulation of Yeast Gene Expression Through Fuzzy K-means Clustering", *Genome Biology*, Vol. 3, 2002.

27. Hastie, T., R. Tibshirani, M. B. Eisen, A. Alizadeh, R. Levy, L. Staudt, W. C. Chan, D. Botstein, and P. Brown, "'Gene shaving' as a Method for Identifying Distinct Sets of Genes with Similar Expression Patterns", *Genome Biology*, Vol. 1, 2000.

28. Liben-Nowell, D. and J. Kleinberg, "The Link Prediction Problem for Social Networks", *Proceedings of the 12th Annual ACM International Conference on Infor-*

mation and Knowledge Management (CIKM), pp. 556–559, 2003.

29. Lü, L. and T. Zhou, "Link Prediction in Weighted Networks: The Role of Weak Ties", *EPL (Europhysics Letters)*, Vol. 89, 2010.

30. de Sa, H. R. and R. B. C. Prudencio, "Supervised Link Prediction in Weighted Networks", *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 2281–2288, 2011.

31. Getoor, L. and C. P. Diehl, "Link Mining: A Survey", *ACM SIGKDD Explorations Newsletter*, Vol. 7, pp. 3–12, 2005.

32. Hoseini, E., S. Hashemi, and A. Hamzeh, "Link Prediction in Social Network Using Co-clustering Based Approach", *26th International Conference on Advanced Information Networking and Applications Workshops*, pp. 795–800, 2012.

33. Shashua, A., R. Zass, and T. Hazan, "Multi-way Clustering using Super-symmetric Non-negative Tensor Factorization", *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 595–608, Book Chapters, 2006.

34. Ding, C., T. Li, W. Peng, and H. Park, "Orthogonal Nonnegative Matrix Tri-factorizations for Clustering", *ACM SIGKDD*, pp. 126–135, 2006.

35. Banerjee, A., S. Basu, and S. Merugu, "Multi-way Clustering on Relation Graphs", *Proceedings of the 7th SIAM International Conference on Data Mining*, 2006.

36. Ding, C., X. He, and H. D. Simon, "On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering", *Proceedings of the SIAM International Conference on Data Mining*, pp. 606–610, 2005.

37. Ferguson, T. S., "A Bayesian Analysis of Some Nonparametric Problems", *The Annals of Statistics*, p. 209–230, 1973.

38. Teh, Y. W., "Dirichlet Processes", *Encyclopedia of Machine Learning*, Springer,

2010.

39. Ghahramani, Z., "Non-parametric Bayesian Methods", IPAM Probabilistic Models of Cognition Lectures, Department of Engineering, University of Cambridge and Machine Learning Department, Carnegie Mellon University, 2007.

40. Sethuraman, J., "A Constructive Definition of Dirichlet Priors", *Statistica Sinica*, Vol. 4, pp. 639–650, 1994.

41. Blackwell, D. and J. B. Macqueen, "Ferguson Distributions via Pólya Urn Schemes", *The Annals of Statistics*, Vol. 1, pp. 353–355, 1973.

42. Teh, Y. W., M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet Processes", *Journal of the American Statistical Association*, Vol. 101, pp. 1566–1581, 2006.

43. Mauldin, R. D., W. D. Sudderth, and S. C. Williams, "Pólya Trees and Random Distributions", *The Annals of Statistics*, Vol. 3, pp. 1203–1221, 1992.

44. Neal, R. M., "Density Modeling and Clustering Using Dirichlet Diffusion Trees", *Bayesian Statistics*, Vol. 7, pp. 619–629, 2003.

45. Griffiths, T. L. and Z. Ghahramani, "Infinite Latent Feature Models and the Indian Buffet Process", *Advances in Neural Information Processing Systems*, pp. 475–482, MIT Press, 2005.

46. Ishwaran, H. and L. F. James, "Generalized Weighted Chinese Restaurant Processes for Species Sampling Mixture Models", *Statistica Sinica*, Vol. 13, pp. 1211–1211, 2003.

47. West, M., "Hyperparameter Estimation in Dirichlet Process Mixture Models", Technical report, Institute of Statistics and Decision Sciences, Duke University, 1992.

48. Teh, Y. W., "Bayesian Nonparametric Modelling", Machine Learning II Lectures, Gatsby Computational Neuroscience Unit, University College London, 2008.

49. Antoniak, C. E., "Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems", *The Annals of Statistics*, Vol. 2, pp. 1152–1174, 1974.

50. Neal, R. M., "Markov Chain Sampling Methods for Dirichlet Process Mixture Models", Technical report, Department of Statistics, University of Toronto, 1998.

51. Ghahramani, Z. and M. J. Beal, "Propagation Algorithms for Variational Bayesian Learning", *Advances in Neural Information Processing Systems*, pp. 507–513, MIT Press, 2001.

52. Jordan, M. I., "An Introduction to Variational Methods for Graphical Models", *Machine Learning*, pp. 183–233, MIT Press, 1999.

53. Blei, D. M. and M. I. Jordan, "Variational Inference for Dirichlet Process Mixtures", *Bayesian Analysis*, Vol. 1, pp. 121–144, 2005.

54. Bro, R., E. E. Papalexakis, E. Acar, and N. D. Sidiropoulos, "Coclustering—A Useful Tool for Chemometrics", *Journal of Chemometrics*, Vol. 26, p. 256–263, 2012.

55. Wasserman, S. and K. Faust, *Social Network Analysis: Methods and Applications*, pp. 64–65, Cambridge University Press, 1994.

56. Bhattacharjee, A., W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker, and M. Meyerson, "Classification of Human Lung Carcinomas by mRNA Expression Profiling Reveals Distinct Adenocarcinoma Subclasses", *Proceedings of the National Academy of Sciences (PNAS)*, Vol. 98, p. 13790–13795, 2001.

57. Lee, M., H. Shen, J. Z. Huang, and J. S. Marron, "Biclustering via Sparse Singular Value Decomposition", *Biometrics*, Vol. 66, pp. 1087–1095, 2010.

58. Liu, Y., D. N. Hayes, A. Nobel, and J. S. Marron, "Statistical Significance of Clustering for High-Dimension, Low-Sample Size Data", *Journal of the American Statistical Association*, Vol. 103, pp. 1281–1293, 2008.

59. Jung, S. and J. S. Marron, "PCA Consistency in High Dimension, Low Sample Size Context", *Annals of Statistics*, Vol. 37, pp. 4104–4130, 2009.

60. Jain, S. and R. Neal, "A Split-Merge Markov Chain Monte Carlo Procedure for the Dirichlet Process Mixture Model", *Journal of Computational and Graphical Statistics*, Vol. 13, pp. 158–182, 2000.

61. Clauset, A., C. Moore, and M. E. J. Newman, "Hierarchical Structure and the Prediction of Missing Links in Networks", *Nature*, Vol. 453, pp. 98–101, 2008.

62. Alpert, C. and S.-z. Yao, "Spectral Partitioning: The More Eigenvectors, The Better", *32nd ACM/IEEE Conference on Design Automation*, p. 195–200, 1995.

63. Socher, R., A. Maas, and C. D. Manning, "Spectral Chinese Restaurant Processes: Nonparametric Clustering Based on Similarities", *14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.