Quality Analysis of Correlation Clustering

Mamata Samal

Quality Analysis of Correlation Clustering

Thesis submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

by

Mamata Samal

Under the supervision of

Dr. V. Vijaya Saradhi
Prof. Sukumar Nandi

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
Guwahati 781039, India
May 2015

Dedicated to

The Almighty God who made it happen

Declaration

I certify that

a. The work contained in this thesis is original and has been done by myself and the general supervision of my supervisors.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

d. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT Guwahati
Date:

Mamata Samal
Research Scholar
Department of Computer Science and Engineering,
Indian Institute of Technology Guwahati,
Guwahati, Assam, INDIA 781039

Certificate

This is to certify that the thesis entitled "**Quality Analysis Of Correlation Clustering**" being submitted by **Mamata Samal** to the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, is a record of bona fide research work under our supervision and is worthy of consideration for the award of the degree of Doctor of Philosophy of the Institute.

Dr. V. Vijaya Saradhi
Department of CSE,
Indian Institute of Technology Guwahati,
Guwahati, Assam, INDIA 781039
Date:

Prof. Sukumar Nandi
Department of CSE,
Indian Institute of Technology Guwahati,
Guwahati, Assam, INDIA 781039
Date:

Acknowledgements

First of all I would like to thank the Almighty God for giving me strength and patience to complete my thesis.

At this moment of accomplishment, I would like to express my sincere gratitude to my supervisors Dr.V.Vijaya Saradhi and Professor Sukumar Nandi, for their continuous support and proper guidance during my PhD work. Without their support this dissertation would not have been possible. I am indebted to Dr.V.Vijaya Saradhi for providing me enough freedom to express myself and I appreciate his thoughtful insight, placid composer and tireless editing of the thesis. Thank you Sir for all your support. I would like to express my gratitude to Professor Sukumar Nandi for his intellectual suggestions and definitive advices which helped me immensely to carry out my research.

I would like to express my sincere appreciation to my Doctoral committee Dr.G. Sajith, Dr.S.V.Rao and Dr.Sukant Pati for their valuable feedback, suggestions and encouragement.

I am also thankful to the entire faculty members of Computer Science & Engineering Department of IIT Guwahati for their support and help. I would like to wholeheartedly acknowledge the Department of Computer Science and Engineering, IIT Guwahati for the facilities provided to me during my research work. I would like to thank Scientific Officers Nanu, Bhirguraj, Raktajit and office staff Malakar, Nabo, Souvik, Prashanta and Prabin for helping and providing me the materials at time of need.

I would always cherish the unforgettable time I spent with my friends Satyashree, Lipika, Biduyt and Rajendra. I would like to extend my sincere thanks to Srinivasa Sir, Shashi, and Pravati who were always ready to share all my personal and work related problems. Their spontaneous and friendly nature along with the helpful suggestions from time to time gave me the strength to fight the difficult times during my journey. My special thanks to all my friends Nilu, Shirshendu, Mayank, Awinsh, Satish, Sibaji, Shilpa, Basant and Debanjan for adding charm to my life at IITG.

No acknowledgement would be complete without mentioning special thanks to my parents, Nana, Bhauja , Chintu and Mantu for their unconditional love, support, encouragement and believe on me. I am also grateful to my in-laws for their patience and support. I am also indebted to Suna, Sumi and Dugu for their support during my stay at IITG.

I would love to thank each and everyone who is directly or indirectly responsible for the successful completion of my research work.

Last but not the least I would like to thank my husband Sraban and my daughter Shree for their selfless love, support and patience. Without them especially my daughter Shree, this journey would had never reached the destination.

Place: IIT Guwahati

Date: **Mamata Samal**

Abstract

Correlation clustering (CC) is a graph based clustering method. This method assumes that the graph G with a set of nodes and associated relationships, is available beforehand to perform clustering. The nature of relationship between two nodes in G is limited to specifying whether two nodes are *related* or *not related* to one another. MAXAGREE is a well known objective in obtaining partitions of G which maximizes the agreement of the edge labels within a cluster and maximize the disagreement of the edge labels across clusters. Despite several theoretical results, CC has not gained popularity among practitioners in the machine learning and data mining community. The main challenges in applying CC in practice are:

1. CC assumes the availability of graph. Construction of the graph and its type from a given vector dataset are however not part of CC. In many practical applications, graphs are not naturally available. One needs to model the dataset at hand in the form of a graph for application of CC.

2. A relaxed version of MAXAGREE is re-formulated as a semidefinite programming (SDP) formulation. SDP formulations are expensive to solve when large number of optimization variables are involved. As the dataset size available for clustering increases, solving SDP formulations pose challenge.

3. MAXAGREE formulation yields soft cluster solution. To obtain hard cluster solution, one needs to employ rounding techniques. The implication of various rounding techniques has been theoretically analyzed. However, their application in practice is not explored in the literature given the various dataset characteristics.

4. Being a variant of constraint clustering, comparing CC's performance with other constraint clustering methods is a necessity for establishing CC's applicability in practice.

In this thesis, all the above identified points are examined carefully and proposed applicability of CC in practice through analysis of quality of clusters generated.

Contents

List of Figures

List of Tables

Nomenclature

BFGS	Broyden Fletcher Goldfarb Shanno method
CC	Correlation Clustering
cSC	Constraint Spectral Clustering
COP-KMEANS	Constraint K-means clustering
CVQE	Constrained Vector Quantization Error
FCSC	Flexible Constrained Spectral Clustering
FI	Fisher Information
GES	Graph Edge Sharpening
GrBias	Grouping with Bias
K-NN	K-Nearest Neighbor
LP	Linear Programming
LCVQE	Linear time Constrained Vector Quantization Error
ML	Marginal Likelihood
MSB	Multilevel recursive Spectral Bisection
MPC K-means	Metric Pairwise Constrained K-Means
PTAS	Polynomial Time Approximation Scheme
RPR^2	Random Projection and Randomized Rounding
SDP	Semi Definite Programming
SL	Spectral Learning algorithm
SCLP	Spectral Constraint clustering with Local Proximity measure
SDP-CC	Semi Definite Programming of Correlation Clustering
SKL	Spectral Kernel Learning
SSDP-CC	Scalable Semi Definite Programming of Correlation Clustering
RC SDP-CC	Reduced Constraint Semi Definite Programming of Correlation Clustering
SDPNAL	A MATLAB software for semidefinite programming based on a semi-smooth Newton-CG augmented Lagrangian method
SDPT3	A MATLAB software for semidefinite-quadratic-linear programming

List of Symbols

V_p	Set of positively correlated vertices
V_g	Set of vertices of graph G
G	Graph
E	Set of Edges
E_{one}	Matrix of all ones
'+'	Positively labeled edges
'-'	Negatively labeled edges
n	Number of vertices or number of data points
w_{ij}	Weight specified on the edge (i, j)
x_{ij}	Indicator variable
C	Coefficient Matrix
X	Variable Matrix
$\mathbf{v}_i$	Unit vector
W_{in}	Denotes the weight of edges which lie inside a partition
W_{out}	Denotes the weight of edges which lie between any pair of partitions
$diag(V)$	Denotes diagonal of matrix V
$Diag(\mathbf{v})$	Denotes a diagonal matrix whose diagonal elements are the elements of vector $\mathbf{v}$
$\mathbb{R}^n$	Set of real numbers
$\mathbf{v}_i^*$	Solution vector
$\mathbf{r}$	A random vector
S_+, S_-	Clusters obtained through CC
$\mathbf{e}$	Vector of all ones
θ_{ij}	Angle between vectors $\mathbf{v}_i^*$ and $\mathbf{v}_j^*$
γ	Positive real number
W_{total}	Total weight of the edges of the graph

LIST OF SYMBOLS

Z	Objective function value of CC
A_{ratio}	Ratio of Z to total weight of the edges of the graph
C_i	i^{th} cluster
q	Dimension of the dataset
$\mathbf{x}_i$	i^{th} Data point
W	Affinity matrix
$\|\cdot\|_F$	Frobenius norm
d_i	Degree of i^{th} vertex
L	Graph Laplacian matrix
M	Edge encoded matrix
A	Matrix obtained by considering absolute values of matrix U
$a(\mathbf{x})$	Generalized linear estimator of a
β_0	Regression coefficient
$\boldsymbol{\beta}$	Regression coefficients
$c_=$	Must link
$c_\neq$	Cannot link
D	Diagonal matrix
Q	Constraint matrix
J_{cSC}	Constraint spectral clustering objective
J_{SC}	Spectral clustering objective
J_{CM}	Cannot-link and must-link objective
L_Q	Laplacian matrix of the constrained graph
D_Q	Degree matrix of constraints graph
$\text{vol}(\mathcal{G})$	Denote the total degree of vertices
o	Denote element wise multiplication of two given matrices
R	Rectangular Matrix
$V \succeq 0$	V is a positive semidefinite matrix
R_{ij}^0	Initial value of matrix R

Chapter 1

Introduction

Correlation clustering (CC) is a graph based clustering method. This method assumes that the graph G with a set of nodes and associated relationships is available beforehand to perform clustering. The nature of relationship between two nodes in G is limited to specifying whether two nodes are related to each other or not. CC may take into account the strength of relationship between nodes in terms of weights between vertices of G. Directionality information is also not assumed while performing clustering. Unlike other clustering methods, CC does not depend on any information other than the node relationship and associated weights.

Note that the expressed relationship between nodes need not adhere to derived relationships. For example, if two pair of nodes v_1 and v_2 are positively correlated, v_2 and v_3 are positively correlated, then the derived relationship between nodes v_1 and v_3 need not adhere to the positive correlation. Let a group of nodes $V_p = \{v_1, v_2, \cdots, v_l\}$ be positively correlated. Let two nodes v_i, v_{l+1} are negatively correlated where $v_i \in V_p$. In CC, given the fact that v_{l+1} negatively correlated with v_i does not imply that v_{l+1} is negatively correlated with all the nodes in V_p. Despite these contradictory relations, CC algorithm must produce a cluster solution which adheres to *all* the specified relations.

Clustering methods assume that the class label information associated with data points is not available. In recent years, there is an increasing interest to incorporate domain knowledge about subset of data points to obtain cluster solution. This calls for new ways of incorporating domain knowledge into existing clustering methods. These class of clustering methods are well known as *constrained clustering* [8]. The domain knowledge is expressed in terms of two well known constraints: *must-link* and *cannot-link*. Must-link constraint specifies that two data points *must be part of a same cluster* in the resulting

clustering solution. Cannot-link constraint specifies that two data points *cannot be part of a same cluster* in the resulting clustering solution. These two constraints are also referred to as instance level constraints.

CC is an extended case of constraint clustering in which information about data points is available only in the form of constraints. In other words, relation between every pair of data points is available. CC methods have been extensively studied in the literature [6, 15, 16, 31]. These studies are limited to theoretical development. This thesis makes an informed effort in analyzing quality of clusters obtained by CC from the application point of view. Limitations are identified in applying CC to practical problems and solutions are proposed for the identified problems.

1.1 Correlation Clustering

Given a complete graph $G = (V_g, E)$ with V_g as the set of vertices and E as the set of edges with every edge labeled either '+' or '-', CC aims at obtaining partitions of G. Three objective functions are proposed to obtain partitions of G. First one is MAXAGREE in which the objective is to maximize the agreement of the edge labels within a cluster and maximize the disagreement of the edge labels across clusters. Second one is MINDISAGREE in which the objective is to minimize the disagreement within clusters and the agreement in between the clusters. Third one is derived using the above two objectives and is known as MAXCORR. It maximizes difference between MAXAGREE and MINDISAGREE. Note that one does not need to provide as input the number of clusters to be obtained for CC.

The MAXAGREE formulation for different graphs such as complete graphs, general graphs and general weighted graphs are studied in [6, 16, 31]. For general graphs, which include weighted undirected graphs, Giotis *et al.* [31] proposed a two-cluster CC formulation, known as MAXAGREE2. This formulation has a promising approximation value of 0.87856. This thesis focuses on the MAXAGREE formulation and analyzes the quality of the clustering solution obtained by this formulation. This thesis confines to two-cluster solutions.

1.2 Motivation of the Research Work

Despite several theoretical results, CC has not gained popularity among practitioners in the machine learning and data mining community. Challenges in applying CC in practice

are:

1. CC assumes the availability of graph G, type of G (complete, general, or bipartite). However, construction of G from a given vector dataset is beyond the scope of CC. In many practical applications, graphs are not naturally available. One needs to model the dataset at hand in the form of a graph.

2. The MAXAGREE formulation is a polynomial time algorithm with time complexity of $O(n^{4.5})$ [77] where n is the number of vertices in the graph. As the dataset size available for clustering increases, solving MAXAGREE formulations pose a challenge.

3. MAXAGREE formulation yields soft cluster solution. To obtain hard cluster solution, one needs to employ rounding techniques. The implication of various rounding techniques has been theoretically analyzed. Their application in practice however is not explored in the literature when various dataset characteristics are given.

4. Being a variant of constraint clustering, comparing CC's performance with this class of clustering provides insights into CC's merits.

1.3 Contributions of the Thesis

This thesis focuses on **empirical study** of CC. The above identified challenges are closely examined. In particular the following problems are addressed in this thesis:

1. Sensitivity of rounding techniques on the quality of CC.

2. Role of graph construction methods on the quality of CC. The role of **optimal graph construction methods** is explored in particular.

3. Being a variant of constraint clustering, comparing CC's performance with other constraint clustering methods is a necessity for establishing CC's applicability in practice.

4. Proposed two variants of scalable CC formulation and applied on real world datasets and **large scale benchmark graph datasets**.

The following subsections elaborate on each of the above outlined contributions.

1.3.1 Sensitivity to Rounding Techniques

The MAXAGREE formulation on complete graph solves a semidefinite programming (SDP) formulation [77]. Resulting solution yields soft clusters. The soft cluster solution is in turn subjected to rounding procedure to obtain hard cluster solution. CC employs a naive rounding technique, namely hyperplane rounding technique [32]. The use of other rounding techniques in CC to obtain better approximation values is not studied.

The implication of applying various rounding techniques in the context of CC is examined empirically. The MAXAGREE formulation of CC has been analyzed thoroughly from the internal quality perspective, namely approximation value by employing hyperplane rounding technique. The influence of the dataset characteristics on the obtained cluster quality for a given rounding technique is not examined closely in the literature. This thesis explores the following points for the first time which is one of the contributions.

1. The impact of dataset characteristics on the quality of obtained clusters when different rounding techniques are employed.
2. The application of outward rotation as well as random projection and randomized rounding (RPR^2) techniques in the context of CC to analyze the quality of the obtained clusters.

The following points are the outcome of the empirical study:

1. Rounding techniques are sensitive to dataset characteristics.
2. Contradictions from the theoretical findings on the outward rotation and RPR^2 methods are observed from the experimentation.
3. Superiority of hyperplane rounding technique under certain dataset characteristics is shown.

1.3.2 Role of Optimal Graph Construction Methods

The following are the assumptions made by CC:

1. **Availability of Graph G:** CC assumes that a graph is available before hand for clustering. Many machine learning and data mining applications provide vector data points for clustering. In case of graph clustering methods, one explicitly constructs a graph out of the given vector data points. The process of graph construction

influences the clustering quality. Often graph clustering (including CC) methods do not take into account the methods involved in obtaining graph.

2. **About the Type of Graph:** CC assumes that the given graph is of a particular type and solves MAXAGREE formulation accordingly. This information is crucial as the MAXAGREE optimization formulations vary according to the type of graph. Role of the type of graph has been extensively studied and approximation values are proposed for three types of graphs namely complete, general and bipartite [2, 6, 16]. Their construction however, is not taken into account.

3. **Edge Labels:** Every edge needs to be labeled either '+' or '-' depending on the correlation between two vertices of G. The process of labeling is not taken into account for partitioning the graph. Labeling however, influences the quality of the obtained clusters.

CC is sensitive to all the above points. A principled way of constructing graphs from vector data is explored in this thesis as another contribution to address the above points. Two recent approaches for *optimal* graph construction are considered:

1. Fitting graph to vector data for obtaining an optimal weighted undirected general graph is referred as `fitgraph`.

2. A distance metric that uses Fisher information in constructing optimal complete graph.

The impact of optimal graphs on internal quality, external quality and time taken to obtain cluster solution is experimentally observed to have an edge over non-optimal graphs. Experiments performed on synthetic and real world datasets show that CC employed using both optimal and non-optimal graphs **converge**; That is both internal and external quality indices stabilize as the number of data points to be clustered increase exponentially. In some cases optimal graphs are observed to converge faster than non-optimal graphs.

1.3.3 Comparison of Different Constraint Clustering Algorithms with CC

Constrained clustering is a young area in semi-supervised learning. Various constrained clustering algorithms are proposed based on K-means algorithm (which uses vector dataset) and spectral clustering algorithm (which uses graph dataset) [78, 80, 81, 86]. In

the literature spectral constraint clustering algorithm (cSC) [78] and constraint spectral learning algorithm (SL) [39] are compared for their performance and it has been shown that cSC algorithm outperforms SL algorithm [78] when the number of constraints are less. Flexible constrained spectral clustering (FCSC) [81] algorithm is compared to other two existing clustering algorithms, namely SL algorithm [39] and Grouping with bias algorithm (GrBias) [88] and shown that the FCSC algorithm's performance is consistently better than the other two algorithms. Spectral constraint clustering with proximity measure (SCLP) [86] algorithm's performance is compared with SL algorithm [78] and it is shown that SCLP algorithm outperforms SL algorithm.

CC, being a variant of constrained form of clustering next contribution of the thesis is a comparative study with a class of constrained clustering methods. Specifically, performance of four well known constrained clustering algorithms are compared with CC. Three out of four are graph based constrained clustering algorithms, namely flexible constraint spectral clustering, spectral clustering with local proximity structure and constraint spectral clustering with integration of constraints in optimization criterion [78, 81,86]. Fourth one is a well known K-means constrained clustering (COP-KMEANS) [80]. These five constraint clustering algorithms are compared according to the external quality criterion namely **Rand index**. The empirical outcome on synthetic and real world datasets is that CC competes with other constrained clustering algorithm variants under certain conditions.

1.3.4 Scalability of CC Through Variable Reduction

The MAXAGREE formulation of CC was not *applied* in practice to obtain clusters. One reason for this is the computational time involved in obtaining partitions as CC formulation involves solving an expensive SDP. SDP formulations are expensive to solve when large number of data points (variables) are involved. The computational complexity to solve the SDP formulation is given by $O(n^{4.5})$ [77] where n is the number of variables involved in the SDP formulation (or number of vertices in G or number of data point in the dataset).

In order to apply CC to large scale datasets, speeding of the SDP formulation is the key. Two well known scalable formulations for SDP, namely variable reduction [13] and constraint reduction [23] are examined. Burer *et al.* [13] proposed a variable reduction method to obtain approximate solution to the original SDP formulation through low rank Broyden Fletcher Goldfarb Shanno (BFGS) method [70]. Their method has been applied to MAX CUT problem and reported significant gains in the computational time and memory requirements. This work extends the above variable reduction method to CC for

scalability as next contribution. Key points of this extension are:

1. Address scalability of CC.
2. Apply CC in practice for **large datasets**[1].
3. Analyze CC's external quality on variety of datasets including pure graph datasets obtained from benchmark graph dataset repository `http://staffweb.cms.gre.ac.uk/~wc06/partition/`.
4. As CC is one form of constrained clustering method, CC cluster's quality is compared with that of constrained spectral clustering method [78], which takes into account pairwise data point constraints in obtaining clusters.
5. From external clustering quality point of view, scalable CC is experimentally observed to be a competitive method having better quality compared to original CC formulation and constrained spectral clustering method.

1.3.5 Scalability of CC Through Constraint Reduction

Instead of reducing number of variables involved in the SDP formulation, number of constraints are reduced to addresses the scalability problem of CC. In this direction, a scalable SDP formulation using constrained reduction method is proposed in this thesis. The proposed formulation is solved efficiently using SDPNAL tool. The proposed scalable formulation is compared with other scalable variants namely variable reduction based CC. Experimental results on synthetic, real world datasets whose graph sizes range from 100 vertices to 13000 vertices are tested with both the scalable formulations. Large scale benchmark graph datasets are also tested whose sizes range from 2395 vertices to 13992 vertices. The proposed formulation is shown to have an edge over the original CC formulation, variable reduction variant of CC and a constraint clustering method, namely constrained spectral clustering.

1.4 Organization of the Thesis

This dissertation is organized as follows:

1. **Chapter 2**: Presents a survey of CC methods.

[1]large number of nodes as well as edges

2. **Chapter 3:** Influence of dataset characteristics on quality of obtained clusters when different rounding techniques are employed is studied in this chapter.

3. **Chapter 4:** Two optimal graph construction techniques, namely `fitgraph` and complete graph based on Fisher information distance metric are discussed in this chapter. `fitgraph` method is applied on vector data to construct general graphs. Fisher information based distance metric is employed for constructing complete graphs on vector data. The optimal graph construction role on quality of the obtained clusters is examined in this chapter.

4. **Chapter 5:** A comparative study of constrained clustering methods is undertaken in this chapter. In particular, performance of four different constraint clustering algorithms, namely `COP-KMEANS, cSC and its variants` are compared with CC.

5. **Chapter 6:** This chapter presents a scalable CC formulation by reducing the number of variables involved in MAXAGREE SDP formulation. Reduction in variables is achieved by changing the structure of SDP formulation.

6. **Chapter 7:** A constraint reduction method is presented in this chapter to address the scalability of CC. Role of reducing the number of constraints instead of number of variables is examined.

7. **Chapter 8:** This chapter summarizes the overall contributions and discusses future research directions on CC for practical applications.

Chapter 2

Literature Survey and Background

2.1 Introduction

Discovering *natural groups* in a given dataset is the central theme of clustering. The discovered natural groups should be non-empty, disjoint and union of all the individual groups should account for the given dataset [87]. Data clustering is widely applied in diverse application domains. It is employed to gain understanding about the underline structure of the given data. Clustering is also employed to compress the data and it is used for speeding the training time of many supervised learning methods [57]. A comprehensive survey on various data clustering methods is performed by Anil Jain *et al.* [34, 35], Rui Xu *et al.* [87], Satu Elisa Schaeffer [69], Maria *et al.* [56] and Maurizio *et al.* [28].

This chapter focuses on a particular clustering method, namely CC, which uses similarity/dissimilarity information *alone*, unlike data clustering methods, to perform clustering. Labeling between every pair of elements is specified as '+' or '-'. This is the only information available qualitatively for clustering. Given such pair-wise relations, the objective is to place *similar* data points in the same clusters and *dissimilar* data points into different clusters. This objective is trivially met by obtaining connected components in a graph of similar pairs [84]. However, when errors in the relationship between pairs of points are introduced, the problem of finding clusters that differs from true clusters on minimum number of pairs is non-trivial. Bansal *et al.* noted the number of triangles within which two edges having '+' label and one edge with '-' label forms the lower bound on number of disagreements [6].

This problem is posed as a graph theoretic problem. Let $G = (V_g, E)$ be a graph with V_g vertices and E edges. Every edge is labeled either '+' (similar) or '-' (dissimilar). In CC, the objective is to find a *clustering* of the vertices of G based on the edge label

information. Two building blocks are used for setting the objective function. These are:

i. **Agreement**: An edge is in agreement with given clusters if a '+' labeled edge lie *within* a cluster OR a '-' labeled edge lie *across* clusters.

ii. **Disagreement:** An edge is in disagreement with a given clusters if a '+' labeled edge lie *across* clusters OR a '-' labeled edge lie *within* a cluster.

Three widely referred variants of CC are:

1. **MAXAGREE:** The objective is to maximize the agreement of edge labels within a cluster and maximize the disagreement of edge labels across clusters.
2. **MINDISAGREE:** The objective is to minimize the disagreement of edge labels within clusters and minimize the agreement of edge labels in between the clusters.
3. **MAXCORR:** The objective here is to maximize the difference between number of agreements and number of disagreements.

CC is first proposed by Ben-Dor *et al.* for analyzing gene expression data [9] with only available information relation between genes i.e. two genes are related or not related. The existence of true partitions of genes is assumed. Similarity between *measured* gene expression patterns are obtained. As the measurements (of gene expressions) introduce errors in similarity computations they are (errors) *modeled* by assuming that the similarity computation is a result of modifying the true clusters. A stochastic error model has been proposed which recovers true partitions with high probability using the available similarity information.

Bansal *et al.* independently attempted the CC problem. In particular *approximate solutions* to MAXAGREE and MINDISAGREE are proposed for **complete graphs**. Key contributions of their work are [6]:

1. MINDISAGREE or MAXAGREE objectives are proved to be NP-complete.
2. Proposed a constant factor approximation[1] for MINDISAGREE objective and a polynomial time approximation scheme (PTAS) for MAXAGREE objective.

[1]

Definition 2.1. *An α-approximation algorithm for an optimization problem is a polynomial-time algorithm that for all instances of the problem produces a solution whose value is within a factor of α of the value of an optimal solution [83].*

Charikar *et al.* have given results pertaining to **general graphs** for MINDISAGREE and MAXAGREE objectives of CC [16]. Specifically:

i. A factor $O(\log n)$ approximation algorithm for MINDISAGREE is presented. A relaxed version of linear programming (LP) formulation is proposed for MINDISAGREE as given below:

$$\begin{aligned} \text{minimize.} \quad & \sum_{+(i,j)} w_{ij} \times x_{ij} + \sum_{-(i,j)} w_{ij} \times (1 - x_{ij}) \\ \text{subject to.} \quad & x_{ik} \leq x_{ij} + x_{jk} \; \forall \; i, j, k \\ & x_{ij} \in \{0, 1\} \; \forall \; i, j; \end{aligned} \tag{2.1}$$

In (2.1), x_{ij} is an indicator variable which takes value 0 when vertex i and j are in same cluster and 1 otherwise. w_{ij} is weight specified on every edge on the graph. First term of (2.1) specify to minimize all positively labeled edges which are placed in different clusters. In this situation the indicator variable x_{ij} assumes value 1. Similarly second term of the (2.1) specify to minimize all negatively labeled edges which are placed in same cluster. In this situation the indicator variable x_{ij} assumes value 0. The objective of the LP formulation is to minimize all such disagreements.

ii. For the MAXAGREE objective, a variant of the LP formulation given in (2.1) is obtained as given in (2.2).

$$\begin{aligned} \text{maximize.} \quad & \sum_{+(i,j)} w_{ij} \times (1 - x_{ij}) + \sum_{-(i,j)} w_{ij} \times x_{ij} \\ \text{subject to.} \quad & x_{ik} \leq x_{ij} + x_{jk} \; \forall \; i, j, k \\ & x_{ij} \in \{0, 1\} \; \forall \; i, j; \end{aligned} \tag{2.2}$$

The LP formulation given in (2.2) is argued to have a poor *integrality gap*[2]. To alleviate the above difficulty, the integer programming formulation in (2.2) is first

[2]The optimization given in equation (2.2) is an integer linear programming problem. By relaxing the constraint $x_{ij} \in \{0, 1\}$ to $x_{ij} \in \mathbb{R}$ and reformulating the problem to yield a convex optimization formulation. Let the true solution to (2.2) be denoted by OPT. Let the solution resulting from the constraint relaxation be FRAC. Let the solution obtained by subjecting the FRAC to rounding procedure be ROUND. Relation between ROUND, OPT and FRAC is expressed as ROUND $\leq$ OPT $\leq$ FRAC. The quality of the obtained solution through relaxed convex formulation is measured by the fraction $\frac{\text{FRAC}}{\text{OPT}}$. Integrality gap is the supremum of this ratio overall instances of the problem [19].

relaxed to a semi-definite programming (SDP) formulation[3]. Let $\mathbf{v}_i$ be a unit vector. Let the inner product between two vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ denoted as $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$ be 1 when these two vectors are in the same cluster. Otherwise inner product between them assumes a value 0 (that is $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = 0$). The SDP formulation with this setting is given below:

$$\begin{aligned} \text{maximize.} \quad & \sum_{+(i,j)} w_{ij} \langle \mathbf{v}_i, \mathbf{v}_j \rangle + \sum_{-(i,j)} w_{ij} \left(1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle\right) \\ \text{subject to.} \quad & \langle \mathbf{v}_i, \mathbf{v}_i \rangle = 1 \; \forall \;\; i \\ & \langle \mathbf{v}_i, \mathbf{v}_j \rangle \geq 0 \; \forall \;\; i, j \end{aligned} \tag{2.4}$$

Solution obtained by solving (2.4) is then subject to rounding procedure to obtain integer solution. The rounded solution is shown to have an approximation value of 0.7664.

In similar lines to [16], Chaitanya Swamy proposed a 0.7666 approximation algorithm using *any* graph for MAXAGREE objective in which weights on edges may take negative values [15]. For obtaining hard cluster solution from the solution obtained solving (2.4), hyperplane rounding technique [32] is employed. Multiple hyperplanes are used for producing clusters when there are more than 2 clusters. Dotan *et al.* presented a $O(\log n)$ approximation algorithm for MINDISAGREE objective for problems involving unweighted general graphs and weighted general graphs [24].

To improve the approximation value, a two cluster CC formulation for the MAXAGREE objective function, denoted as MAXAGREE2, has been proposed in [31]. The authors have shown a very good approximation value of 0.87856 for the two cluster CC formulation. The optimization formulation for the same is as follows:

[3]

Definition 2.2. *A semi-definite programming is an optimization formulation involving a linear objective function which is subject to linear constraints and a positive semi-definite constraint.*

Standard primal formulation for the SDP is written as follows:

$$\begin{aligned} \text{minimize.} \quad & \text{trace}(CX) \\ \text{subject to.} \quad & \text{trace}(A_i X) = b_i \; \forall \; i \\ & X \succeq 0 \end{aligned} \tag{2.3}$$

where $A_i \in \mathbb{R}^{n \times n}$ and $b_i \in \mathbb{R}^n$.

$$\begin{aligned} \max \quad & \frac{1}{2}\left(\sum_{-(i,j)} w_{ij}(1-\langle \mathbf{v}_i, \mathbf{v}_j\rangle) + \sum_{+(i,j)} w_{ij}(1+\langle \mathbf{v}_i, \mathbf{v}_j\rangle)\right) \\ \text{such that} \quad & \langle \mathbf{v}_i, \mathbf{v}_i\rangle = 1 \ \ \forall\, i \\ & \langle \mathbf{v}_i, \mathbf{v}_j\rangle \geq 0 \ \ \forall\, i,j \end{aligned} \tag{2.5}$$

In equation (2.5) first term represent sum of all the weights of the edges that lie across clusters. Second term represent sum of all the weights of the edges that lie within a cluster. As $\mathbf{v}_i$'s are integers, introducing relaxations as proposed in the seminal work [32], yields a semi-definite programming (SDP) formulation for CC and is given as:

$$\begin{aligned} \max \quad & \text{trace(C V)} \\ \text{such that} \quad & diag(V) = \mathbf{e} \\ & V \succeq 0 \end{aligned} \tag{2.6}$$

where $C = \frac{1}{4}\left((Diag(W_{out}\mathbf{e}) - W_{out}) + (Diag(W_{in}\mathbf{e}) + W_{in})\right)$. W_{in} denotes the weight of edges which lie inside partition and W_{out} denotes weight of edges which lie between any pair of partitions. V is a positive semidefinite matrix. $\mathbf{e}$ is the vector of all ones. $diag(V)$ denotes diagonal of matrix V and $Diag(\mathbf{v})$ denotes a diagonal matrix whose diagonal elements are the elements of vector $\mathbf{v}$.

Henceforth equation (2.6) is referred as SDP-CC. Solving SDP-CC yields optimal vectors $\mathbf{v}_i^*$ which stand for vertices belonging to distinct partitions. However SDP-CC yields soft clusters that is $\mathbf{v}_i^* \in \mathbb{R}^n$, where $n = |V_g|$. To obtain hard cluster solution, the hyperplane rounding technique is employed as proposed in [32].

CC for **bipartite graphs** is proposed by [2]. Claire Mathieu *et al.* [55] studied the **online CC** method where data items arrive in a regular intervals. Error bound for CC is proposed by [37]. Using the error bounds, they have analyzed how the accuracy of CC scales with the number of clusters and sparsity of the graph.

2.2 Application of CC

Correlation clustering found applications in entity deduplication [4], signed social network graphs such as world wide web for spam detection, product recommendation, online social networks for community detection given the information about friendship (+) and enmity (-) labels [17], link classification or edge label classification [14]. Arasu *et al.* [4] proposed a declarative language which captures constraints beyond the well known must link and

cannot link, and an efficient algorithm to identify duplicates. In this a graph in which nodes represent an entity reference and edges represent associated constraints (entities positively related (soft/hard +)/negatively related (soft/hard -)) are taken as inputs to obtain clusters. Clusters are obtained by minimizing the disagreement cost in the presence of constraints. A variant of CC is proposed to obtain clusters in the presence of relation between edges that go beyond + and - as proposed by Bansal *et al.* Nicolö *et al.* studied the problem of link classification (predicting an edge label to be + or -) in a theoretical frame work [14]. In particular, complexity of edge labeling is characterized through correlation clustering index. Recently Flavio *et al.* proposed a scalable correlation clustering using MINDISAGREE formulation in the MapReduce frame work [18] and has shown results on large scale real world graph datasets such as twitter dataset.

In [44] author proposed a modified version of the correlation clustering for image segmentation. Correlation clustering is applied on hyper graph of an image. Hyper graph is constructed by defining binary label between pair of nodes (super pixels). The binary label takes value 1 when the node pair belong to same region and 0 otherwise. A discriminant function over the given image and edge labels is defined as sum of similarity function between node pair. The objective function in turn is expressed as inner product of a weight vector and a feature mapping $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$. Objective is to obtaining an optimal labeling which maximizes the value of the discriminant function.

Entity searching is increasingly used by web surfers. Web people search centrally consists of searching for persons through web search engines. To effectively search for people on the web, search results together with clustering of similar persons is proposed by [60]. The authors employee correlation clustering for obtaining clusters from the obtained web search results. Effectiveness of correlation clustering has been demonstrated. However, it is not clear which optimization formulation (MAXAGREE/MINDISAGREE) was employed in order to obtain clustering solution.

Coleman *et al.* proposed a 2-approximation local search algorithm for the two cluster correlation clustering in case of complete graph [20]. Authors empirically established the performance of the proposed local search algorithm in case of general graphs on regulatory network human epidermal growth factor (EGFR) and two synthetic datasets.

A variant of CC which is known as labeled CC is studied by [50]. Input to the labeled

CC is an instance of CC. Author studied the computational complexity of LCC and shown that LCC is NP-complete. It has been shown a O($\sqrt{nlogk}$) approximation algorithm for general general LCC by applying L-reduction to the minimum Labelled Multicut problem.

CC is applied in the cross lingual link detection problem [76]. Cross lingual link detection is the problem of identifying news articles in multiple languages that report the same news event. In [76], the authors have proposed a linear programming chunking to process large dataset in case of MINDISAGREE objective. Linear programming chunking is a technique to divide a large linear program into smaller sub problems iteratively [12]. VanGael *et al.* has shown the edge of their method over the hierarchical clustering approaches commonly used in link detection problem. Zhang *et al.* have proposed a CC based on genetic algorithm for documents clustering [89]. Experimentally the performance of genetic algorithm is shown to be better than that of the performance of other clustering techniques.

2.3 Summary

This chapter briefed various attempts in the literature on the CC method. The existing literature has focused on the theoretical improvements to various problem settings in CC. The MAXAGREE objective of CC has been analyzed thoroughly from the approximation value perspective (in turn internal quality) by employing variants of hyperplane rounding technique, namely use of more than one hyperplane for rounding [15] to obtain multi cluster solution.

In the literature, different techniques are proposed for rounding the obtained SDP solutions [46]. The rounding technique variants are studied on MAX CUT, MAX 2SAT, MAX DICUT, MAX BISECTION *etc.* Next chapter examines the merit of other rounding techniques in the context of CC, namely **outward rotation** as well as **random projection and randomized rounding** (RPR^2) techniques which are argued to be superior to hyperplane rounding technique [27, 92].

Chapter 3

Role of Rounding Techniques

Solving the SDP-CC formulation for clustering yields soft cluster solutions. In particular, $\mathbf{v}_i^*$'s are obtained by solving equation (2.6). Note that $\mathbf{v}_i^* \in \mathbb{R}^n$. To obtain hard cluster solution, $\mathbf{v}_i^*$'s are subjected to rounding technique. One way of obtaining hard clusters is to employ inexpensive clustering method such as K-means clustering to group $\mathbf{v}_i^*$ [29]. This method is extensively used in well known graph clustering methods such as spectral clustering [51]. Three well known rounding techniques employed for rounding the SDP-CC solutions are:

1. Hyperplane rounding [32].
2. Outwards rotation rounding [92].
3. Random projection and randomized rounding (RPR^2) [27].

Outward rotation and RPR^2 are theoretically argued to be better than hyperplane rounding technique on various problem instances [27].

This chapter study the implication of applying various rounding techniques in the context of CC. In all the CC variants − that is CC on complete graphs, general graphs, two-cluster formulation − only approximation value is reported in the literature. However, the quality of the obtained clusters when the data characteristics are varied for a given rounding technique is not examined closely. This chapter empirically studies the impact of dataset characteristics on the quality of CC when various rounding techniques are employed. To the best of our knowledge this is the first time the implication of outward rotation and RPR^2 rounding techniques in the context of CC is studied and the quality of the obtained clusters is analyzed. Outward rotation and RPR^2 rounding techniques are experimentally observed to be sensitive to the data characteristics. Contradictions from

the literature are observed that outward rotation and RPR^2 are inferior when compared to hyperplane rounding technique under the influence of varying data characteristics.

3.1 Rounding Techniques

Three different rounding technique are studied for rounding the obtained soft cluster solution, namely: hyperplane rounding, outward rotation rounding and RPR^2 rounding techniques. These techniques are detailed below:

3.1.1 Hyperplane Rounding

Goemans and Williamson *et al.* proposed the hyperplane rounding technique for rounding the solution obtained by solving SDP formulation (in particular to MAX CUT problem) [32]. The hyperplane rounding technique separate the solution vectors into two distinct sets as shown in the Figure 3.1. Let $\mathbf{v}_1^*, \mathbf{v}_2^*, \cdots, \mathbf{v}_n^*$ be the solution obtained after solving MAX CUT SDP formulation. Chose a random vector $\mathbf{r}$ from a uniformly distributed unit sphere. Let $\langle \mathbf{v}_i^*, \mathbf{r} \rangle$ represent the inner product between vectors $\mathbf{v}_i^*$ and $\mathbf{r}$. Vector $\mathbf{v}_i^*$ is placed in a set when this inner product exceeds a specified threshold value (say 0). Let S be the set of vectors defined based on specified threshold as follows: $S = \{\mathbf{v}_i^* \mid \langle \mathbf{v}_i^*, \mathbf{r} \rangle \geq 0\}$; When the random hyperplane rounding technique is applied to SDP-CC formulation, two sets are obtained namely S_+ and S_- defined as follows $S_+ = \{\mathbf{v}_i^* \mid \langle \mathbf{v}_i^*, \mathbf{r} \rangle \geq 0\}$ and $S_- = \{\mathbf{v}_i^* \mid \langle \mathbf{v}_i^*, \mathbf{r} \rangle < 0\}$. S_+ and S_- stand for the clusters obtained through CC.

In the Figure 3.1 nine vectors are represented by $\mathbf{v}_1$ to $\mathbf{v}_9$. A random hyperplane is drawn from a unit sphere. All the vectors that lie above the hyperplane, namely $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ and $\mathbf{v}_4$ are placed in one cluster. Rest of the vectors are placed in a different cluster.

Applications

Hyperplane rounding is applied in MAX K-CUT, MAX 3SAT, MAX BISECTION, MAX 2SAT and MAX DICUT optimization problems [25, 26, 30, 41, 42] to improve the approximation value of respective problems. In the case of two cluster CC by applying this rounding technique, the approximation value is improved to 0.87856 [31]. In [15] Chaitanya Swamy applied the extended hyperplane rounding technique to CC by selecting more than one hyperplane passing through the origin to get a better approximation value of 0.7666. Various problem instances where success of hyperplane rounding technique was reported is presented in Table 3.1.

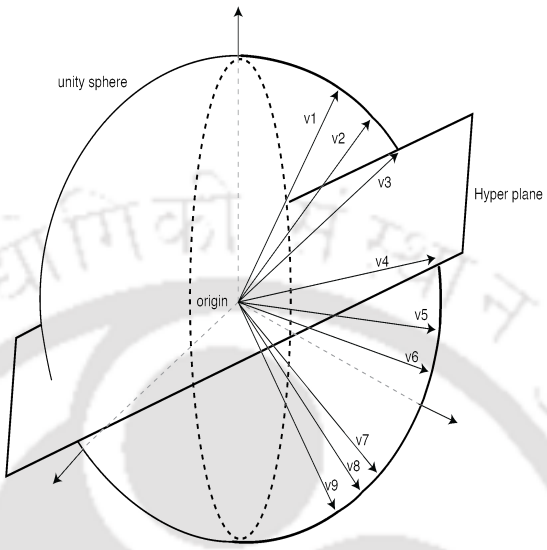

Figure 3.1: Hyperplane Rounding

Table 3.1: Hyperplane Rounding Result

Problem	Approximation Value (Other than Hyperplane Rounding Technique)	Approximation Value in the case of Hyperplane Rounding Technique
MAX CUT	0.5 [68]	0.87856 [32]
MAX SAT	0.5 [38]	0.758 [32]
MAX 2SAT	0.618 [48]	0.878 [32]
MAX BISECTION	0.5 [68]	0.6514 [30]
MAX DICUT	0.25 [61]	0.790 [32]

3.1.2 Outward Rotation Rounding

Outward rotation rounding is a generalization of the hyperplane rounding technique. This technique is proposed by Zwick [92] to enhance the performance of various optimization problems such as MAX CUT. In case of MAX CUT, the hyperplane rounding technique fares better when cut sizes are very large (more than 85% of edges are in the cut). Such large cuts do not occur in practice. For graphs whose cut sizes are smaller (less than 85% of edges are in the cut), the approximation value obtained from hyperplane rounding is close to 0.87856. Outward rotation rounding improves this approximation value for graphs whose cut sizes are small. Outward rotation rounding technique is a simple rearrangement of the solution vectors $\mathbf{v}_i^*$ before applying the hyperplane rounding technique. Rearrangement of the solution vectors is achieved using a combination of *independent random choice* along with random hyperplane rounding.

These vectors are first embedded into a higher dimensional space $\mathbf{v}_i^* \in \mathbb{R}^{2n}$ (and hence the name *outward*). This is achieved by simply adding remaining n dimensions with 0's to every $\mathbf{v}_i^*$. Let $\left\{\mathbf{e}_i \in \mathbb{R}^{2n}\right\}_{i=1}^{n}$ be a set of orthonormal vectors which are also orthogonal to $\mathbf{v}_i^*$'s. $\mathbf{v}_i^*$'s are *rotated* by an angle γ ($0 \leq \gamma \leq \frac{\pi}{2}$) towards $\mathbf{e}_i$ (and therefore the name *outward rotation*). Let rotated vectors be denoted by $\{\mathbf{v}_i'\}_{i=1}^{n}$. The angle between vectors $\mathbf{v}_i^*$ and $\mathbf{v}_j^*$ be θ_{ij}. The angle between rotated vectors be θ_{ij}'. Then relation between these two angles as shown in [92] is:

$$cos(\theta_{ij}') = cos^2(\gamma) \times cos(\theta_{ij}) \tag{3.1}$$

The vectors $\{\mathbf{v}_i'\}_{i=1}^{n}$ are obtained in such a way that the above equation is satisfied. That is $\langle \mathbf{v}_i', \mathbf{v}_j' \rangle = cos^2(\gamma)\langle \mathbf{v}_i^*, \mathbf{v}_j^* \rangle$. Adopting the notations given in [92], the complete algorithm is as given below:

W_{total}	$= \sum_{+(i,j)} w_{ij} + \sum_{-(i,j)} w_{ij}$	Total weight of the edges of the graph
Z	$= \sum_{-(i,j)} w_{ij} \frac{1-\langle \mathbf{v}_i, \mathbf{v}_j \rangle}{2} + \sum_{+(i,j)} w_{ij} \frac{1+\langle \mathbf{v}_i, \mathbf{v}_j \rangle}{2}$	Objective function value of CC
A_{ratio}	$= \frac{Z}{W_{\text{total}}}$	Ratio of Z to total weight of the edges of the graph

1. Solve the SDP-CC formulation given in equation (2.6) to obtain solution vectors $\mathbf{v}_1^*, \mathbf{v}_2^*, \cdots, \mathbf{v}_n^*$. Using these vectors obtain A_{ratio}.

2. If $A_{\text{ratio}} \geq 0.84458$ (when cut size is greater than 84.458%), $\gamma = 0$.

Table 3.2: Outward Rotation Rounding Result

Problem	Approximation Value (Other than Outward Rotation Rounding Technique)	Approximation Value in the Case of Outward Rotation Rounding Technique
MAX CUT	0.8785 [32]	0.9119 [92]
MAX SAT	0.758 [32]	0.797 [92]
MAX BISECTION	0.6514 [30]	0.7016 [33]
MAX DICUT	0.790 [32]	0.874 [47]
MAX 2SAT	0.878 [32]	0.940 [47]

3. If $A_{\text{ratio}} < 0.84458$ (when cut size is less than 84.458%) then $\gamma = \arccos(\sqrt{c})$ where c is obtained by solving the following two equations according to the lemma 1 of [92]:

$$\frac{\arccos(c \times (1-2t)) - \arccos(c)}{t} = \frac{2c}{\sqrt{1-c^2 \times (1-2t)^2}}$$

and

$$\frac{1-\frac{t}{A}}{\sqrt{1-c^2}} = \frac{(1-2t)}{\sqrt{1-c^2 \times (1-2t)^2}}$$

4. Obtain the *outward rotated vectors* $\mathbf{v}'_1, \mathbf{v}'_2, \cdots, \mathbf{v}'_n$ by rotating the vectors $\mathbf{v}^*_1, \mathbf{v}^*_2, \cdots, \mathbf{v}^*_n$ by an angle γ. The outward rotated vectors satisfies equation (3.1).

5. Subject $\{\mathbf{v}'_i\}_{i=1}^n$ to hyperplane rounding technique to obtain hard cluster solution.

Zwick [92] applied outward rotation rounding technique for MAX NAE-3-SAT to improve approximation value from 0.87856 to 0.90871. Xu *et al.* [85] applied Outward rotation rounding technique for $MAX\frac{n}{2}UNCUT$ problem, to obtain a better approximation value than the existing 0.6436 value. The problem instances where outward rounding technique employed successfully is presented in Table 3.2.

3.1.3 Random Projection Randomized Rounding

RPR^2 rounding technique [27] is a family of rounding procedures namely random projection and randomized rounding defined by a function $f : \mathbb{R} \to [0, 1]$. RPR^2 rounding has the following two steps:

Table 3.3: RPR^2 Rounding Result

Problem	Approximation Value (Other than RPR^2 Rounding Technique)	Approximation Value in the Case of RPR^2 Rounding Technique
MAX CUT	0.9119 [92]	0.9128 [27]
MAX BISECTION	0.7016 [33]	0.7027 [27]

1. **Random Projection:** A random vector $\mathbf{r}$ is chosen which follow multivariate Gaussian distribution. $\mathbf{v}_i^*$'s are projected onto the random vector $\mathbf{r}$. Let the resulting vector be $\mathbf{v}_i$.

2. **Randomized rounding:** The vector $\mathbf{v}_i$ is rounded as follows: Let $x_i = \langle \mathbf{v}_i, r \rangle$. x_i is set to 1 with a probability $f(x_i)$; 0 otherwise, where the function $f(.)$ is a linear function. In the present experimentation $f(x) = \frac{1}{2} + \frac{x}{2s}$ where $s = 0.263$; this is adopted from [27].

Random hyperplane rounding and outward rotations are both special cases of RPR^2 rounding. The performance of rounding technique depends on the choice of the function f. Feige [27] has chosen a linear function to obtain a better approximation value of MAX CUT and MAX BISECTION problem. Various problem instances where success of RPR^2 rounding technique was reported is presented in Table 3.3.

In the literature, outward rotation rounding technique is shown to yield approximation values better than hyperplane rounding technique. RPR^2's superiority over outward rotation rounding technique is argued in [27]. Note that these theoretical results holds for SDP formulations for problem context other than correlation clustering as discussed in section 3.1.1 and 3.1.2 respectively. In this thesis these three rounding techniques are employed in the context of CC to study the implication of various rounding techniques on the quality of obtained clusters.

3.2 Empirical Study

This section presents experimental results to show the impact of various rounding techniques on the performance of SDP-CC. Specifically the aim is to show through experiments that: (1) the rounding technique performance depends on the characteristic of datasets; and (2) how internal quality (*Approximation Value*) differs from external quality

of SDP-CC with respect to different rounding techniques. An external quality measures the closeness of the obtained clusters, C_i, and true clusters, C_j. Through out this thesis one external quality measure, namely Rand Index [64] is employed. Rand Index is the ratio of the number of agreements between any two given partitions to the total number of possible data pairs.

When two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are placed in same partition by a clustering algorithm and their true labels match then there is an agreement between the obtained partition and their true labels. Let the number of such agreements be 'a'. When two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are placed in different partitions by the same clustering algorithm and their true labels dis-agree, then there is a match between obtained partitions and true labels. Let the number of such agreements on dis-agreement be 'b'. Rand Index is computed using the total agreements as: $RandIndex\,(\text{obtained clusters}, \text{true labels}) = \frac{a+b}{n*(n-1)/2}$.

Minimum value of rand index 0 is attained when obtained clusters do not have any pair of data in common with the true labels. Maximum value 1 is attained when the obtained clusters is same as the true labels. The higher the rand index, the better is the quality of the obtained clusters.

3.2.1 Datasets

Two types of datasets are considered, namely synthetic datasets in which the dataset characteristics are varied and real world datasets taken from UCI machine learning dataset repository `http://www.ics.uci.edu/~mlearn/`.

The `Pendigit` dataset is about the handwritings of people explaining 10 digits − 0 to 9. As two cluster formulation is considered for CC, data points belonging to two digits, namely 0 and 1 are considered. `Yeast` dataset contains 10 classes and each data point is of 8 dimensions. A total of 1484 training data points are present in this dataset. Two classes namely 'cyt' and 'nuc' are considered in the `Yeast` dataset. `Vehicle` dataset contains 4 classes with each data point described with 18 attributes and comprising of a total of 946 data points. 'Bus' and 'van' classes are considered in the `Vehicle` dataset. Table 3.4 show the characteristics of each of the dataset.

Synthetic dataset: Four kinds of synthetic datasets are generated and used for experimentation:

1. **Well Separated Clusters:** In this case, data points from Gaussian distributions having well separated means $(3, 5)$, $(10, 5)$ and identity matrix as covariance matrix are generated.

2. **Overlapped Clusters:** Data points from two Gaussian distributions, having $(3, 5)$, $(6, 5)$ as means and identity matrix as covariance matrix, are generated.

3. **Unbalanced Clusters:** Dataset in which the overlapping clusters have varying cluster sizes.

4. **Noise:** In addition to the above three, noise is added to each cluster. The additive noise is Gaussian noise and Uniform noise. In case of Gaussian-noise dataset, five percent of noise generated from Gaussian distribution is added to the overlapped clusters dataset to understand the sensitivity of SDP-CC to noise. In case of Uniform-noise dataset, five percent noise generated from Uniform distribution is added to the overlapped clusters dataset.

Each Gaussian stands for a cluster in these datasets. A total of 5000 data points are sampled from the two distributions. Each dataset consists of 25 realizations. Realization of a dataset means that every dataset is sampled from a particular distribution in question 25 times, each time sampling 5000 data points. For example in the case of well separated dataset, 25 sets of well separated data points each having 5000 instances are generated. Experimentation is performed on each of the realization. Average rand index value and its variation are presented while reporting the results.

In Table 3.4 the number of data points (nodes in the graph), dimension of each data point and number of classes the dataset contains are depicted in 2nd, 3rd and 4th columns respectively. Fifth column denotes imbalance in the dataset; that is for every positive data point, how many negative data points are sampled. Balanced datasets are those datasets having ratio 1:1 or close values (say 1:1.07). Rest are all unbalanced datasets.

Edge Labels: The key component of SDP-CC lie in obtaining edge labels (positively correlated or negatively correlated). Vertex of the graph stand for a data point and pair of vertices are correlated according to Euclidean distance measure. A complete graph is constructed using each of the dataset and an edge is labeled positive if the distance measure between the two vertices is less than a specified threshold (taken as mean of the affinity matrix throughout this thesis) and labeled negative otherwise.

The following are the key steps in proposed empirical study:

1. Every edge (between pair of points) is labeled positive, if the weight is less than or equal to a specified threshold otherwise negative.

2. Solve equation (2.6) to obtain soft clusters in the form of $\mathbf{v}_1^*, \mathbf{v}_2^*, \cdots, \mathbf{v}_n^*$.

Table 3.4: Synthetic and Real World Datasets

Dataset	data points	dim	classes	ratio
`Well separated`	5000	3	2	1 : 1
`Gaussian`	5000	3	2	1 : 1
`Unbalanced Gaussian`	5000	3	2	1 : 10
`Gaussian noise`	5000	3	2	1 : 1
`Unbalanced Gaussian noise`	5000	3	2	1 : 10
`Uniform noise`	5000	3	2	1 : 1
`Iris`	150	4	3	1 : 1
`Yeast`	1484	8	10	1 : 1.07
`Vehicle`	946	18	4	1.98 : 2.49
`Pendigits`	3498	16	10	1 : 1

3. Obtain hard clusters by applying various rounding techniques discussed in section 3.1.

4. Measure internal (A_{ratio}) and external quality of the obtained hard clusters.

SDPT3, which is designed to solve the conic programming problems, is used for solving SDP-CC formulation. This solver uses predictor-corrector primal-dual path following algorithm [73, 75].

3.2.2 Results

The obtained results are classified based on the data characteristics. Three groups are formed based on the considered datasets. These are:

1. **Balanced Datasets:** These datasets include `Gaussian`, `Well-separated Gaussian`, `Gaussian noise`, `Uniform noise`, `Iris`, `Yeast` and `Pendigits`.

   - **Internal Quality:** Figure 3.2 presents the obtained approximation value after rounding the obtained solution averaged over all the realizations. Dotted vertical line on each bar of the dataset represent variation in the approximation value. For all the balanced datasets, the hyperplane rounding technique's internal quality is observed to be better than that of outward rotation and RPR^2 rounding techniques. In the literature however, outward rotation and RPR^2 rounding techniques are theoretically shown to be better than hyperplane

rounding technique. Experimental result presented clearly contradicts the theoretical result.

- **External Quality:** Figure 3.3 presents the obtained external quality, namely Rand index on each of the dataset. From the external quality perspective, hyperplane rounding technique outperforms outward rotation and RPR^2 rounding techniques on majority of the datasets.

2. **Unbalanced Datasets:** These datasets include `Unbalanced Gaussian, Unbalanced Gaussian noise` and `Vehicle`.

   - **Internal Quality:** From the Figure 3.2, observing the above unbalanced datasets, note that hyperplane rounding technique outperforms outward rotation and RPR^2 rounding techniques. This result once again contradicts theoretical results. It is also observed that RPR^2 is inferior to outward rotation rounding technique where as in the literature RPR^2 is claimed to be a generalization of the outward rotation rounding technique [27].
   - **External Quality:** From the Figure 3.3 outward rotation rounding technique is observed to be performing better than hyperplane and RPR^2 rounding techniques.

3. **Noisy Datasets:** These datasets include `Gaussian noise, Unbalanced Gaussian noise` and `Uniform noise`. In this case, irrespective of the noise, both internal quality and external quality of CC depend on the ratio of the number of data points belonging to positive class and the number of data points belonging to negative class of a given dataset. Above discussed trends with respect to balanced dataset are observed to follow in case of noisy datasets.

3.3 Analysis

This section addresses the key question: `When Hyperplane rounding technique outperforms Outwards rotation rounding technique?` Let the first term, which involve edges in between the clusters, in the equation (2.5) be denoted by B and it is normalized with respect to the edge weights. Let the second term in the equation (2.5) be denoted by C, which involve edges within the cluster, and it is normalized with respect to the edge weights. As $\mathbf{v}_i$ and $\mathbf{v}_j$ are unit vectors, the angle between these two vectors (computed

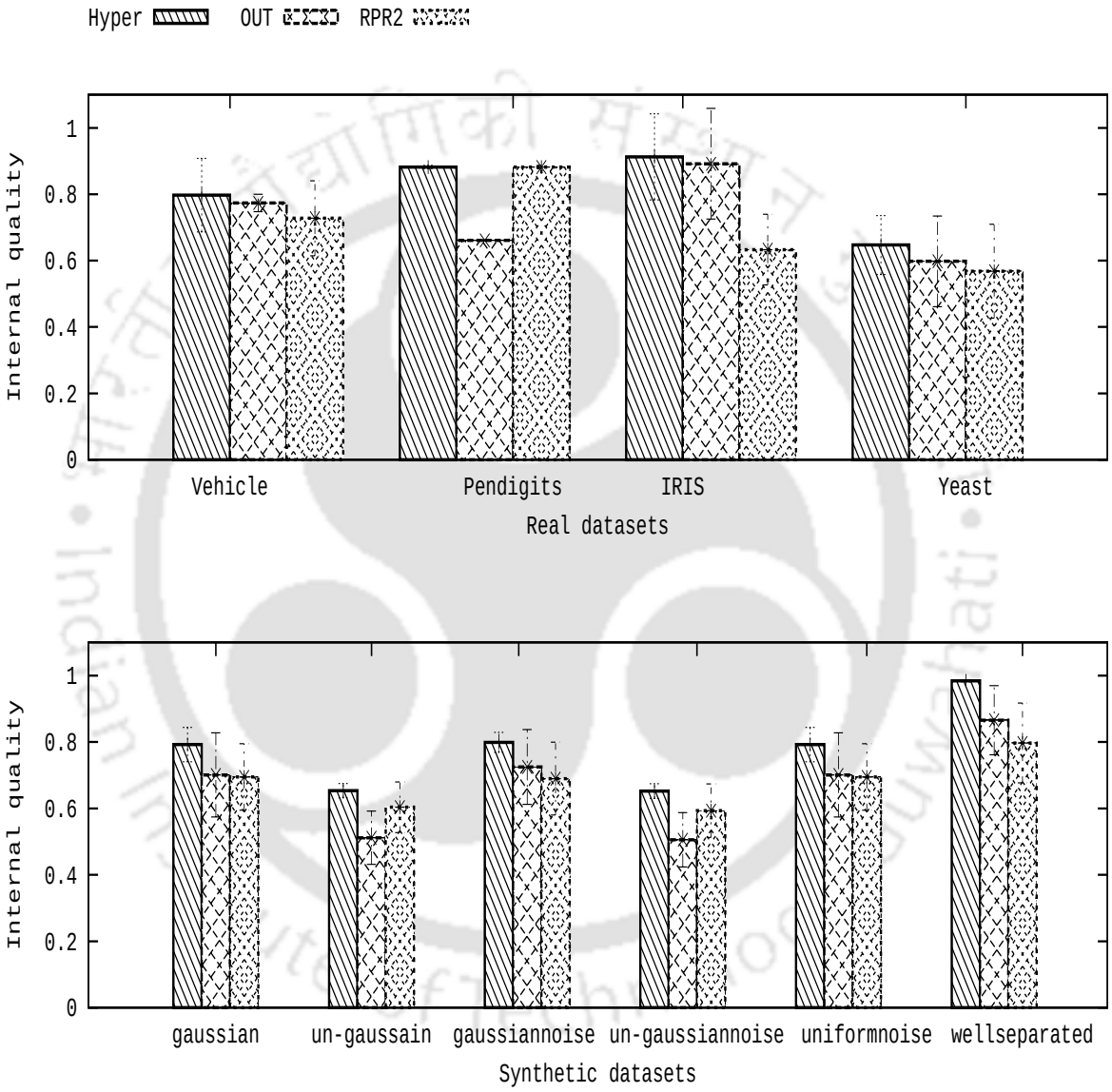

Figure 3.2: Internal Quality

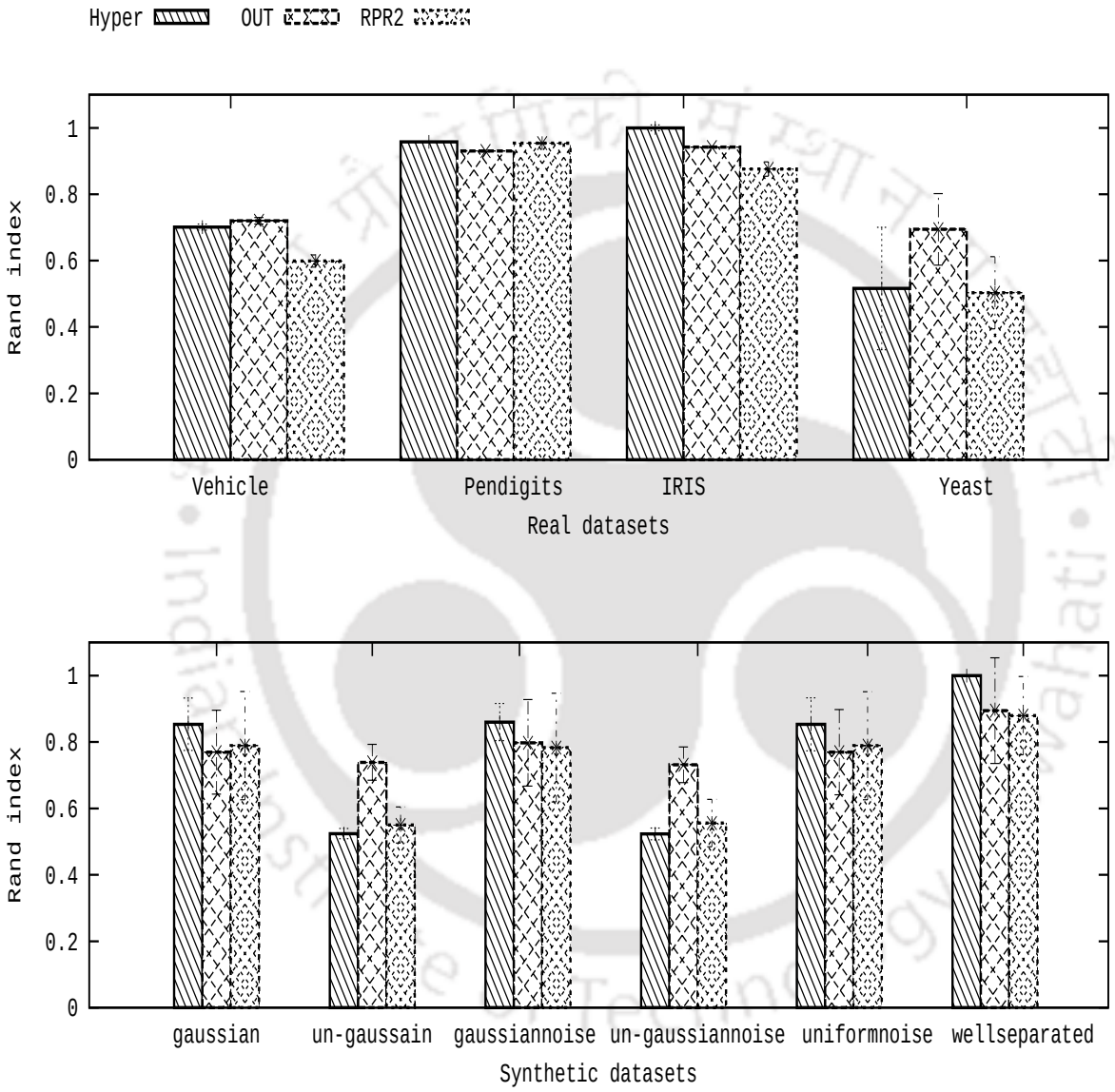

Figure 3.3: External Quality

using the dot product) for data points that lie in between clusters is $cos(\theta)$. Obtain $\theta_{\text{in between}}$ by solving $B = \frac{1}{2}(1 - cos(\theta_{inbetween}))$; therefore $\theta_{inbetween} = arccos(1 - 2B)$. In similar lines, the angle between vectors that lie within the clusters is obtained as $\theta_{within} = arccos(2C - 1)$. For all the considered datasets when $\theta_{inbetween} \geq 72^{\circ}$ and $\theta_{within} \leq 108^{\circ}$ hyperplane rounding technique is observed to outperform outward rotation technique.

3.4 CC with More Than Two Clusters

The above discussion on the empirical analysis of rounding techniques rely on the fact that CC obtains two clusters as discussed in chapter 2, equation (2.6). To extend the CC formulation for obtaining more than 2 clusters, Charikar *et al.* have proposed a 0.7664 approximation algorithm [16] by solving the SDP formulation of multi cluster for weighted undirected graphs given below:

$$\begin{aligned} \max \quad & \left(\sum_{-(i,j)} w_{ij}(1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) + \sum_{+(i,j)} w_{ij}(\langle \mathbf{v}_i, \mathbf{v}_j \rangle) \right) \\ \text{such that} \quad & \langle \mathbf{v}_i, \mathbf{v}_i \rangle = 1 \ \ \forall\, i = 1, 2, \cdots, |V_g| \\ & \langle \mathbf{v}_i, \mathbf{v}_j \rangle \geq 0 \ \ \forall\, i \neq j \end{aligned} \tag{3.2}$$

where w_{ij} is the weight on the edge $(\mathbf{v}_i, \mathbf{v}_j)$. $\mathbf{v}_i$ is a unit vector having dimension $|V|$. Solving this formulation and extending the hyperplane, outward rotation and RPR^2 rounding techniques to take into account more than one hyperplane provide insights into the effect of these rounding techniques on multiple clusters. Present work limit the empirical analysis to two cluster formulation.

3.5 Conclusion

In this chapter, external quality measure for CC is studied empirically with respect to three different rounding techniques, namely hyperplane, outward rotation and RPR^2 rounding technique. The relation between the internal quality and external quality is analyzed irrespective of the rounding techniques. It is observed that the outward rotation rounding technique is better in case of unbalanced datasets and hyper-plane rounding technique outperforms outward rotation and RPR^2 rounding technique in case of balanced datasets. These two empirical observations contradicts the theoretical results for outward rotation

and RPR^2 rounding techniques. This is due to the fact that the rounding technique analysis has not considered the dataset characteristics in theory.

Note that for employing CC one needs to have a graph before hand for obtaining clusters. In many machine learning applications however, graphs are not available directly. Instead, one has to obtain data points in the form of real vectors [54]. Well known graph clustering methods like spectral clustering [51,71] construct graph to obtain clusters. The graph construction introduces sensitivities in the final obtained cluster solutions. Next chapter deals with various graph construction methods and explores a recent trend of obtaining **optimal** graphs given vector dataset.

Chapter 4

Role of Optimal Graph Construction Methods

Graph as an abstraction of given dataset consisting of set of vectors has been widely used for well known task, namely clustering. In graph clustering, the objective is to obtain clusters consisting of vertices based on the edge similarity/dis-similarity. As in the case of data clustering, graph clustering is broadly divided into *hierarchical* and *partitional* [69] methods. In the hierarchical graph clustering the central idea is to start with as many clusters as the number of vertices. Iteratively merge smaller clusters into bigger ones which leads to optimizing a specified measure, say maximizing modularity, till desired number of clusters are reached [58]. In the partitional graph clustering technique the key idea is to obtain clusters by optimizing a specified objective. For example, in the case of spectral clustering, the objective is to optimize the `cut value` between two specified clusters say C_i and C_j. The cut value is defined as the sum of the weights of the edges that connect clusters C_i and C_j. The normalized cut value is optimized over every possible partitioning. In spectral clustering, eigenvectors of the normalized graph Laplacian matrix are used for clustering. Depending on the partition criteria, many variants of the spectral clustering technique are proposed [51, 56]. Normalized cuts [71] and Ng method [59] are popular among the spectral clustering methods.

In all the above mentioned graph clustering methods one of the main assumption is: availability of the graph. Effect of different graph construction methods on graph clustering namely spectral clustering is studied in [54]. The authors show the effect of K-NN and $\epsilon-$ neighborhood graph construction methods on the quality of clusters using spectral clustering. This spectral clustering involves two kinds of cut value (i) normalize

cut (Ncut) and (ii) Cheeger cut (CheegerCut). Following points are emphasized from their theoretical study.

1. Cluster quality and objective function value are sensitive to the employed cut criteria for *a given graph*.

2. Employing a particular cut criteria on *different graphs* lead to varying cluster quality and objective function values.

The convergence of cluster quality is established based on Ncut and CheegerCut as the sample size approaches to infinity. The conditions under which convergence take place for K-NN graphs and $\epsilon-$ neighborhood graphs are demonstrated.

Optimal Graphs: Recent development for constructing *optimal* graphs [1] using a given dataset are of particular importance as they do away with the choice of the free parameters (ϵ or k). Two ways of constructing optimal graphs are proposed in the literature. These are:

1. **Unsupervised Method:** In this method, class labels associated with the data points are not taken into account while constructing the optimal graph. Samuel *et al.* [21] have proposed a principled way of constructing *optimal sparse graphs* from a given vector data. Optimality of the constructed graphs is achieved through minimizing the sum of the distances between a data point and weighted sum of the data points' neighbors. Depending on the constraints, two variants of this optimization problem are posed, namely hard graph formulation and soft graph formulation. Samuel *et al.* conjecture that for a given set of vectors both the formulations are unique and have an average degree of at most $2 \times (q + 1)$ and $(q + 1) \times n$ edges; where q is the dimension of the dataset and n is the number of data points. The obtained optimal graphs are employed for classification, clustering and regression problems. Experimental results using optimal graphs have shown to outperform the traditional graph construction methods. Note that the graphs that result from Samuel's graph construction algorithm are weighted, general and undirected graphs.

2. **Supervised Method:** In this, class label associated with the data points are taken into account for constructing the optimal graph. Several supervised graph

[1] By optimal graph we mean a graph that is obtained by optimizing a specified objective function whose output is an affinity matrix which in turn is used as input to the correlation clustering algorithm.

construction methods are proposed in the literature, namely graph edge sharpening (GES) [72], spectral kernel learning (SKL) [91], marginal likelihood (ML) [40], supervised k-NN method [66] and Fisher Information distance metric based graph construction method [67].

Héctor *et al.* [67] proposed a framework for visualizing high dimensional labeled data in the form of similarity networks by employing Fisher information similarity measure. This weighs each dimension differently by taking into account class label information. The proposed method constructs a weighted complete similarity graph which is subjected to spectral clustering yielding better performance compared to traditional graph construction techniques from the given vector data. The graphs that are generated from Héctor's method are weighted, undirected and complete graphs.

Role of optimal graphs in the context of CC is of particular importance due to the following:

1. CC involves *only edge labels* unlike experiments performed using optimal graphs [21, 67] in which the distance measure between data points is also considered while performing clustering.

2. Approximation values for different types of graphs for CC were proposed in the literature and are detailed in section 4.1 assuming the availability of the graph. To attain theoretical approximation values, free parameters in the similarity based graph construction methods need to be tuned. The optimal graph construction methods provide a handle to achieve theoretical approximation value, that is obtaining optimal internal quality and associated external quality.

This is the *first attempt* to explore the **effectiveness of optimal graphs** in the context of CC. Instead of choosing a particular type of non-optimal graph (complete, general or bipartite), an optimal graph is constructed on vector datasets using the methods proposed in [21, 67]. The impact of *optimal* graph construction methods on quality of CC is compared with the quality obtained when CC uses non-optimal graphs. Convergence of the quality of CC is also studied with respect to optimal and traditional graph construction methods.

An empirical study is carried to understand the influence of optimal graph construction methods on the approximation value and in turn the quality of the obtained clusters. The convergence of CC's cluster quality is also examined when optimal graphs are employed. The following results are noted empirically from various non-optimal and optimal graph construction methods:

1. Approximation values obtained by employing optimal graph construction techniques have an edge over traditional graph construction methods.

2. Optimal graph construction methods obtain better quality clusters compared to non-optimal graph construction methods in CC.

3. Convergence of quality of CC is studied with respect to both optimal and non-optimal graph construction by varying the number of data points. Note that CC converges as the number of data points increase.

4. Time taken to solve CC using sparse optimal graphs obtained through `fitgraph` has an edge over CC based on complete graphs.

5. Time taken to construct the optimal graph using Fisher information method is less than that of optimal graph construction using `fitgraph`.

4.1 Similarity Graphs

To understand the effectiveness of CC, experiments are performed using both complete graphs and general graphs. Complete graphs are constructed using two methods (1) Well known similarity graphs and (2) Fishers information metric based graph construction [67]. General graphs are constructed using optimal graph construction technique [21] detailed in section 4.2.1.

For any graph clustering method, data should be in the form of a graph. When data points $\{\mathbf{x}_i\}_{i=1}^n$ to be clustered are available in the vector form, a similarity based graph is constructed [51] using the vector data. In similarity graphs, data points (that is each $\mathbf{x}_i$) stand for vertices of the graph (that is every $\mathbf{v}_i$) and similarity/dissimilarity measure between pair of data points stand for the edge relationship. Two vertices are connected if they are similar. The graph is represent by an affinity matrix W where rows and columns stand for vertices of the graph. An element in the matrix W, say w_{ij} represent the similarity between two vector data points $\mathbf{x}_i$ and $\mathbf{x}_j$. Well known similarity based graph constructions methods are:

1. ϵ-neighborhood graph: Vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected when similarity between them is less than or equal to the specified parameter ϵ (≥ 0).

2. K-nearest neighbor (K-NN) graphs: Vertex $\mathbf{x}_i$ is connected to vertex $\mathbf{x}_j$ if $\mathbf{x}_j$ is among K neighbors of $\mathbf{x}_i$. Resulting similarity graph will be a directed graph. An

undirected graph is obtained by explicitly having an edge between $\mathbf{x}_j$ and $\mathbf{x}_i$ when there is an edge between $\mathbf{x}_i$ and $\mathbf{x}_j$.

3. Mutual nearest neighbor graph: Vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected if $\mathbf{x}_i$ is among the K neighbors of $\mathbf{x}_j$ and vice-versa.

A similarity graph is then employed in graph clustering methods, *viz.,* spectral clustering [51]. Influence of similarity graphs on the spectral graph clustering algorithm is theoretically studied [53]. The quality of obtained clusters constructed through ϵ-nearest neighborhood method has been shown to be different from that of K-NN method. In this study ϵ-neighborhood graph is constructed for obtaining complete graphs.

4.2 Optimal Graph Construction

Two recent methods for constructing (a) weighted general graph and (b) weighted complete graph are discussed in this section. First method involves solving constrained quadratic optimization formulation to obtain weighted general graph and the second method emphasize the weighted similarity distance measure to construct a complete graph.

4.2.1 Optimal Weighted Undirected General Graph - `fitgraph`

Given the vector data points, $\{\mathbf{x}_i\}_{i=1}^n$, constructing a similarity graph that explains the data in best possible way is challenging. Daitch *et al.* [21] formulated the problem of obtaining *optimal* graph given the vector data. Optimality is defined as minimizing the sum of the distances between a data point $\mathbf{x}_i$ and weighted sum of $\mathbf{x}_i$'s neighbors by constraining the degree of each node. Intuition of the proposed objective is to express a given data point $\mathbf{x}_i$ in terms of its neighbors by obtaining weights appropriately. This problem is casted as a quadratic programming formulation.

Let each element w_{ij} in the weight matrix W denote weight between data points $\mathbf{x}_i$ and $\mathbf{x}_j$. Let $d_i = \sum_j w_{ij}$ denote the weighted degree of vertex i. $w_{ij} = 0$ if there is no edge between the vertex i and j; Self loops are not allowed in the optimal graph construction. The graph to be constructed is assumed to be symmetric ($w_{ij} = w_{ji}$). The following cost function is minimized to obtain optimal graph W.

$$f(W) = \min_W \sum_i \parallel d_i \mathbf{x}_i - \sum_j w_{ij} \mathbf{x}_j \parallel^2 \tag{4.1}$$

Based on the above objective, two optimal graph construction methods are proposed. These two formulations differ only in the constraints imposed to obtain optimal graphs. These are known as `hard graph` formulation and `soft graph` formulation. Minimizing the above objective function yields a graph that best explain the given vector data. To simplify the above function, let X be an $n \times q$ matrix with i^{th} row $\mathbf{x}_i \in \mathbb{R}^q$. The graph Laplacian matrix L is defined as:

$$L_{i,j} = \begin{cases} -w_{ij} & \text{if } i \neq j \\ d_i & \text{if } i = j \end{cases}$$

Introducing graph Laplacian in the objective function yields equivalent objective function as given below:

$$f(W) = \min_{W} \parallel LX \parallel_F^2 \tag{4.2}$$

where $\| \cdot \|_F$ is the Frobenius norm.

Hard Graph

When the vector data does not contain any noise, one minimizes the above objective function with a constraint that every vertex should at least have a degree greater than or equal to 1. The objective function is written in terms of a matrix M that encodes data points and edge information between pair of data points such that minimizing $\|LX\|_F$ amounts to minimizing the following function:

$$f(\mathbf{w}) = \min_{\mathbf{w}} \parallel M\mathbf{w} \parallel^{\mathbf{2}} \tag{4.3}$$

subject to the constraint that degree of every vertex is greater than or equal to 1 where $\mathbf{w}$ denotes a vector containing all the edge weights (whose length is $|E|$). Re-formulating this in terms of quadratic optimization form to obtain the hard graph formulation is given as:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \parallel M\mathbf{w} \parallel^{\mathbf{2}} \\ \text{subject to} \quad & d_i \geq 1 \end{aligned} \tag{4.4}$$

Soft Graph

To deal with datasets having outliers, the degree constraint for each vertex in the hard graph is relaxed by bounding the total degree. Sum of the degrees of the vertices is

bounded by a factor of α times the total number of vertices. That is to introduce the following constraint in the hard graph formulation:

$$\sum_i (max(0,(1-d_i)))^2 \leq \alpha \times n$$

As the scaled values of $\mathbf{w}$ leads to improved function values $f(\mathbf{w})$, the above inequality is converted to an equality constraint.

$$\begin{array}{ll} \underset{\mathbf{w}}{min} & \parallel M\mathbf{w} \parallel^{\mathbf{2}} \\ \text{subject to} & \sum_i (max(0,(1-d_i)))^2 = \alpha \times n \end{array} \tag{4.5}$$

The above constraint is eliminated by employing Lagrangian to obtain an unconstrained optimization formulation as given below:

$$\underset{\mathbf{w},\mathbf{s}}{min} \quad \|M\mathbf{w} \|^2 + \mu \|\mathbf{1} - A\mathbf{w} - \mathbf{s}\|^2$$

where $\mathbf{s}$ is a Lagrange multiplier associated with the relaxed constraint. A is matrix obtained by considering absolute values of matrix U. Every column of U encodes edges of the graph. Let there be an edge between vertices i and j in l^{th} edge then l^{th} column has exactly two non-zero entries such that $U_{i,l} = 1$ indicating starting of l^{th} edge, $U_{j,l} = -1$ indicating ending of l^{th} edge and rest of the entries of U_l are zero. The above unconstrained optimization problem is solved using standard toolboxes to obtain the unweighted undirected graph W.

4.2.2 Fisher Information Based Graph: Optimal Weighted Complete Graph

To construct a complete graph on given set of data points, similarity between every pair of data points is computed using a specified similarity/dissimilarity measure as discussed in section 4.1. In practice the Euclidean distance (a dissimilarity measure) is employed which does not take into account the relative importance of features. Other distance measures such as Mahalanobis distance take into account weighting each dimension by the variance along that dimension.

Ruiz *et al.* [67] proposed the use of Fisher information (FI) based distance metric which measures distance between two neighboring data points $\mathbf{x}_i$ and $\mathbf{x}_j$ by measuring the distance between the posterior distributions $p(c|\mathbf{x}_i)$ and $p(c|\mathbf{x}_j)$. $\mathbf{x}_j$ is obtained by introducing a small change to $\mathbf{x}_i$; $\mathbf{x}_j = \mathbf{x}_i + \delta\mathbf{x}_i$. In the case of binary classification (where

$c \in \{0, 1\}$), the posterior probability $p(c|\mathbf{x})$ takes the following form:

$$p(c|\mathbf{x}) = \frac{c + (1-c)e^{-a(\mathbf{x})}}{1 + e^{-a(\mathbf{x})}} \tag{4.6}$$

where $a(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}^T\mathbf{x}$, stands for generalized linear estimator of a evaluated at $\mathbf{x}$. The regression coefficients β_0 and $\boldsymbol{\beta}$ are estimated using linear logistic regression on the given dataset. Using the regression coefficients, distance between two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ is computed as:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \left| 2 \left\{ \arctan\left(e^{-a(\frac{\mathbf{x}(t)}{2})} \right) \right\}_{a(\mathbf{x}_i)}^{a(\mathbf{x}_j)} \right| \tag{4.7}$$

The above distance computation not only weigh each dimension of data points but also take into account class label information while performing the distance computation. For constructing a complete graph using FI metric based distance computation, the affinity matrix employ distance measure based on equation (4.7).

4.3 An Empirical Study

This work attempts for the first time to understand the role of optimal graphs in CC experimentally. In particular, optimal graphs are employed in CC method. Objective of this empirical study is to understand:

1. The role of optimal graphs in CC.
2. Effectiveness on the resulting clusters obtained through CC.
3. Convergence of the quality measures obtained using CC, internal as well as external, when optimal graphs are employed.

In order to understand the above mentioned objectives various graphs are constructed, namely complete graph using $\epsilon-$ neighborhood method as discussed in section 4.1, and complete graph using distance measure based on FI metric discussed in section 4.2.2, weighted general graph employing hard graph and soft graph formulations as discussed in section 4.2.1. SDP-CC is applied on these constructed graphs for measuring the internal quality, external quality and time taken to obtain cluster solution. Obtained results are compared in terms of quality of clusters and the effectiveness of optimal graph construction in SDP-CC is tested.

Synthetic Dataset: The synthetic dataset is generated as described in chapter 3. Instead of employing all six variations in data characteristics only 3 synthetic datasets are

Table 4.1: UCI Machine Learning Repository Datasets

Data set	n	dim	classes
`Sonar`	208	60	2
`Yeast`	1484	8	10
`Iris`	150	4	3
`Halfmoon`	200	2	2
`Pima`	768	8	2
`Titanic`	150	3	2
`Madelon`	4400	500	2
`Wine`	178	13	3
`Vowel`	528	10	11

used for experimentation, namely `Gaussian dataset` (overlapped), `Gaussian dataset` (well separated) and dataset containing noise (Gaussian dataset with Gaussian noise).

UCI Datasets: All the real world datasets employed for experimentation in [21] are considered. Table 4.1 describes the UCI datasets used for empirical study [5]. These datasets are of small size and are employed for experimentation to understand the merit of SDP-CC on **optimal graph** construction methods. Of all the datasets, `Madelon` has maximum number of data points and maximum dimension. `Vowel` dataset contains highest number of classes. As the two cluster SDP-CC formulation is considered, two classes at any time are considered in the case of multi-class datasets, namely `Yeast, Iris, Wine` and `Vowel`.

4.3.1 Edge Labeling Through Similarity Measures

The key component of CC lies in obtaining edge labels. After constructing graph using the above discussed methods, edges between every pair of vertices is labeled either positive or negative by examining the weight on the edge connecting those two vertices. If the weight between a pair of vertices is less than a specified threshold, that edge is labeled positive; otherwise that edge is labeled negative. Threshold is taken as the average value of all elements of the affinity matrix.

The objective of experimentation is to study the following:

1. Implication of the optimal graph construction (both complete and general graphs)

on approximation value of CC.

2. The influence of optimal graph construction on external quality obtained by CC.

3. Convergence of the above two quantities in optimal graphs and non-optimal graphs.

Effectiveness of SDP-CC using optimal graphs is studied by comparing along three distinct ways:

1. **Internal Quality - Approximation Value:** The ratio between the cost of the optimal solution to the cost of the solution produced by the approximation algorithm. This is computed in terms of ratio of the objective function value for the true clusters to the objective function value of the obtained clusters through SDP-CC. The approximation value of SDP-CC for MAXAGREE2 formulation is 0.87856 as discussed in section 2.1. The computed approximation value differs across datasets for complete graph constructed based on the value of ϵ. In majority of the cases, ϵ value needs to be tuned to obtain the specified theoretical value of 0.87856[2]. Figure 4.1 depicts the approximation value achieved by various graph construction techniques along with the theoretical value. The following are noted from this graph:

   (a) $\epsilon-$ neighborhood graph construction couldn't achieve the theoretical value on majority of the datasets. It also lags behind the optimal graph construction techniques with respect to the approximation value.

   (b) FI metric based complete graph construction achieves close approximation value to that of the theoretical one reported.

   (c) The theoretical approximation value reported in the literature for general graphs is 0.7666 [15]. This approximation value is taken as the baseline for comparing the performance in the case of hard and soft graphs. Both these techniques obtained approximation values close to that of the theoretical one.

   (d) In the case of `Wine` dataset, the $\epsilon-$ neighborhood graph's approximation value (0.876) is close to that of the given theoretical value (0.878). Soft graph formulation couldn't run on `Wine` and `Madelon` datasets and hence results on these two datasets are not reported in the Figure 4.1.

2. **External Quality:** Rand index is considered for measuring the quality of the obtained clusters using SDP-CC. Figure 4.2 depicts the rand index measure obtained

[2]For two cluster CC formulation, the reported theoretical approximation value is 0.87856 [31]

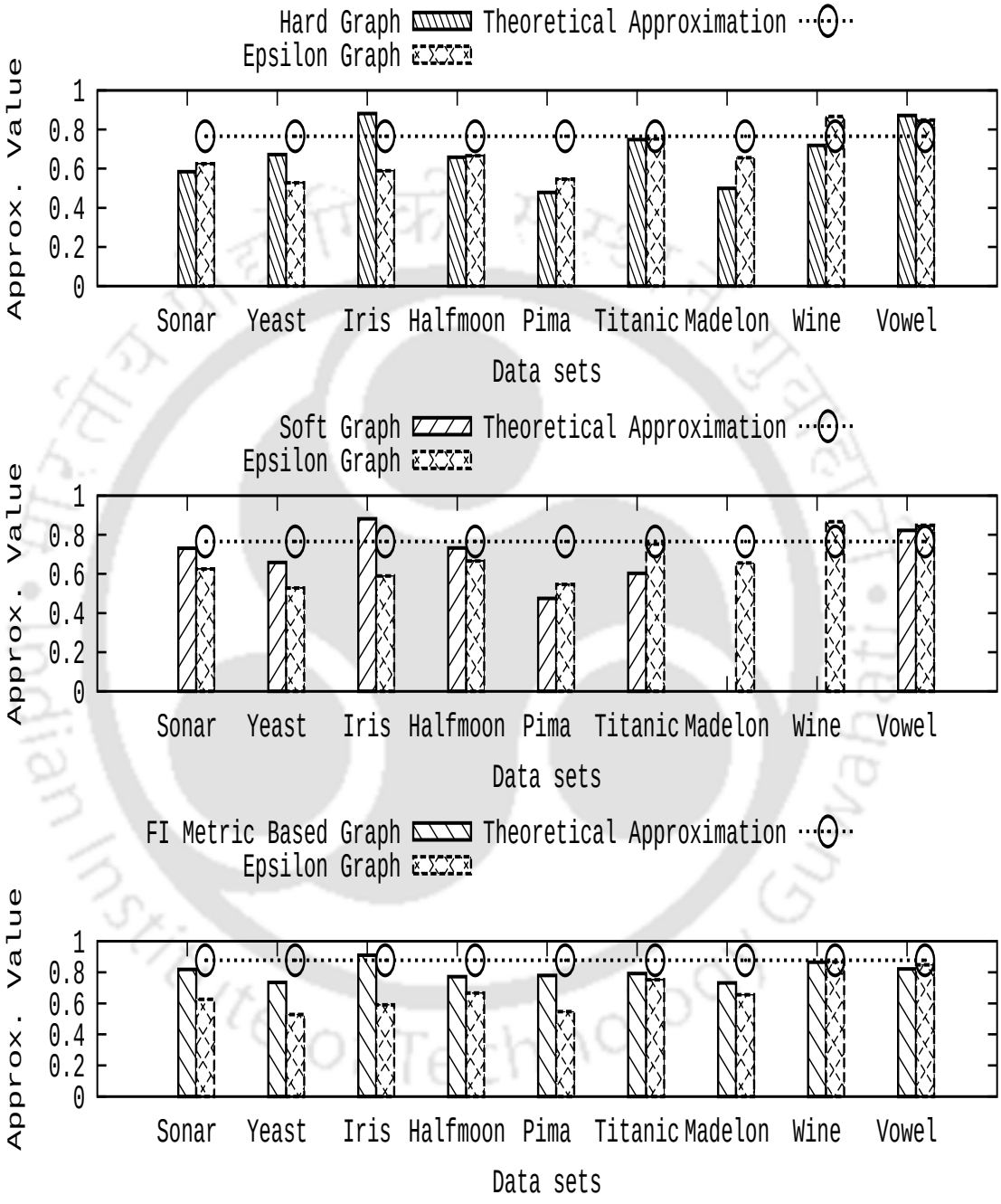

Figure 4.1: Approximation Value: Hard, Soft, FI Graph, $\epsilon-$ Neighborhood Graph vs Theoretical Approximation Result

using hard graph, soft graph, Fisher information metric based complete graph and $\epsilon-$ neighborhood graph. It is observed from this graph that the optimal graph construction techniques have significant edge compared to traditional way of constructing the graph and applying CC.

Comparison is also done among the optimal graph construction techniques, namely hard graph, soft graph and FI distance metric based complete graph to understand the competitiveness of optimal graph construction techniques. Figure 4.3 depicts the comparison of these methods. It is be observed from this graph that FI metric based complete graph construction technique outperforms the hard and soft graphs. Following are the reasons attributed to this result: (1) FI distance metric is a supervised way of obtaining optimal graph; however, soft graph and hard graph construction are totally unsupervised methods. (2) Both hard and soft graph formulations involve parameters which need to be tuned for obtaining the optimal graph.

3. **Time taken to obtain clusters:** Time taken to obtain clusters, once the graph is given to the SDP-CC solver, is measured. Figure 4.4 depicts time comparison of these methods. As hard and soft graph formulation yields sparse graphs, these graphs have significant edge over FI metric based distance measure. FI metric has an edge over $\epsilon-$ neighborhood graph. Note that while comparing the computational times, the time taken to obtain the (optimal) graph is excluded.

4.3.2 Convergence

Convergence of internal and external qualities of the SDP-CC formulation as the number of data points increase is studied. For this purpose, SDP-CC is first employed using a small number of data points. Both internal and external quality are measured on the resulting cluster solution. The number of data points is doubled and the experiment is repeated for understanding the internal and external quality obtained by SDP-CC. Figures 4.5, 4.6 and 4.7 depict the internal and external quality obtained on varying the number of data points using FI metric based graph. The following are observed from the obtained experimental results.

1. In the initial stages there is a significant variation of both the quality measures; however, as the number of data points increase, graphs flatten suggesting that both the internal and external quality indices converge.

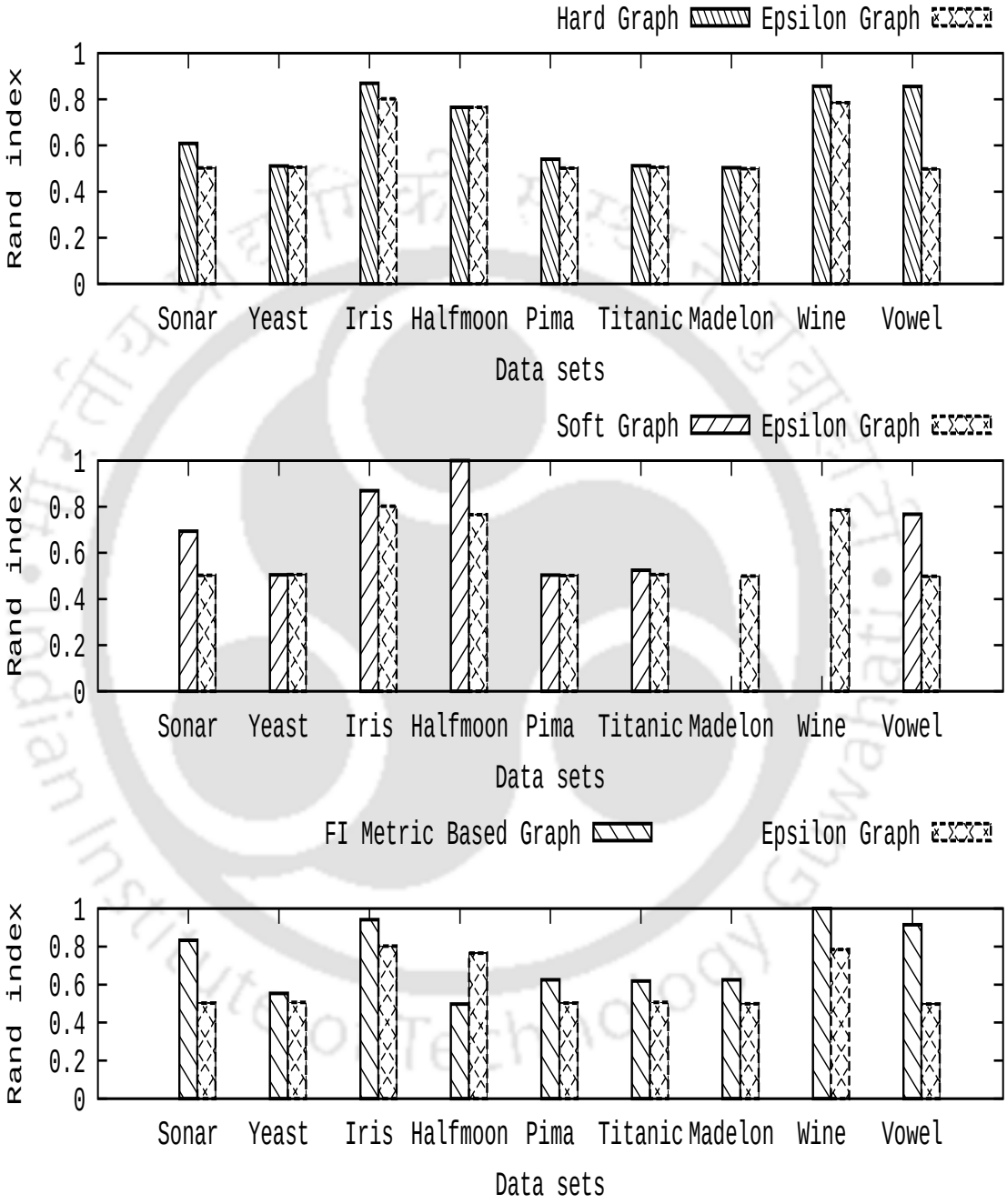

Figure 4.2: Rand Index: Hard, Soft, FI Graph vs $\epsilon-$ Neighborhood Graph

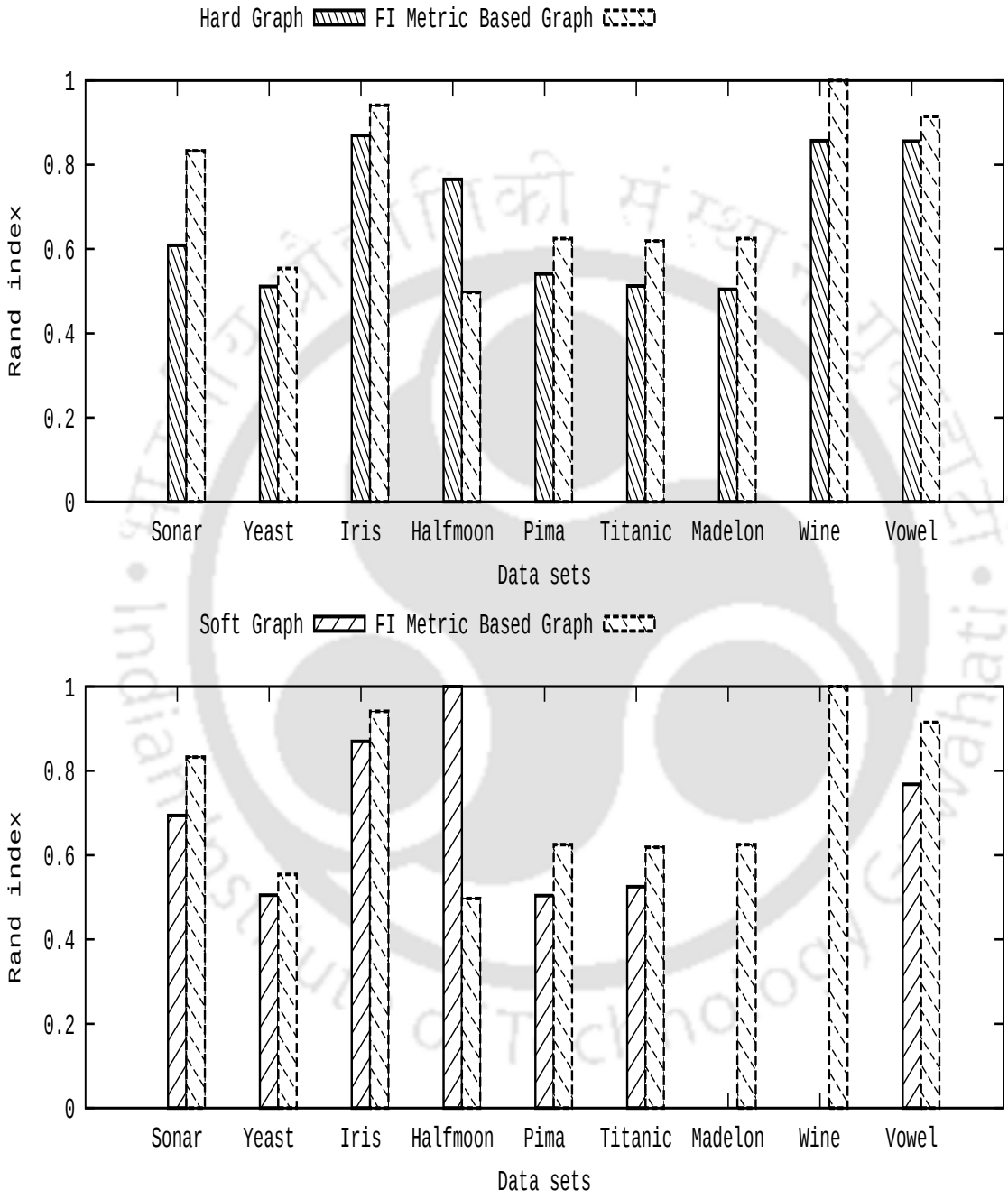

Figure 4.3: Rand Index: Hard and Soft Graph vs FI Metric Based Graph

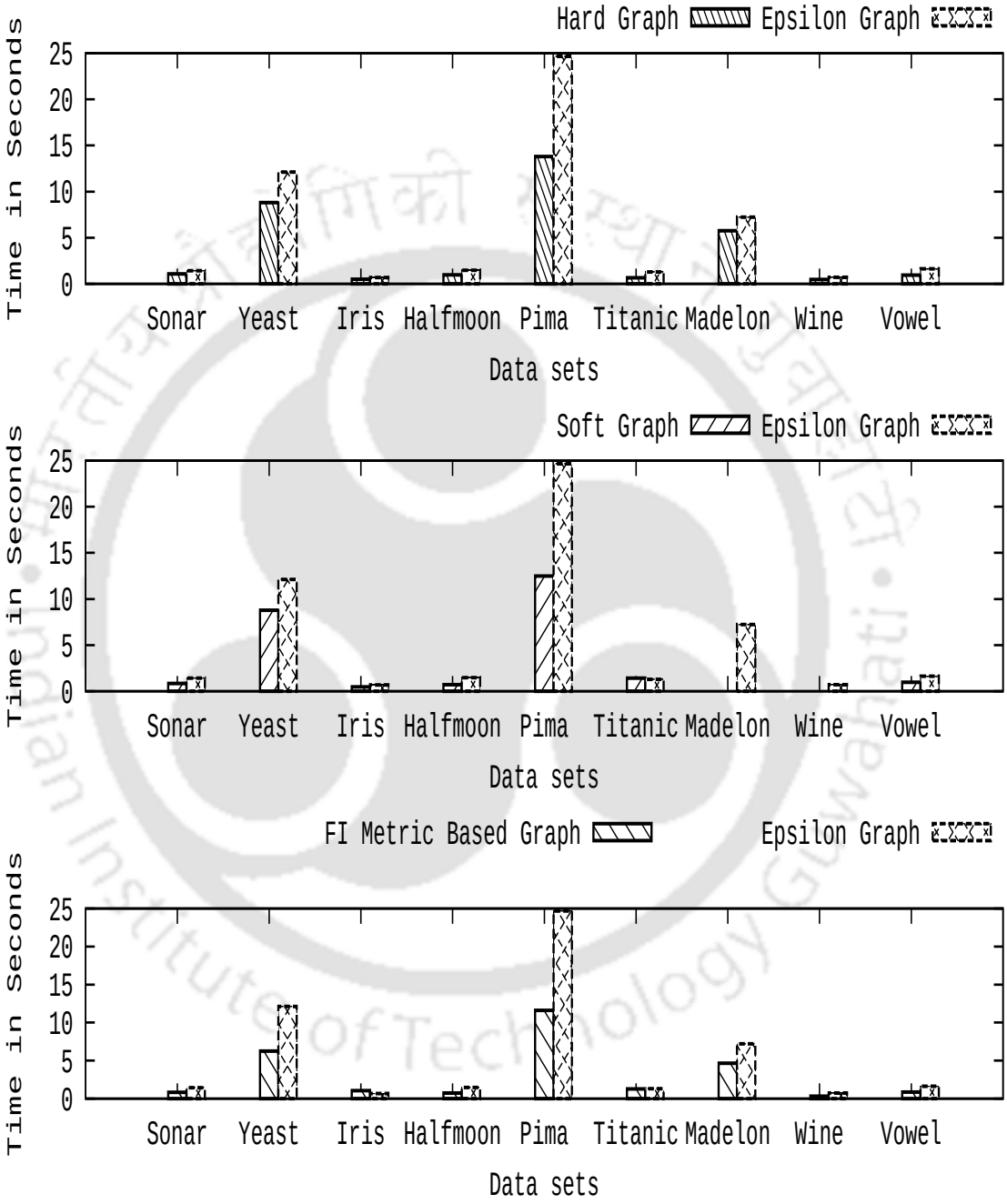

Figure 4.4: Time: Hard, Soft, FI Graph vs $\epsilon-$ Neighborhood Graph

2. The converged values are different for different graphs (that is FI metric based graph or Hard graph or $\epsilon-$ neighborhood graph). Except for the well-separated Gaussian dataset, rest of the datasets have different convergence values for FI metric based graph and $\epsilon-$ neighborhood graph. Both rand index and internal quality have different convergence values. This indicate the sensitivities exhibited by CC when employed different graphs in Figure 4.5.

3. In the case of a few datasets (`Pima, Sonar, Vowel, Wine` and `Yeast`), SDP-CC using FI metric based graph converges faster than that of SDP-CC based on $\epsilon-$ neighborhood graph.

For the hard graph based construction method, convergence results are depicted in Figures 4.8 and 4.9 for external quality and internal quality respectively for all the considered datasets. In Figure 4.8, the optimal graph based CC converges on majority of the datasets (read the '$\circ$' lines; this line should be parallel to x-axis upon convergence). Convergence is observed to be slightly faster than the non-optimal graphs on some datasets; for example `Pima, Titanic, Wine` and `Yeast`. A similar observation is made on the internal quality measure as depicted in Figure 4.9. In the case of hard graph as well, the observations made above on FI metric based graph vs $\epsilon-$ neighborhood graph match.

4.4 Conclusions

This chapter analyzed the effectiveness of CC with respect to different ways of obtaining optimal complete/general graphs and associated edge labels. Two techniques of constructing optimal graphs, namely hard and soft graphs which yield sparse weighted general graphs and FI distance metric based complete graph are examined to understand their merit in CC. It is experimentally noted that the CC's performance in case of optimal graph construction has an edge over the traditional way of constructing complete graphs. The comparison is carried along four diverse ways namely (1) approximation value (2) external quality (3) time taken to obtain the cluster solution and (4) convergence of CC. Optimal graph construction techniques outperform the traditional graph construction techniques along all the 4 considered ways.

A comparative study of constrained clustering methods is presented in the next chapter. In particular, performance of four different constraint clustering algorithms, namely `COP-KMEANS, cSC and its variants` are compared with CC's performance.

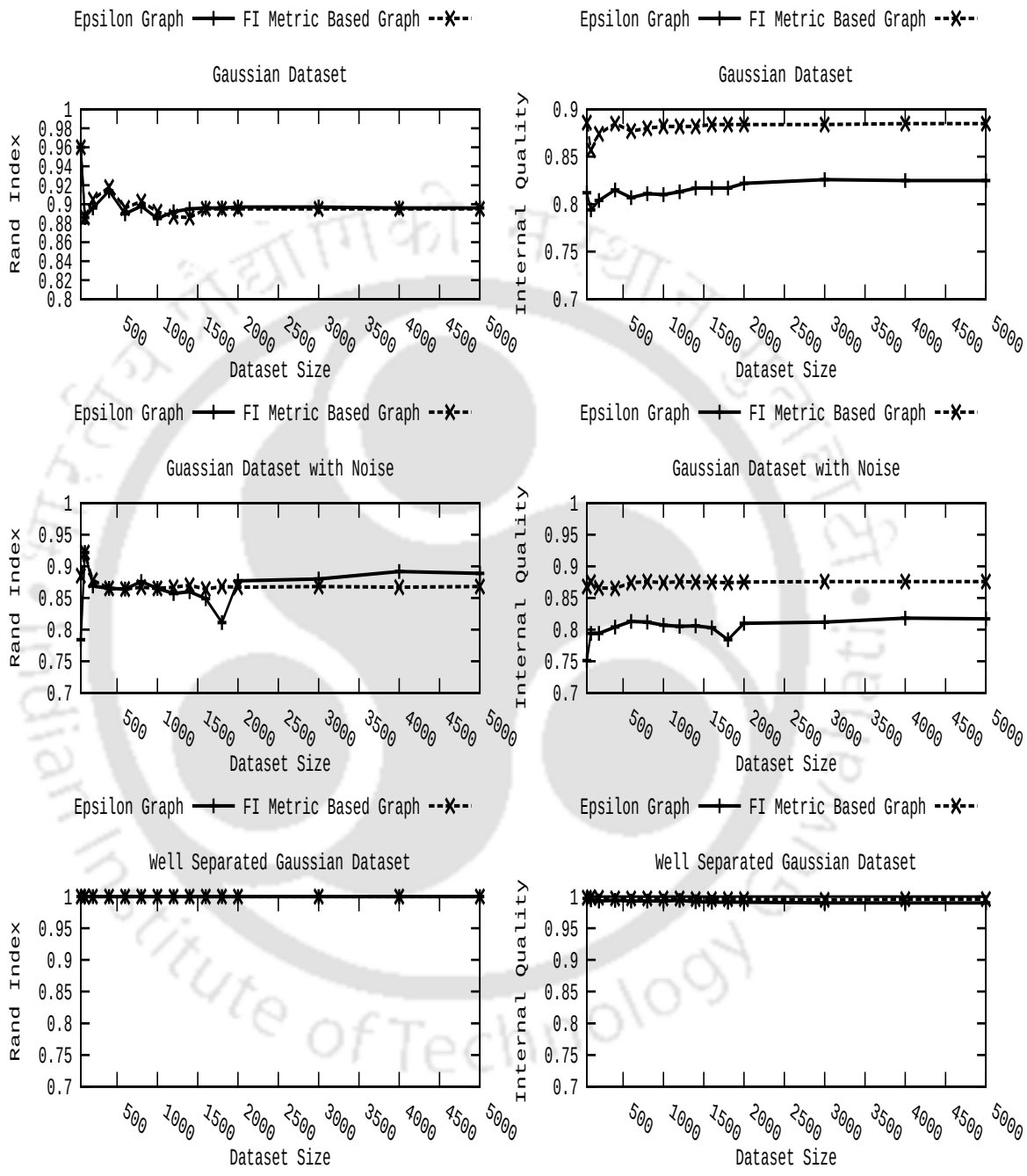

Figure 4.5: Convergence: FI Graph and $\epsilon-$ Neighborhood Graph; As the number of data points increase, observe that both the indexes are reaching constant value for three synthetic datasets.

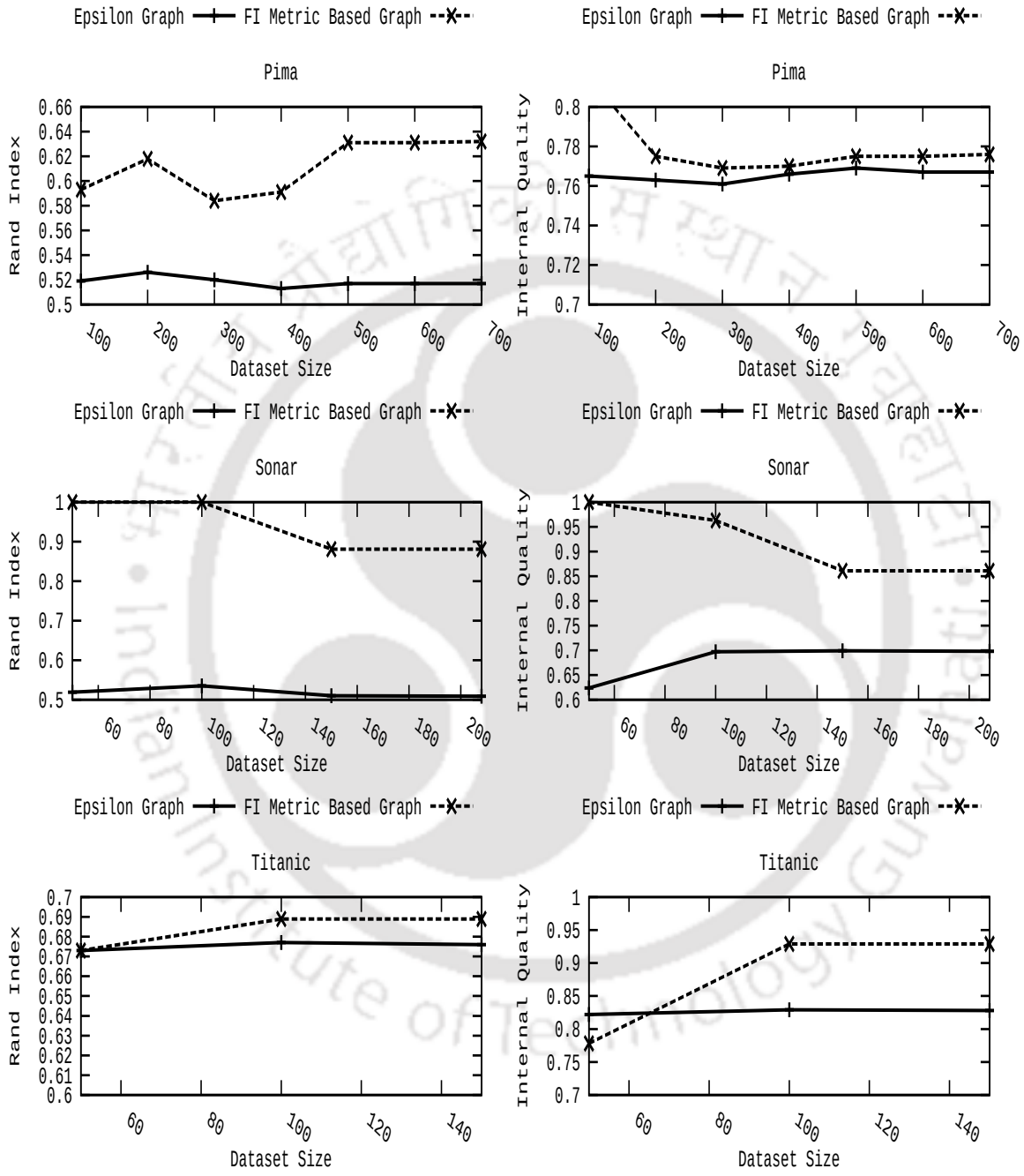

Figure 4.6: Convergence of Rand Index: FI Graph and $\epsilon-$ Neighborhood Graph using Real World Datasets.

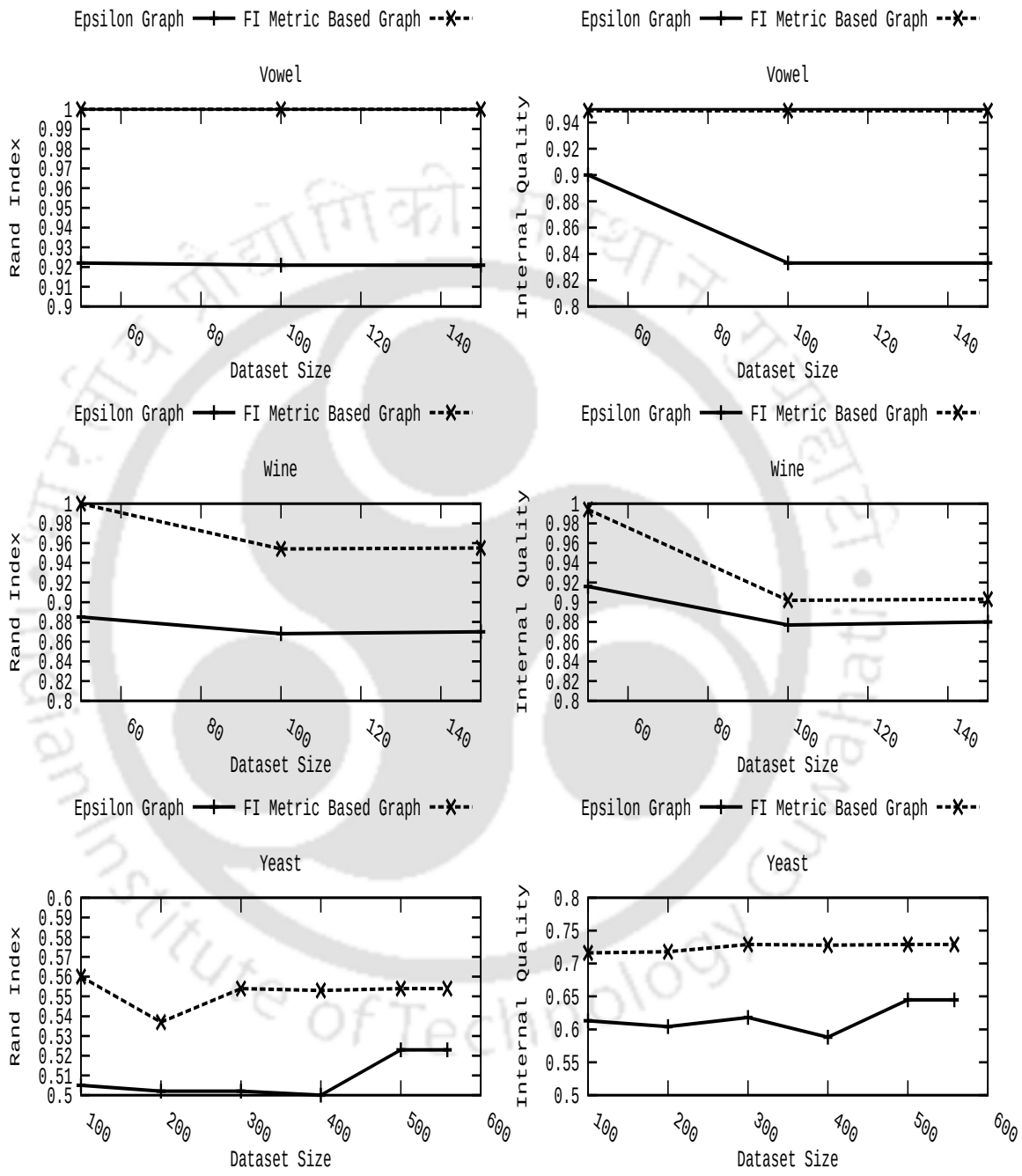

Figure 4.7: Convergence of Rand Index: FI Graph and $\epsilon-$ Neighborhood Graph using Real World Datasets.

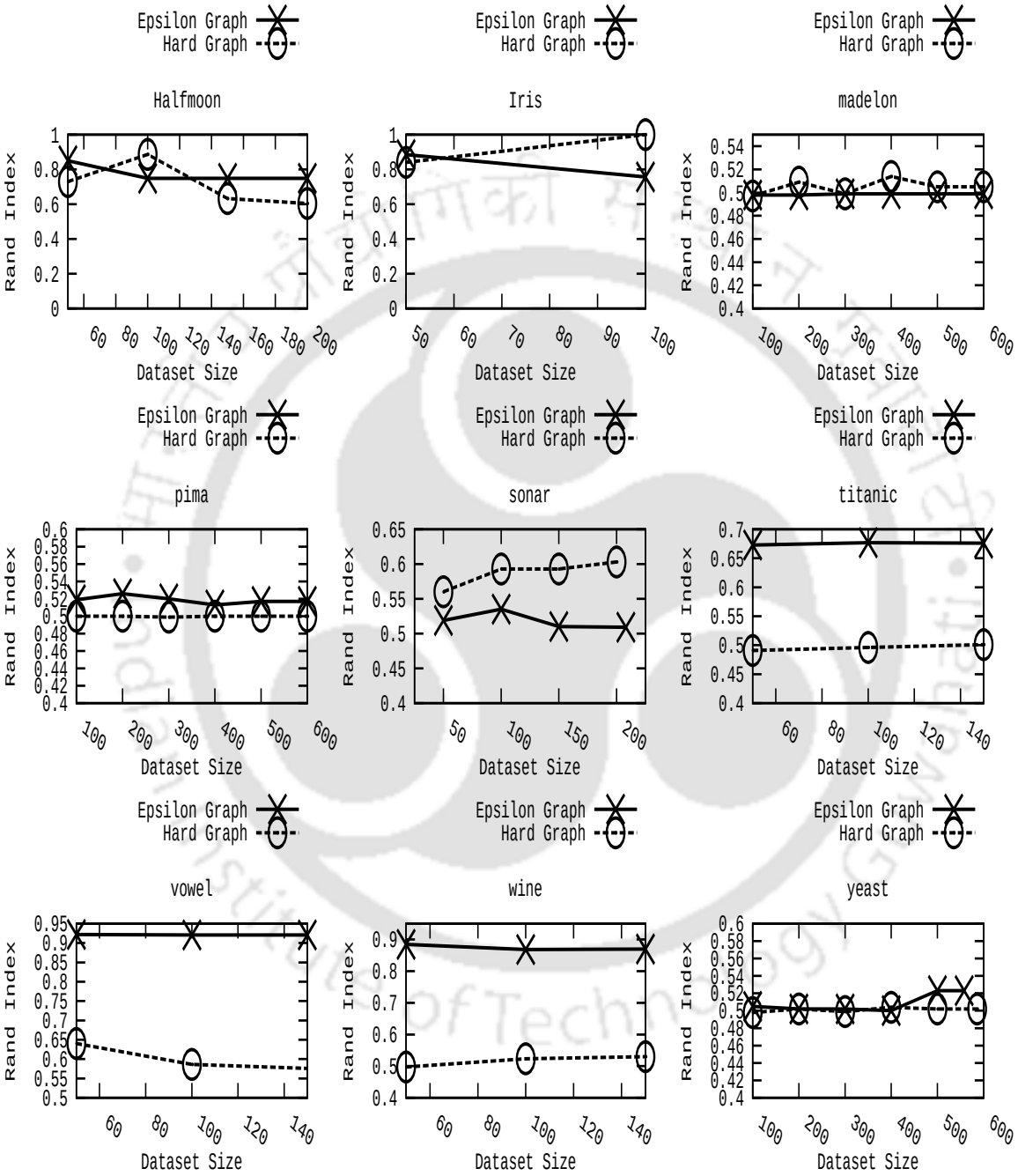

Figure 4.8: Convergence of Rand Index: Hard and $\epsilon-$ Neighborhood Graph

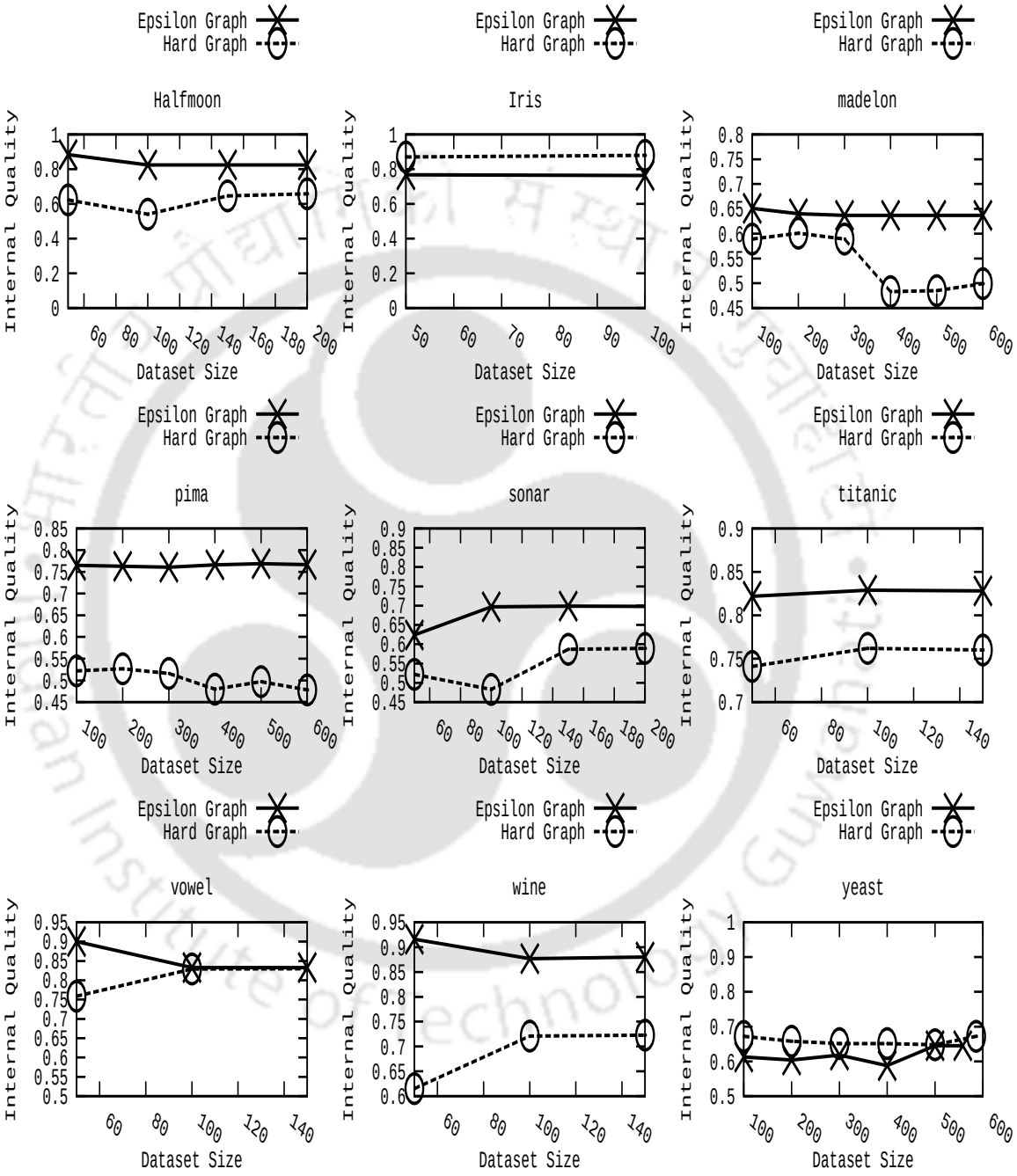

Figure 4.9: Convergence of Internal Quality: Hard and $\epsilon-$ Neighborhood Graph

Chapter 5

Comparison of Constrained Clustering Methods

5.1 Introduction

Semi-supervised clustering or constrained clustering is gaining prominence due to the flexibility in incorporating prior knowledge/domain knowledge or user inputs about possible members in a given partition [8]. The priori knowledge is expressed in the form of constraints which play a significant role in obtaining meaningful clusters. Two types of constraints are prevalent in context of constraint clustering. These are first introduced in [79] and are as given below:

- **Must-link:** This constraint, denoted as $c_{=}(\mathbf{x}_i, \mathbf{x}_j)$, specifies that two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ should belong to the same cluster.
- **Cannot-link:** This constraint, denoted as $c_{\neq}(\mathbf{x}_i, \mathbf{x}_j)$, specifies that two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ should not belong to the same cluster.

Must-link constraints form an equivalence relation by satisfying symmetric, reflexive and transitive relations. On the other hand cannot-link constraints are symmetric, anti-reflexive and non-transitive. Must-link and cannot-link constraints are part of input along with the dataset to a constrained clustering algorithm. Two types of constrained clustering methods are prevalent in the literature. These are (1) **constraint-based methods:** In this class, clustering algorithms are modified to respect the specified pairwise constraints. (2) **distance-based methods:** Given the pairwise constraints, this class of methods learn distance measure. Pairwise similarity is computed based on the learned metric.

Constrained clustering methods can further be classified into hierarchical [45], partitional [79, 80] and graph based methods [36, 39, 86] similar to the traditional clustering methods. Thiago *et al.* made an effort in comparing three variants of partitional constrained clustering algorithms, namely constrained vector quantization error (CVQE) [22], linear time CVQE (LCVQE) [62, 63] and Metric Pairwise Constrained K-Means (MPC K-means) [10]. Apart from assessing relative performance of these methods, the authors conclude that derived constraint set has no influence on quality of obtained clusters. That is derived constraints do not help achieve better cluster quality.

The objective of the present chapter is to compare the constrained clustering methods performance to that of CC as it is a variant of constrained clustering. In particular, CC is compared with three graph based constrained clustering methods:

1. Constrained spectral clustering [78].
2. Local proximity based spectral clustering [86].
3. Flexible constrained spectral clustering [81].

and a partitional constrained clustering method, constrained K-means clustering or COP-KMEANS [80]. The difference between CC and other considered methods is that CC uses *only* labeled information on edges, that is *only the constraint set*, for obtaining clusters; where as other methods consider constraints along with the input data. To make the comparison uniform, experiments are conducted using constrained spectral clustering method [78] and local proximity based spectral clustering method [86] such that these methods consider labels alone. In the other two clustering methods, such tuning is not possible and hence the comparison is not a fair one.

5.2 Constrained K-means Algorithm

K-means clustering algorithm is a well known clustering algorithm which has found its applications in diverse disciplines. Cluster centers are initialized (randomly) at the beginning of the K-means algorithm. Distance between each data point to every cluster center is computed. The data point in question is assigned to that cluster center which has least distance. Once the assignment of all the data points to their neighboring cluster centers are completed, cluster centers are updated based on the data points which are closest to the cluster center in question. The above procedure (of assigning data points to their nearest cluster centers) is repeated until there is no variation in cluster centers [52].

This well known algorithm do not take into account the available prior knowledge directly. Wagstaff *et al.* has proposed a variant of the K-means algorithm which take into account the prior knowledge about the data points [80]. In particular the prior knowledge is expressed in terms of must-link and cannot-link constraints. The key idea in assigning a particular data point $\mathbf{x}_i$ to its nearest cluster center (C_j) is to check for two critical cases:

1. For each data point $\mathbf{x}_k \in C_j$ if there is a cannot-link constraint between $\mathbf{x}_i$ and $\mathbf{x}_k$ then $\mathbf{x}_i$ cannot be assigned to the cluster C_j even though C_j is the nearest cluster center.

2. For each data point $\mathbf{x}_k \notin C_j$ if there is a must-link constraint between $\mathbf{x}_i$ and $\mathbf{x}_k$, then $\mathbf{x}_i$ cannot be assigned to cluster center C_j so as to respect the must-link constraint.

The above two cases are examined before assigning $\mathbf{x}_i$ to its nearest cluster center C_j in the K-means algorithm. The complete COP-KMEANS algorithm is reproduced below [80].

1. Initialize cluster centers $C_1, C_2, \cdots, C_K$.

2. For each data point $\mathbf{x}_i$ find the nearest cluster center. Let the nearest cluster center be C_j.

3. Check for the two critical cases described above. If they violate then safely assign $\mathbf{x}_i$ to cluster center C_j.

4. Update the cluster centers.

5. Repeat steps 2 to 4 until convergence.

6. Return cluster centers: $\{C_1, C_2, \cdots, C_K\}$.

Superiority of including constraints is shown through experimental results over traditional K-means clustering algorithm.

5.3 Constraint Spectral Clustering

Inspired from the success of constraint clustering methods in partitional clustering such as K-means (COP-KMEANS), a recent effort is made to incorporate these constraints in the graph based clustering method, namely spectral clustering algorithm. To incorporate instance level constraints in spectral clustering method two directions are proposed:

1. Include pairwise constraints in the affinity matrix [39]. The constraints are included by modifying the affinity matrix as follows:

$$w_{ij} = \begin{cases} +1 & if\ c_{=}(\mathbf{x}_i, \mathbf{x}_j) \text{ is true} \\ 0 & if\ c_{\neq}(\mathbf{x}_i, \mathbf{x}_j) \text{ is true} \end{cases}$$

The modified affinity matrix is explicitly incorporated in the graph Laplacian matrix $L = D - W$; where D is the diagonal matrix in which every diagonal element denotes the total degree of that vertex. Eigenvectors of L are clustered using a simple K-means algorithm for obtaining the final clusters.

2. Include pairwise constraints in the objective function of spectral clustering [81, 82]. To achieve this objective, must-link and cannot-link constraints are embedded in a new matrix, Q, for every pair of data points as follows:

$$Q_{ij} = Q_{ji} = \begin{cases} +1 & if\ c_{=}(\mathbf{x}_i, \mathbf{x}_j) \text{ is true} \\ -1 & if\ c_{\neq}(\mathbf{x}_i, \mathbf{x}_j) \text{ is true} \\ 0 & \text{no constraint is specified} \end{cases} \tag{5.1}$$

The measure $\mathbf{u}^T Q \mathbf{u}$ signifies how well the constraints are satisfied; where $\mathbf{u}$ is an indicator vector denoting whether an element $\mathbf{u}_i$ in $\mathbf{u}$ belongs to $+1$ or -1 cluster in the case of binary clustering. The Q matrix which encodes the must-link and cannot-link constraints is explicitly introduced into the objective function of the spectral clustering method. The modified objective function of the spectral clustering method, cSC, is given as follows [78].

$$J_{cSC} = \gamma J_{SC} + (1 - \gamma) J_{CM} \tag{5.2}$$

Where J_{SC} stand for the spectral clustering objective and is given as:

$$\text{min.} \sum_{i,j} (\mathbf{u}_i - \mathbf{u}_j)^2 w_{ij} = \mathbf{u}^T\ L\ \mathbf{u}$$

J_{CM} stand for the cannot-link and must-link objective and is given by

$$\min_{\mathbf{u}} J_{CM} = - \sum_{c_{\neq}(\mathbf{x}_i, \mathbf{x}_j)} (\mathbf{u}_i - \mathbf{u}_j)^2 + \sum_{c_{=}(\mathbf{x}_i, \mathbf{x}_j)} (\mathbf{u}_i - \mathbf{u}_j)^2 \quad = \mathbf{u}^T\ L_Q\ \mathbf{u}$$

where L_Q is the Laplacian matrix of the constrained graph. L_Q is computed as $L_Q = D_Q - Q$ with D_Q is the degree matrix of constraints graph. Substituting J_{SC} and J_{CM} in equation (5.2), the constrained spectral clustering is given as:

$$\min_{\mathbf{u}} J_{cSC} = \mathbf{u}^T \left(\gamma \times L + (1-\gamma) \times L_Q\right) \mathbf{u}. \tag{5.3}$$

For empirical study of cSC, Wacquet *et al.* [78] obtained the constraints from the class label information of the given data points. To compare CC with constraint spectral clustering, constraints are generated in this chapter based on the weight of the edges. To understand the strength of constraints *alone*, the γ parameter in equation (5.3) is equated to 0. The choice of $\gamma = 0$ is motivated from the fact that the first term in (5.3) is neglected which amounts to nullifying the spectral clustering objective J_{SC} and provide maximum weight to the constraint graph on the data points. Thus one can draw reasonable conclusions on both constrained spectral clustering method and correlation clustering method.

5.4 Spectral Constraint Clustering with Local Proximity Measure

Kamvar *et al.* introduced pairwise constraints in the affinity matrix as follows [39]:

$$w_{ij} = \begin{cases} +1 & if\ c_{=}(\mathbf{x}_i, \mathbf{x}_j) \text{ is true.} \\ 0 & if\ c_{\neq}(\mathbf{x}_i, \mathbf{x}_j) \text{ is true.} \end{cases}$$

The above affinity matrix is reformulated into a transition probability matrix. The central idea is inspired from the *random surfer* model in which a reader begins with a document of interest. The reader moves to *next document* which is *similar* to the current document. The linkage between similarities and transitions probabilities is obtained by normalizing the affinity matrix with degree of each vertex. That is $N = D^{-1}W$ where D is a diagonal matrix with each diagonal element representing degree of corresponding vertex. An alternative method of obtaining this linkage is:

$$N = \frac{1}{d_{\max}} \left(W + d_{\max} I - D\right)$$

where $d_{\max}$ denotes maximum degree that is maximum value of the diagonal elements of D. Qianjun Xu *et al.* identified that singleton clusters are potentially due to the presence of outliers in the data and the way in which N is constructed in the Kamvar's method. To

address the problem of outliers Xu [86] suggested that transition probability matrix to be $N = D^{-1}W$. At the time of constructing the matrix W, Xu *et al.* suggested a heuristic which adheres to the *local proximity structure* [86]. Local proximity structure states that each data point belongs to *same cluster* as its closest neighbor. In this heuristic, affinity between a data point and its k neighbors should be greater than a specified value. The modified algorithm is argued to perform better than the Kamvar *et al.* [39] method.

5.5 Flexible Constrained Spectral Clustering

Wang *et al.* [81] identified two major limitations with the constrained clustering methods. These are:

1. The must-link and cannot-link constraints are hard constraints. In particular application domains, constraint set is available naturally in the form of soft constraints.

2. Partitions are achieved in the above constrained clustering methods by obeying *all* the constraints. In practice, the constraint set itself may have contradictory constraints. Such constraint set cannot be satisfied thus results in an empty cluster solution.

The above two limitations are addressed by [81]. First limitation is addressed by allowing the constraint matrix Q given in equation (5.1) to take real values. Each element, Q_{ij}, in this matrix represent a soft constraint (degree of belongingness) between data points $\mathbf{x}_i$ and $\mathbf{x}_j$.

Second limitation is addressed by obtaining a measure that describes how well the constraint set, encoded as Q matrix, is satisfied. The measure of constraint satisfaction is given by the quantity: $\mathbf{u}^T\ Q\ \mathbf{u}$; where elements of $\mathbf{u}$ denote which cluster the point in question belong. This measure is incorporated as one of the constraint while optimizing the objective function of original spectral clustering.

These two modification lead to the flexible constrained spectral clustering (FCSC or `Wang's method`) whose formulation is given below:

$$\begin{aligned} \underset{\mathbf{v}\in\mathbb{R}^n}{\arg\min} \quad & \mathbf{v}^T\ L_n\ \mathbf{v} \\ \text{subject to.} \quad & \mathbf{v}^T\ Q_n\ \mathbf{v} \geq \alpha \\ & \mathbf{v}^T\ \mathbf{v} = \text{vol}(\mathcal{G}) \\ & \mathbf{v} \neq D^{-\frac{1}{2}}\mathbf{e} \end{aligned} \tag{5.4}$$

Table 5.1: Additional Real World Datasets

Dataset	n	dim	classes
Hepatitis[1]	80	19	2
Ionosphere	351	34	2
Glass	214	9	6
Ecoli	336	7	8
Pendigits	3498	16	10
Vehicle	946	18	4

1: Number of data points excludes missing values.

where L_n denotes the normalized graph Laplacian matrix, Q_n denotes the normalized constraint matrix, α specifies to what extent the constraint set should be satisfied. α is a parameter provided by the user, vol($\mathcal{G}$) denotes the total degree of vertices. Optimal solution for the above formulation is obtained by solving a generalized eigenvalue problem involving L_n and Q_n matrices as given below.

$$L_n \mathbf{v} = \lambda \left(Q_n - \frac{\beta}{\text{vol}(\mathcal{G})} I \right) \mathbf{v}$$

where β is a parameter introduced to formulate the equation (5.4) as generalized eigenvalue problem.

5.6 Datasets and Constraint Generation

Two classes of datasets are considered for experimental comparison. Synthetic dataset and real world dataset. Detailed description of the synthetic dataset is given in chapter 3. A total of 15 datasets from UCI machine learning data repository is considered. In addition to the 9 datasets described in chapter 4, six additional datasets are considered for a thorough comparison. These six datasets are detailed in Table 5.1.

Constraint set generation: The constraint set generation play a central role in performing experiments. Constraints are generated based on the *class labels* of each of the data point in case of COP-KMEANS, local proximity constraint spectral clustering and flexible constrained spectral clustering methods. That is when class labels of two data points match, a must-link constraint is generated. Otherwise cannot-link constraint is generated. Total number of constraints generated is restricted to ten percentage of the size of Q matrix given in equation (5.1).

In the case of cSC method, the constraints are generated based on distance measure between pair of data points. If distance between pair of data points $\mathbf{x}_i$ and $\mathbf{x}_j$ is less than a specified threshold value, then a must link constraint is generated between $\mathbf{x}_i$ and $\mathbf{x}_j$. Otherwise cannot- link constraint is generated.

5.7 Result

The performance of CC is compared with that of the 4 constrained clustering algorithms. Results on the synthetic datasets are presented in Table 5.2. The obtained results are summarized below:

1. **CC vs COP-KMEANS:** From the Table 5.2 it is observed that CC competes with COP-KMEANS algorithm on considered synthetic datasets. In the case of well separated dataset, both COP-KMEANS and CC achieves rand index value of 1.0 signifying that they both obtain clusters that match the true clusters. In the other datasets, CC lags behind COP-KMEANS clustering algorithm. The key difference between these two algorithms is that CC uses *constraint set* alone for clustering whereas COP-KMEANS uses constraint set as well as input data points for obtaining clusters.

   In the case of real world datasets (refer to Table 5.3), COP-KMEANS clearly outweights CC method even with as fewer as 10% constrains. COP-KMEANS is also very effective from the perspective of computational time to obtain clusters compared to CC method. As CC and COP-KMEANS algorithms are heterogeneous in nature, that is one works on graph dataset and constraints alone while another works on dataset along with constraints, the observations resulting from experiments have no surprise in store.

2. **CC vs cSC:** cSC is implemented in such a way that a direct analogy with CC could potentially be drawn. In particular, the choice of $\gamma = 0$ forces cSC to use only *constraint set* information while performing clustering. Results with respect to synthetic datasets are depicted in the Table 5.2. One can observe from this figure that CC's performance is superior compared to cSC method irrespective of the dataset characteristics. However, when the constraints are generated according to the class labels and when γ assumes a value greater than 0, cSC's performance improves significantly.

Results on the real world datasets for cSC and CC are presented in Table 5.3. CC outperforms cSC on the considered real world datasets. On ten out of 15 datasets, CC cluster quality is observed to be better than that of cSC.

3. **CC vs Local Proximity:** On real world datasets, (Table 5.3) CC is observed to compete with the local proximity based constraint spectral clustering method. In seven out of 15 datasets, CC outperforms local proximity measure based algorithm. In case of local proximity based method, the input data point information proved to be vital for its better performance. By increasing the number of constraints from 10% to 100%, the affinity matrix converts to the Q matrix. Clustering based on this matrix results in significantly poor performance compared to CC. In the case of synthetic datasets, (Table 5.2) SCLP outperforms CC. In the case of `Well Separated Gaussians` dataset, CC performs at par with that of SCLP.

4. **CC vs Flexible Constrained Spectral Clustering:** The free parameter involved in Wang's method, α, signifies the extent to which the constraint set is satisfied. The larger this value, the large is the coverage of the constraints. In the present experimentation the value of α is set $\frac{\lambda_{max}}{2}$. From the experimental results it is noted that Wang's method has an edge over CC method from both synthetic and real world datasets as depicted in Table 5.2 (Wang's method vs CC on synthetic datasets) and Table 5.3 (Wang's method vs CC on real world datasets).

   The value of α is varied from 0.2 to 0.9 and the impact of quality of the obtained clusters is observed. Figures 5.1 to 5.4 present the experimental results obtained on the considered real world datasets. The constraint coverage has no direct bearing on the quality of the obtained clusters within Wang's method. That is even when α takes a lower value, the obtained quality has not significantly degraded.

   Comparing Wang's method with CC, on eight out of fifteen datasets CC is found to be competing with Wang's method. Which shows the strength of utilizing constraint set alone for obtaining clusters.

From the above experimental results it is observed that CC performs poorly when compared to other variants of constrained clustering algorithms. However, when the constrained clustering algorithms are stretched to extreme extent of using only constraint information for clustering, CC outperforms those methods.

Table 5.2: Result of constraint clustering methods along with CC on Synthetic datasets

Data sets	SCLP	FCSC	COP-Kmeans	cSC	CC
Gaussian	0.831	0.958	0.934	0.701	0.896
Gaussian noise	0.788	0.957	0.955	0.669	0.889
Wellseparated data	0.996	0.999	1.000	1.000	1.000

Table 5.3: Result of constraint clustering methods along with CC on Real world datasets

Data sets	SCLP	FCSC	COP-Kmeans	cSC	CC
Wine	0.969	0.857	0.634	0.544	0.870
Ionosphere	0.934	0.815	1.000	0.498	0.594
Hepatitis	0.881	0.818	0.859	0.691	0.724
Glass	0.872	0.814	0.565	0.527	0.946
Ecoli	0.657	0.913	0.543	0.841	0.872
Vehicle	0.663	0.712	1.000	0.537	0.701
Pen digits	0.883	0.872	1.000	0.694	0.851
Yeast	0.850	0.808	1.000	0.556	0.523
Sonar	0.773	0.944	1.000	0.572	0.535
Iris	0.851	0.868	1.000	0.632	0.886
Half Moon	0.510	0.970	0.914	0.507	0.748
Pima	0.949	0.826	1.000	0.586	0.517
Madelon	0.938	0.884	0.765	0.578	0.499
Vowel	0.497	0.634	0.880	0.729	0.635
Titanic	0.548	0.875	0.910	0.583	0.676

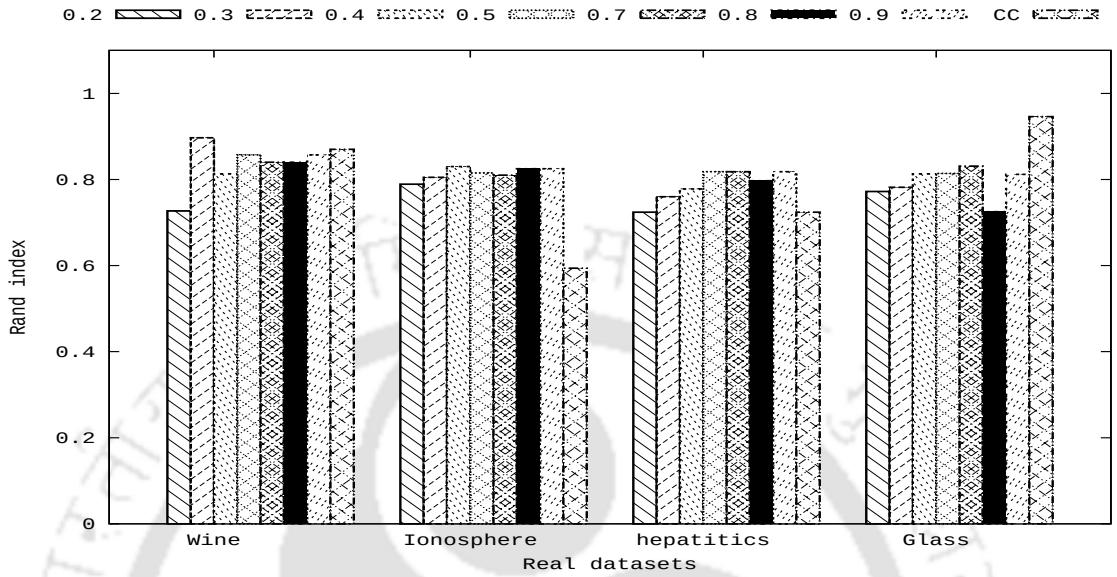

Figure 5.1: Wang's Method: Variation in α for `wine`, `Ionosphere`, `Hepatitis` and `Glass`

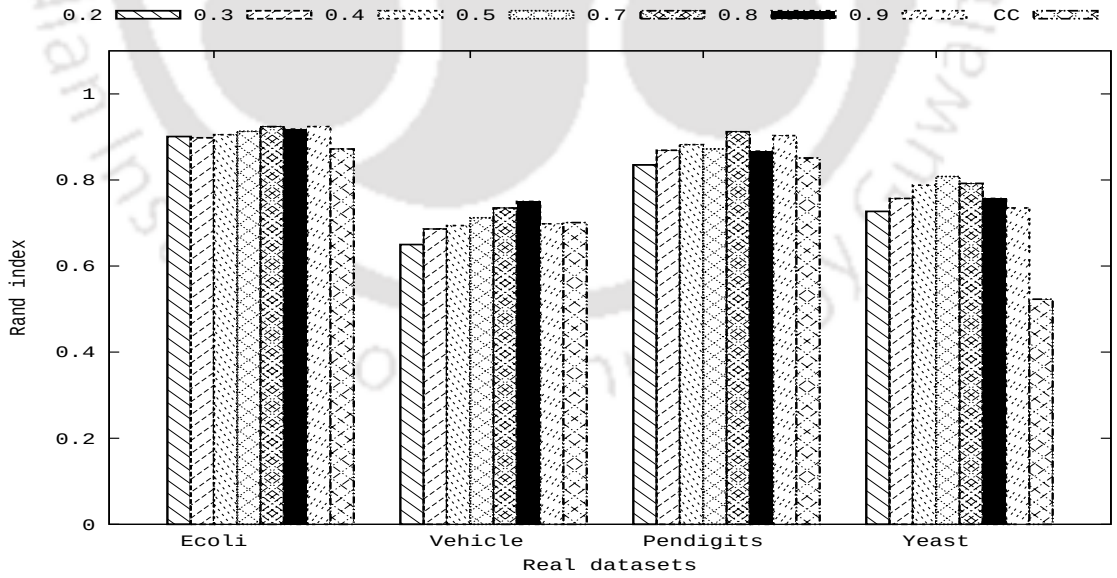

Figure 5.2: Wang's Method: Variation in α for `Ecoli`, `Vehicle`, `Pendigits` and `Yeast`

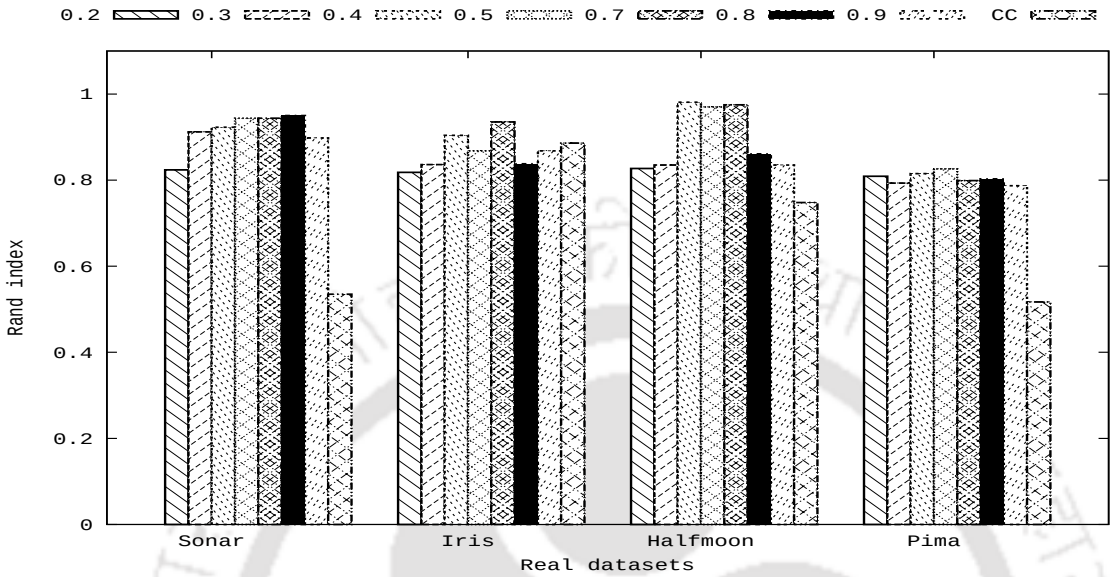

Figure 5.3: Wang's Method: Variation in α for `Sonar, Iris, Halfmoon` and `Pima`

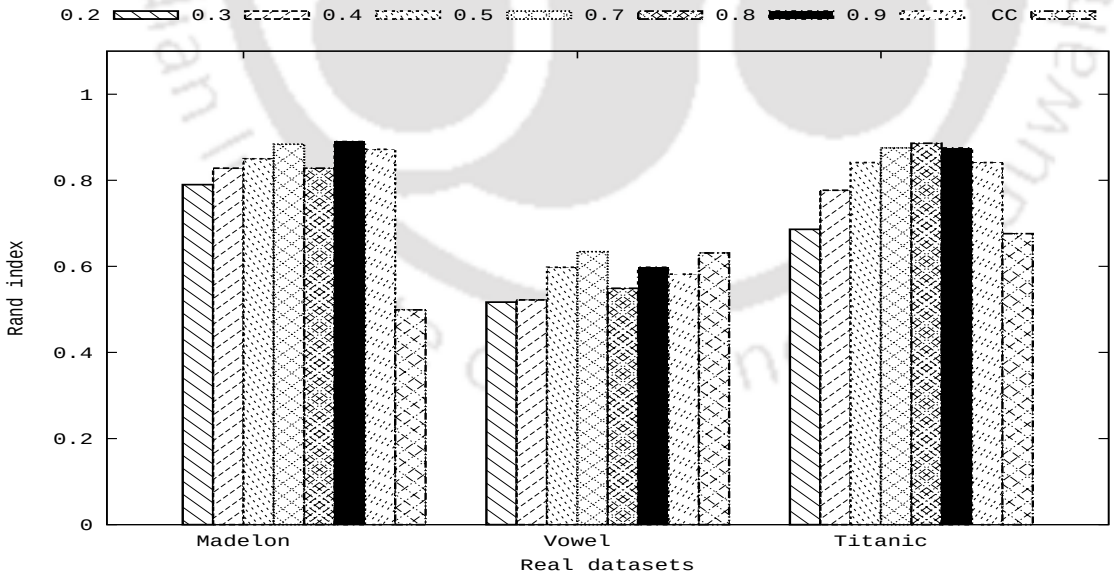

Figure 5.4: Wang's Method: Variation in α for `Madelon, Vowel` and `Titanic`

5.8 Summary

Four variants of constraints clustering methods are compared with CC. It is noted experimentally that all these variants exhibit their strength over CC from the quality of obtained clusters. However when only labeled information is allowed while performing clustering, CC significantly out performs the other constraint clustering variants. cSC and local proximity algorithms are two such cases when only constraints are allowed for clustering, their performance degrades significantly. From the computational time perspective, CC significantly lags behind the compared methods.

The take home point from these differences is that when constrained clustering methods which utilize *only* constraint information are inferior compared to correlation clustering. Other methods do have a clear edge over constraint clustering method.

As the objective functions for the considered constrained clustering methods vary, it is not directly feasible to compare the internal quality of correlation clustering method with the considered methods. For example, the objective function value obtained from flexible constrained spectral clustering is *minimize* $\mathbf{v}^T L_n \mathbf{v}$; where as in the case of MAXAGREE[2] formulation, the objective is to maximize the agreements. Therefore the internal quality cannot be compared directly. Due to these differences across all the four considered constrained clustering methods, we chose to limit comparison with respect to external quality measure, namely Rand Index.

Quality of the obtained clusters through CC is observed to be competitive when used labeled information *alone* available on the edges. The applicability of CC in practice is limited by (1) difficulty in constructing graphs (2) solving SDP formulation given in (2.6) is difficult as the size of the graph increases, in terms of number of vertices and number of edges. Chapter 4 address the problem of graph construction through methodical approach, namely optimal graph construction. Next chapter focuses on addressing the problem of scalability of CC by *reducing the number of variables* involved in the SDP formulation.

Chapter 6

Scalability of CC Through Variable Reduction

The MAXAGREE formulation of CC has not been *applied* in practice due to the computational time involved in obtaining partitions as CC formulation involves solving an expensive SDP formulation having large number of variables. The computational complexity to solve the SDP formulation is given by $O(n^{4.5})$ [77] where n is the number of variables involved in the SDP formulation (or number of vertices in G or number of data point in the training set).

In order to apply CC to large scale datasets, speeding of the SDP formulation is the key. Burer *et al.* [13] proposed a variable reduction method to obtain approximate solution to the original SDP formulation through low rank Broyden Fletcher Goldfarb Shanno (BFGS) method [70]. Their method has been applied to MAX CUT problem and reported significant gains in the computational time and memory requirements.

The above variable reduction method is extended to the MAXAGREE CC formulation for scalability. The key contributions in this chapter are:

1. Address scalability of CC.
2. Apply CC in practice for **large datasets**[1].
3. Analyze CC's external quality on variety of datasets including pure graph datasets obtained from benchmark graph dataset repository `http://staffweb.cms.gre.ac.uk/~wc06/partition/`.

[1]Large dataset: large number of nodes as well as edges

4. As CC is one form of constrained clustering method, CC cluster's quality is compared with that of constrained spectral clustering method [78], which takes into account pairwise data point constraints in obtaining clusters.

   Though CC is a graph based clustering method, it uses only labeled information available on edges unlike other scalable graph based clustering methods *viz.* multilevel recursive spectral bisection (MSB) [7] and fast multilevel partitioning methods [43]. These methods involve three steps namely reducing the size of the graph, finding Fiedler vector of the reduced graph, interpolate to the Fiedler vector of the original graph. These methods do not utilize the edge label information. CC obtains partitions based on the edge labels alone and falls under the constraint clustering methods. Comparison of CC and its variants is therefore confined to graph based constrained clustering methods namely constrained spectral clustering [78].

5. From external clustering quality point of view, scalable CC is experimentally observed to be a competitive method having better quality compared to original CC formulation and constrained spectral clustering method.

6.1 Various SDP Relaxation Techniques for CC

The equation (2.6) is difficult to solve when either the number of vertices or the number of edges is large. There are several relaxations that are known in the convex optimization literature for solving the SDP formulation involving large number of vertices or edges. These are broadly grouped as given below:

1. **Rank Relaxation:** This is introduced by Goemans and Williamson in their seminal work [32] for MAX CUT which is an NP-complete problem. The authors proposed a polynomial time algorithm by introducing relaxations to the original MAX CUT problem. The MAX CUT formulation is extended to the MAXAGREE CC formulation which is described as follows: Equation (2.6) can be re-written as:

$$\begin{aligned} \max_{V} \quad & \text{trace}(CV) \\ \text{subject to} \quad & diag(V) = \mathbf{e} \\ & \text{rank}(V) = 1 \\ & V \succeq 0 \end{aligned} \tag{6.1}$$

   where

$$C = \frac{1}{4}\left((Diag(W\mathbf{e}) - W) + (Diag(W\mathbf{e}) + W)\right)$$

e is the vector of all ones. $diag(V)$ denotes diagonal elements of any matrix V and $Diag(\mathbf{v})$ denotes a diagonal matrix whose diagonal elements are the elements of vector **v**. Introducing relaxation on the matrix V by allowing the matrix to assume real values sampled from an unit sphere having size $n \times n$ and relaxing the rank constraint to obtain the relaxed SDP formulation:

$$\begin{aligned} \max_{V} \quad & \text{trace}(CV) \\ \text{subject to.} \quad & diag(V) = \mathbf{e} \\ & V \succeq 0 \end{aligned} \tag{6.2}$$

This SDP formulation can be solved in polynomial time using interior point method [75]. However in this formulation it is difficult to handle the semidefinite constraint along with more number of variables and constraints.

2. **Second lifting relaxation:** Instead of eliminating the rank constraint as described in [32], the rank constraint and the positive semidefinite constraints are combined into a single constraint as: $V^2 - nV = 0$. Introducing a redundant constraint $V \; o \; V = E_{\text{one}}$ for meeting the strong duality, the modified SDP formulation is given as:

$$\begin{aligned} \max_{V} \quad & \text{trace}(CV) \\ \text{subject to.} \quad & diag(V) = \mathbf{e} \\ & VoV = E_{\text{one}} \\ & V^2 - nV = 0 \end{aligned} \tag{6.3}$$

Where E_{one} is matrix of all ones and o denotes element wise multiplication of two given matrices. Second lifting technique introduces more number of constraints and hence is not fully scalable. However, this technique help to achieve the objective function value that is closer to the original objective function value [3].

3. **Constraint Reduction:** This is proposed in [23]. The key idea in reducing constraints is to replace the constraint $diag(V) = \mathbf{e}$ with reduced set of constraints. This approach to scale SDP formulation has been applied to MAX CUT with encouraging results.

4. **Variable Reduction:** A scalable formulation for solving SDP is achieved by reducing the number of variables involved in the SDP formulation [13]. The key idea

in this approach is to **reformulate** the SDP formulation in terms of a nonlinear programming problem by expressing the matrix V as a product of low rank matrices ($V = RR^T$). The resulting nonlinear programming formulation has been applied to MAX CUT problem. The formulations edge in computational time and attaining competing objective function values is demonstrated using experiments.

In the case of rank reduction, one constraint is reduced (namely rank of matrix V), allowing V to take real values. This is a significant step in addressing NP-complete problem. In case of second lifting method, number of constraints are *increased* instead of reducing them. They, however, help in achieving strong duality. From the computational time perspective, second lifting method fails to address scalability. The variable reduction method is a promising one in addressing scalability. In this chapter and in the subsequent chapter, the variable reduction and the constraint reduction approaches are extended to the CC formulation.

6.2 Scalable CC Formulation – SSDP-CC

A scalable formulation for SDP-CC is presented in this section by reducing the number of variables involved. A new constraint is introduced to the SDP-CC formulation which involves reducing the rank of the matrix V as given below:

$$\begin{array}{ll} \min & \text{trace(C V)} \\ \text{such that} & diag(V) = \mathbf{e} \\ & rank(V) \leq r \\ & V \succeq 0 \end{array} \tag{6.4}$$

To handle the rank constraint $rank(V) \leq r$, the positive semidefinite matrix V with rank r, (last constraint in equation (6.4)), can be factorized into $V = RR^T$ where R is a rectangular matrix of size $n \times r$ and $r \leq n$. The choice of the rank r is guided by the following theorem proposed by Burer [13].

Theorem 6.1. *If the feasible set of SDP-CC formulation contains an extreme point then there exits an optimal solution V^* of rank r such that $\frac{r(r+1)}{2} \leq n$.*

Re-formulate equation (6.4) into a nonlinear programming problem by substituting $V = RR^T$ for the constraints $V \succeq 0$ and $rank(V) \leq r$ to obtain the following formulation.

$$\begin{array}{ll} \min\limits_{R} & \text{trace(C } (RR^T)) \\ \text{such that} & diag(RR^T) = \mathbf{e} \end{array} \tag{6.5}$$

Further, the constraint $diag(RR^T) = \mathbf{e}$ can be eliminated by normalizing the matrix R in the objective function trace(C (RR^T)) as trace$\left(C\frac{RR^T}{norm(R)}\right)$. Thus the low rank nonlinear formulation of SDP-CC is given as:

$$\min_{R} \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\frac{\langle R_{i.}, R_{j.}\rangle}{\|R_{i.}\|\|R_{j.}\|} \tag{6.6}$$

where $R_{i.}$ denotes i^{th} row of matrix R and c_{ij} is $(i,j)^{th}$ element in the C matrix as given in equation (2.6). Henceforth equation (6.6) is referred to as SSDP-CC formulation. SSDP-CC is solved through efficient limited memory BFGS algorithm [49]. The advantages of SSDP-CC over SDP-CC are:

1. The number of variables in SSDP-CC given in equation (6.6) is nr which is much smaller than the number of variables n^2 in SDP-CC given in equation (2.6).
2. Change of variable $V = RR^T$ avoids storage of the dense matrix V whose dimension is $n \times n$.

Time Complexity Analysis:

In SSDP-CC, the objective function, an un-constraint formulation, is solved through limited memory BFGS algorithm. Total time complexity of SSDP-CC including function evaluation is $O(kMnr + n^2)$; where 'M' is the number of vector pairs utilized for the Hessian approximation and the modest values of 'M' is between 3 and 20 and k is the number of steps in limited memory BFGS.

6.2.1 Objective Function Evaluation

In the MAX CUT, the objective is to find a set of vertices S that maximizes the weight of the edges in the cut (S, V-S). The cut weight is given by $\sum_{i\in S,\, j\in(V-S)} w_{ij}$. The randomized approximation algorithm for the MAX CUT is the first term of the equation (2.6) [32]. The second term of the equation (2.6) corresponds to maximizing the number of positively correlated edges that lie within each cluster. Both MAX CUT and CC have similar SDP formulations and they differ in the coefficient matrix C. These differences are depicted in equation (6.7).

$$C = \frac{1}{4}\left\{(W_{out} - Diag(W_{out}\mathbf{e})) + (-W_{in} - Diag(W_{in}\mathbf{e}))\right\}. \tag{6.7}$$

Hence only the coefficient matrix C changes even in the nonlinear programming formulation given in equation (6.6). The objective function given in equation (6.6) is

evaluated at every step of the low rank limited memory BFGS algorithm. The value of c_{ij} is computed according to the C matrix discussed above.

6.2.2 Gradient Evaluation

Gradient of the unconstrained objective function (6.6) is computed as $2C\{\mathbf{u}_i\mathbf{u}_i^T\}_{i=1}^n R$, where $\mathbf{u}_i$ is a unit vector with 1 in i^{th} position and 0 elsewhere. As R depends on the rank r, gradient computations effects the choice of r. As the constraint gradients $\{2C\mathbf{u}_i\mathbf{u}_i^T R\}$ are independent for all R $\in \mathbb{R}^{n\times r}$, one can play with the free parameter that is rank r.

6.2.3 Variable Transformation

The matrix V in equation (6.4) is replaced with RR^T and plays a pivotal role in reducing the number of variables in the SDP formulation. The variable reduction not only plays a vital role in scalability but also has significant impact on the storage requirements. One can control these factors by tuning the free parameter rank of the matrix V.

6.2.4 Initialization

In the limited memory BFGS, initialization of the matrix R plays an important role. The matrix R is initialized according to the following equation where R_{ij}^0 refers to the i^{th} row j^{th} column of matrix R in the initialization phase of low rank BFGS algorithm.

$$R_{ij}^0 = \begin{cases} 1 + \frac{1}{n} & \text{if } i \geq j \text{ and } j = r \\ \frac{1}{n} & \text{otherwise} \end{cases}$$

The choice of the above initialization is to avoid sparsity in the matrix R.

6.2.5 Termination Condition

Difference between objective function values in two successive iterations guides the limited memory BFGS algorithm in every iteration. The algorithm is terminated if the difference between two successive iterations is found to be very small. This is expressed in terms of the following condition:

$$\frac{f_i^* - f_{i+1}^*}{max(|f_{i+1}^*|, 1)} \leq 10^{-5}$$

where f_i^* denotes the objective function value at i^{th} iteration and f_{i+1}^* refer to objective function value at $(i+1)^{th}$ iteration.

6.3 Computational Study of SSDP-CC

Two classes of datasets are considered; one in which dataset is available in the form of real vectors and second in which graphs are available in the form of adjacency matrix. In the first case, one needs to construct a graph from real vector data and arrive at edge labeling. In the second case, one needs to arrive at only edge labeling.

1. **Vector data** – *Synthetic dataset and real world dataset :* These datasets are described in chapter 3 and chapter 4.

2. **Graph data** – *Graph Partitioning Archive:* A repository of graphs from diverse real world applications is considered [65]. A total of 6 datasets are considered namely `add20, add32, data, UK, bcsstk33` and `bcsstk29`. Of the considered datasets, `bcsstk29` is a very large dataset containing maximum number of vertices and maximum number of edges. Description of these 6 datasets is given in Table 6.1.

6.3.1 Graph Construction

1. **Vector Data:** Similarity based graph construction, namely $\epsilon-$ nearest neighborhood graph is employed on the vector dataset as described in 4.1.

2. **Graph Data:** As the undirected graph is already available, weights on edges of a given pair of vertices is computed based on the Jaccard index which measures the similarity of two vertices based on the neighborhood information and is given by $J(v_i, v_j) = \frac{|V_i \cap V_j|}{|V_i \cup V_j|}$; where V_i and V_j represent neighbors of vertex v_i and v_j respectively.

6.3.2 Edge Labels

In the case of benchmark graph datasets, edge labels are obtained for these datasets using the Jaccard index in which an edge is considered to be positive when the Jaccard index is greater than a specified threshold; Otherwise the edge label is considered to be negative. The weight on each edge is equal to the Jaccard value between the two vertices in question.

6.3.3 Experimental Setup

The SSDP-CC formulation is empirically evaluated using the above datasets. This is the first attempt in understanding the merit of CC on a large scale benchmark graph datasets. The SSDP-CC formulation is solved through the limited memory BFGS algorithm [49].

The key parameter involved in the SSDP-CC is the rank (number of variables involved in the SDP-CC). To understand the role of rank on the cluster quality and time for clustering, the rank is varied from $0.1 \times (\sqrt{2n}+1)$ to $(\sqrt{2n}+1)$ in steps of 0.1.

Table 6.1: Graph Dataset

Dataset	nodes	edges
add20	2395	2866815
data	2851	4062675
UK	4824	11633076
add32	4960	12298320
bcsstk33	8733	38128278
bcsstk29	13992	97880288

The stopping criteria for the SSDP-CC is the limit on the difference between the two objective function values in successive iterations as discussed in section 6.2.5. The complete SSDP-CC (or SDP-CC) algorithm is given below:

1. Construct the graph along with weights and edge labels.
2. Solve SDP-CC formulation using equation (2.6) or SSDP-CC formulation using equation (6.6) to obtain soft clusters in the form of $\mathbf{v}_1^*, \mathbf{v}_2^*, \cdots, \mathbf{v}_n^*$.
3. Obtain hard clusters by employing random hyperplane rounding technique.
4. Evaluate the external quality of the obtained clusters using rand index measure.

6.4 Results of Scalable CC

The SSDP-CC is implemented using the SDPLR C library [13]. Objective function and gradient computations are modified in accordance with the SSDP-CC formulation as given in equation (6.7). The obtained results are compared with (i) SDP-CC formulation and (ii) Constrained spectral clustering method [78] which is discussed in section 5.3.

Rand Index Measure: For measuring the quality of the obtained clusters using SDP-CC and SSDP-CC Rand Index is considered [64].

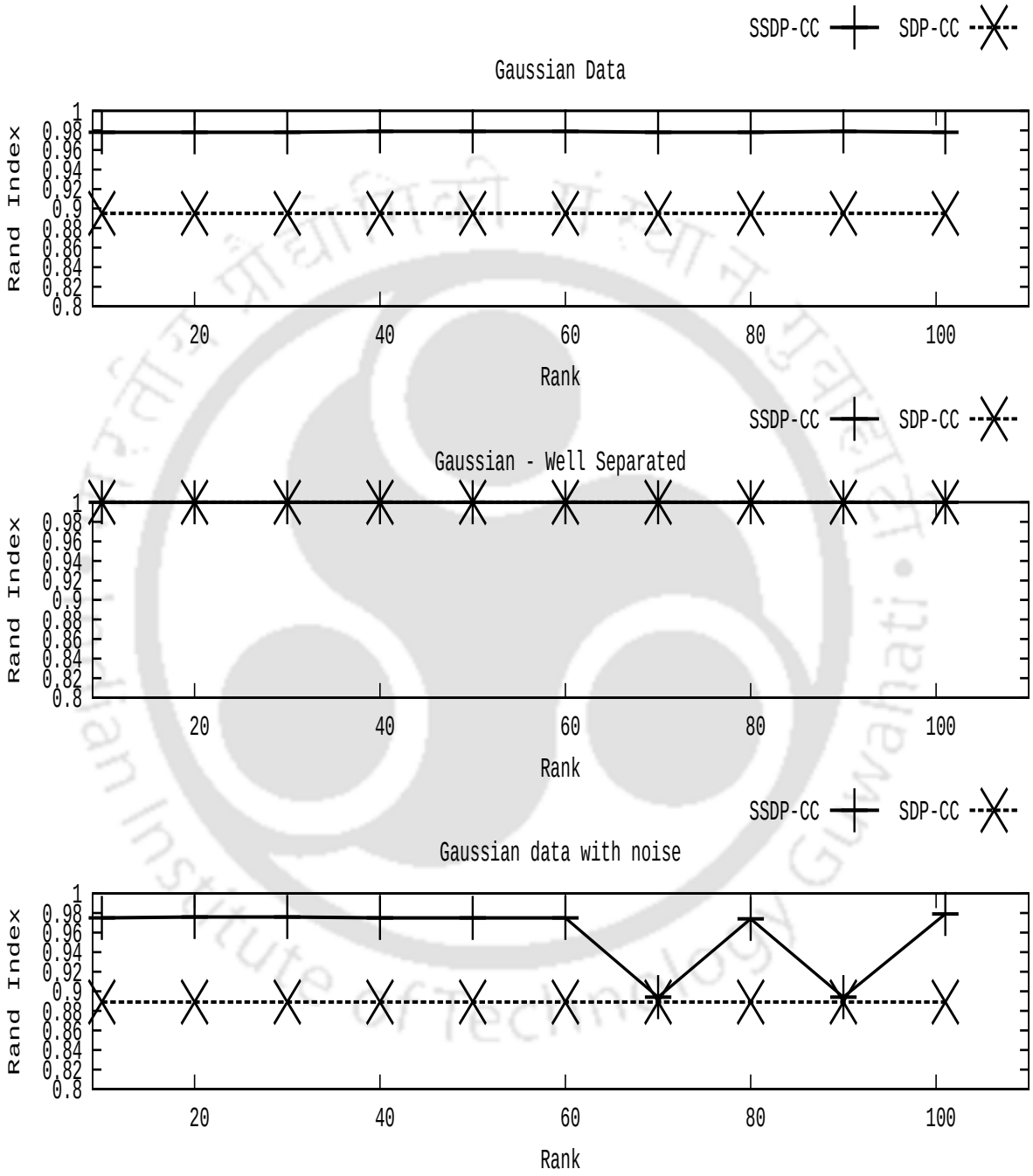

Figure 6.1: Cluster Quality Comparison of SSDP-CC with SDP-CC

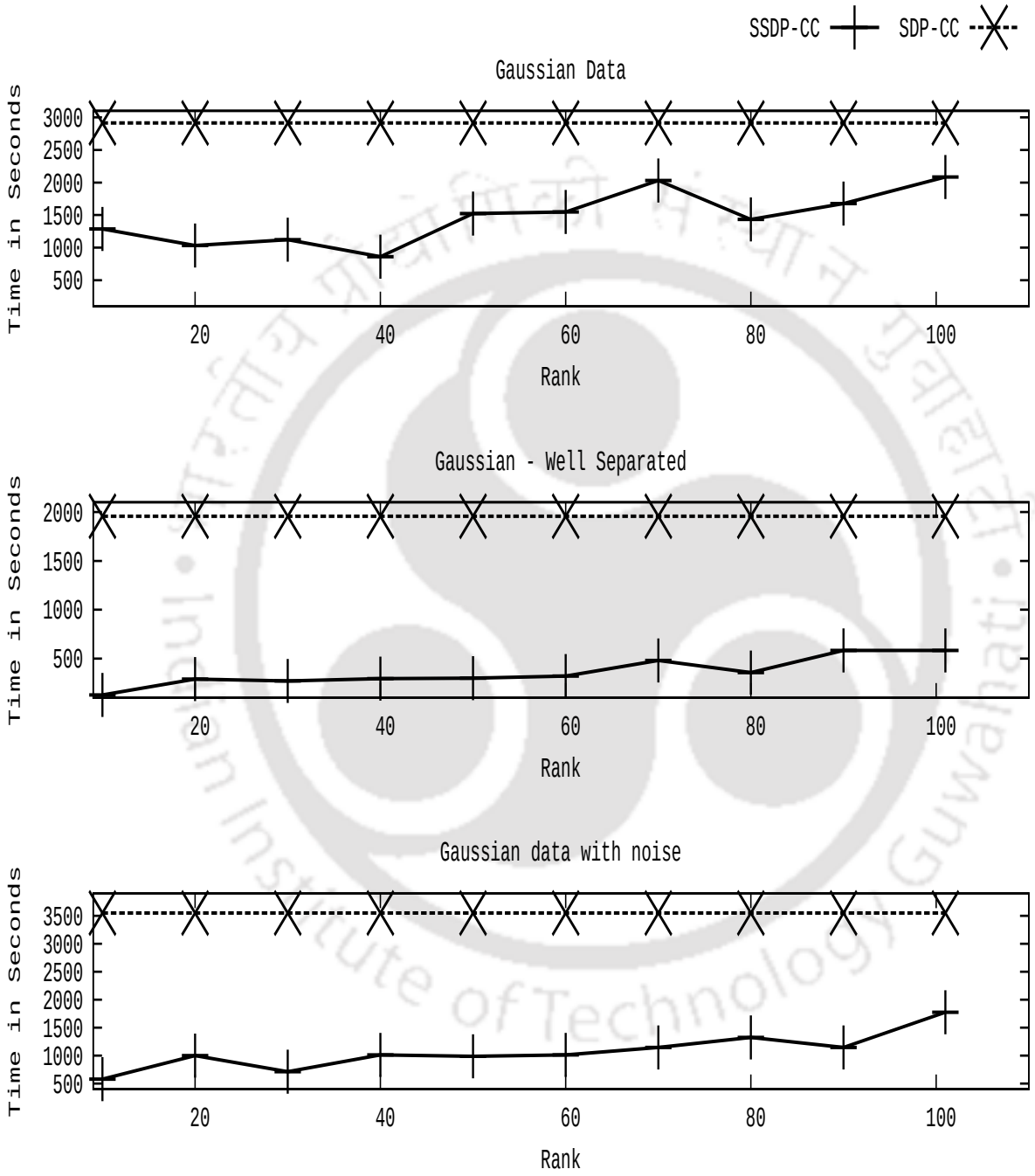

Figure 6.2: Time Comparison of SSDP-CC with SDP-CC

6.4.1 SSDP-CC and SDP-CC - Rand Index Comparison

1. **Synthetic Data Set:** Figure 6.1 shows comparison between SSDP-CC and SDP-CC on three synthetic datasets. In the case of well separated Gaussian dataset, both SSDP-CC and SDP-CC obtain well separated clusters. It is observed that even though the rank of V is reduced to as low as 10, quality of the obtained clusters has not degraded. In case of overlapped Gaussian dataset without-noise and with-noise, SSDP-CC not only retains quality but also outperforms the SDP-CC formulation. The quality of the obtained clusters are sensitive to the rank constraint in the Gaussian noise dataset. In other two datasets, by reducing the rank to as low as 10, quality of the obtained clusters are retained. This is potentially due to the fact that objective function value in case of SSDP-CC is very close to that of the SDP-CC formulation.

2. **Real World Datasets:** In case of real world datasets, SSDP-CC formulation has similar rand index values as that of SDP-CC formulation for both `Pendigit` dataset and `Vehicle` dataset. In case of `Yeast` dataset for the specific classes in consideration, SSDP-CC outperforms SDP-CC formulation. Reducing rank has an effect on the Forbenius norm of the obtained solution matrix which in turn reduces the norm under low rank conditions. We therefore believe that the regularization effect is at play. The rand index results on the UCI datasets considered are depicted in Figure 6.3.

3. **Graph Datasets:** In the case of benchmark graph datasets, SSDP-CC is observed to compete with SDP-CC on the quality perspective. Note that there exists no direct relation between rank of the matrix V and the Rand Index. For example, in case of `add20` graph dataset, for a very low rank (14), Rand Index for SSDP-CC is higher than that of SDP-CC given in Figure 6.5. In case of `add32` however for a very low rank (7), the rand index of SDP-CC is better than that of SSDP-CC. A similar behavior can be observed from the Figure 6.7.

6.4.2 SSDP-CC and SDP-CC - Time Comparison

In the case of SDP-CC, the time complexity for solving SDP-CC is $O(n^{4.5})$. For synthetic datasets n assumes a value of 5000. Where as for solving SSDP-CC the time complexity is $O(kMnr + n^2)$; where the value of $M = 5$, rank r is varied from 7 to 100 and $n = 5000$. Note that as k, M, r are constants, time taken to solve SSDP-CC formulations depends

quadratically on the number of data points. Hence one can obtain significant gains in time for solving SSDP-CC. For real world datasets, rank is varied from 7 to 60, $M = 5$ and n varies between 946 to 3498 (refer to Table 3.4). In the case of Graph datasets the value of M and r are assumed to be same as that of the real world datasets. In graph datasets, n varies from 2395 to 13992. Significant gain in running times is observed due to the quadratic nature of the time complexity in terms of number of variables involved in solving SSDP-CC. The results are explained as given below:

1. **Synthetic Dataset:** Gain in computational time in case of SSDP-CC is as high as 15 times compared to SDP-CC in the best case for well separated Gaussian dataset with rank of 10. This result is depicted in Figure 6.2.

2. **Real World Datasets:** In real world datasets as well SSDP-CC takes less time compared to SDP-CC formulation as depicted in Figure 6.4.

3. **Graph Datasets:** A significant reduction in time can be noted in the case of scalable formulation as depicted in Figure 6.6 and Figure 6.8.

4. **Computational time of SSDP-CC vs Size of data:** A comparison of SSDP-CC computational time with size of the data set is depicted in Figure 6.9 varying the rank from the least rank value to the highest rank value. It can be noted that as the dataset size increases time taken to solve the SSDP-CC formulation increases. However in particular cases where the original rank of the matrix is low, the SSDP-CC formulation takes less time in obtaining the solution as can be observed in the case of synthetic datasets.

6.5 SSDP-CC Comparison with Constrained Spectral Clustering

The proposed scalable formulation of CC is compared with that of the constrained spectral clustering (cSC). As described in chapter 5, the choice of $\gamma = 0$ is motivated from the fact that CC uses labeled edges alone for clustering. Taking $\gamma = 0$ in (5.3) neglects the spectral clustering objective and gives maximum weight to the constraint graph on the data points and thus one can draw reasonable conclusions on both the constrained clustering methods namely SDP-CC/SSDP-CC and cSC.

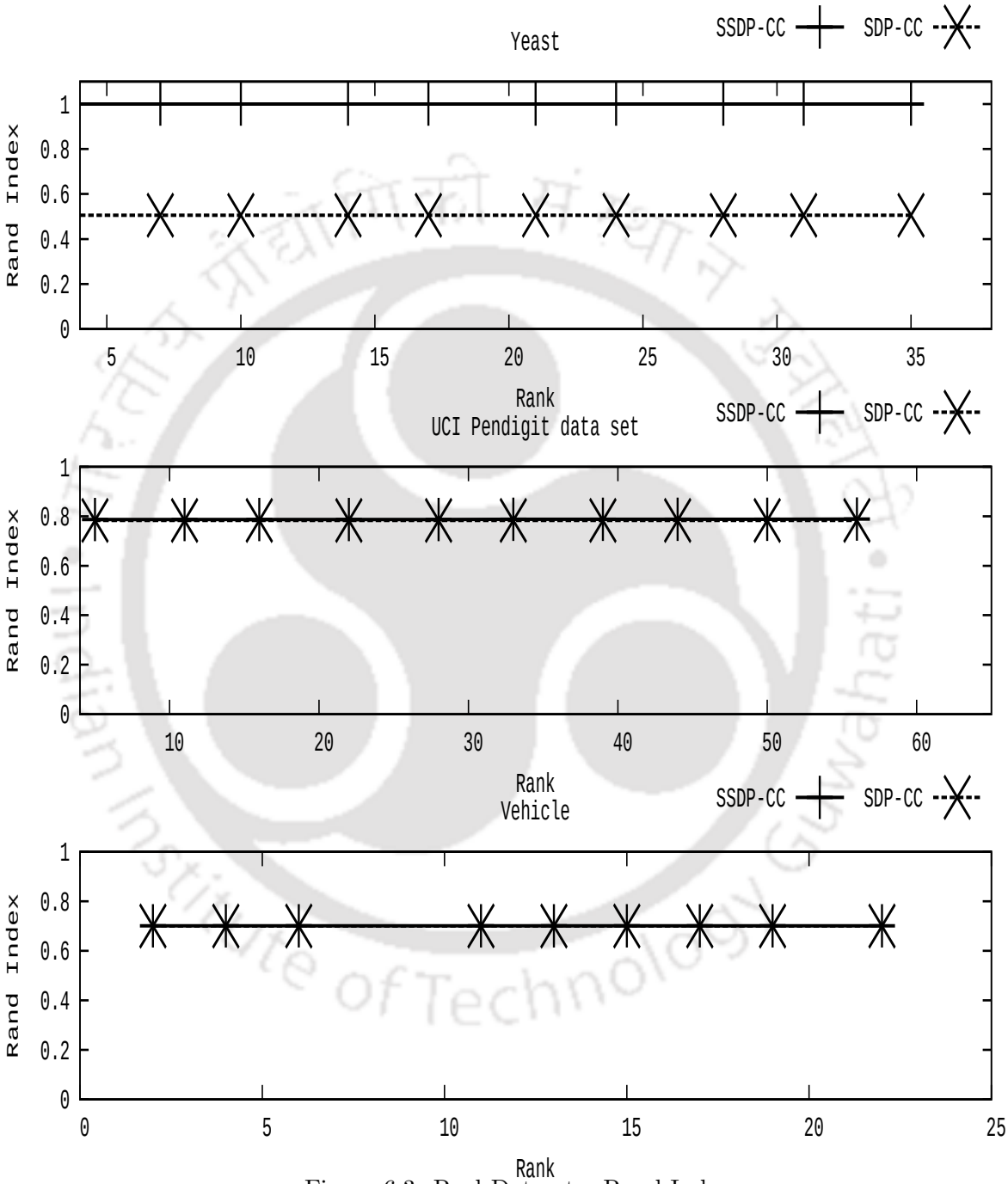

Figure 6.3: Real Datasets: Rand Index

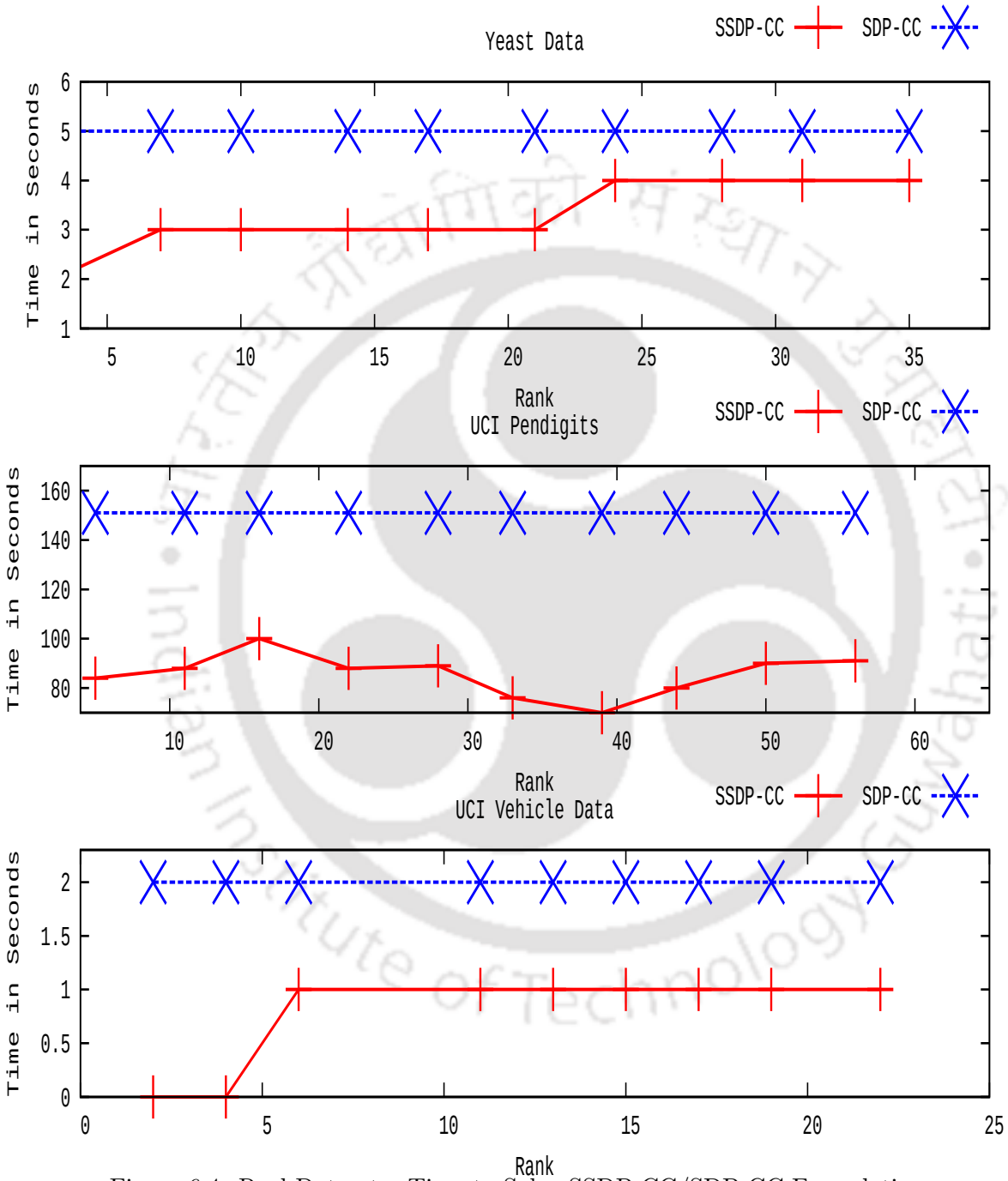

Figure 6.4: Real Datasets: Time to Solve SSDP-CC/SDP-CC Formulation

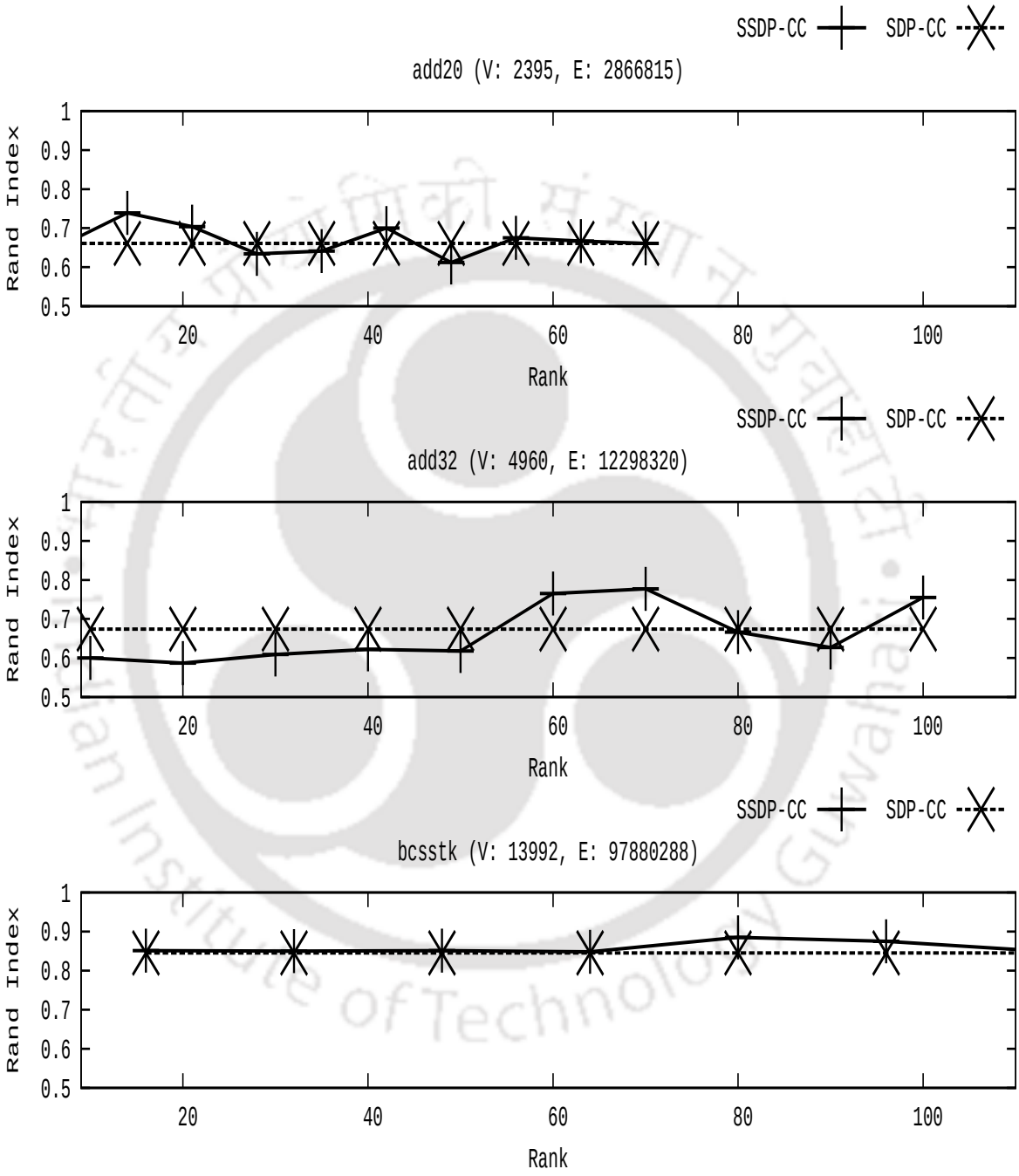

Figure 6.5: Graph Datasets (`add20`, `add32` and `bcsstk29`): Rand Index

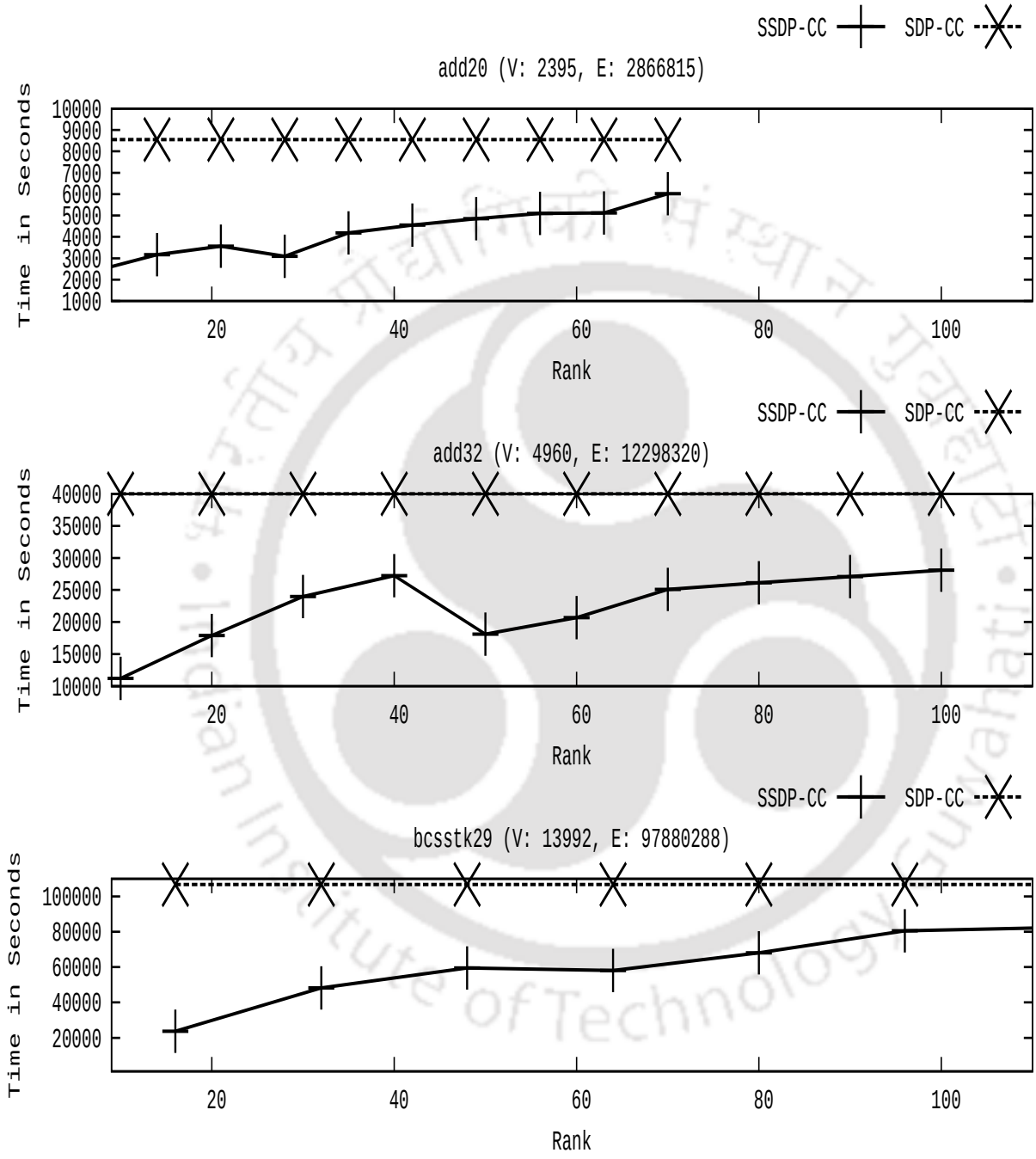

Figure 6.6: Graph Datasets (`add20`, `add32` and `bcsstk29`): Time to Solve SSDP-CC/SDP-CC Formulations

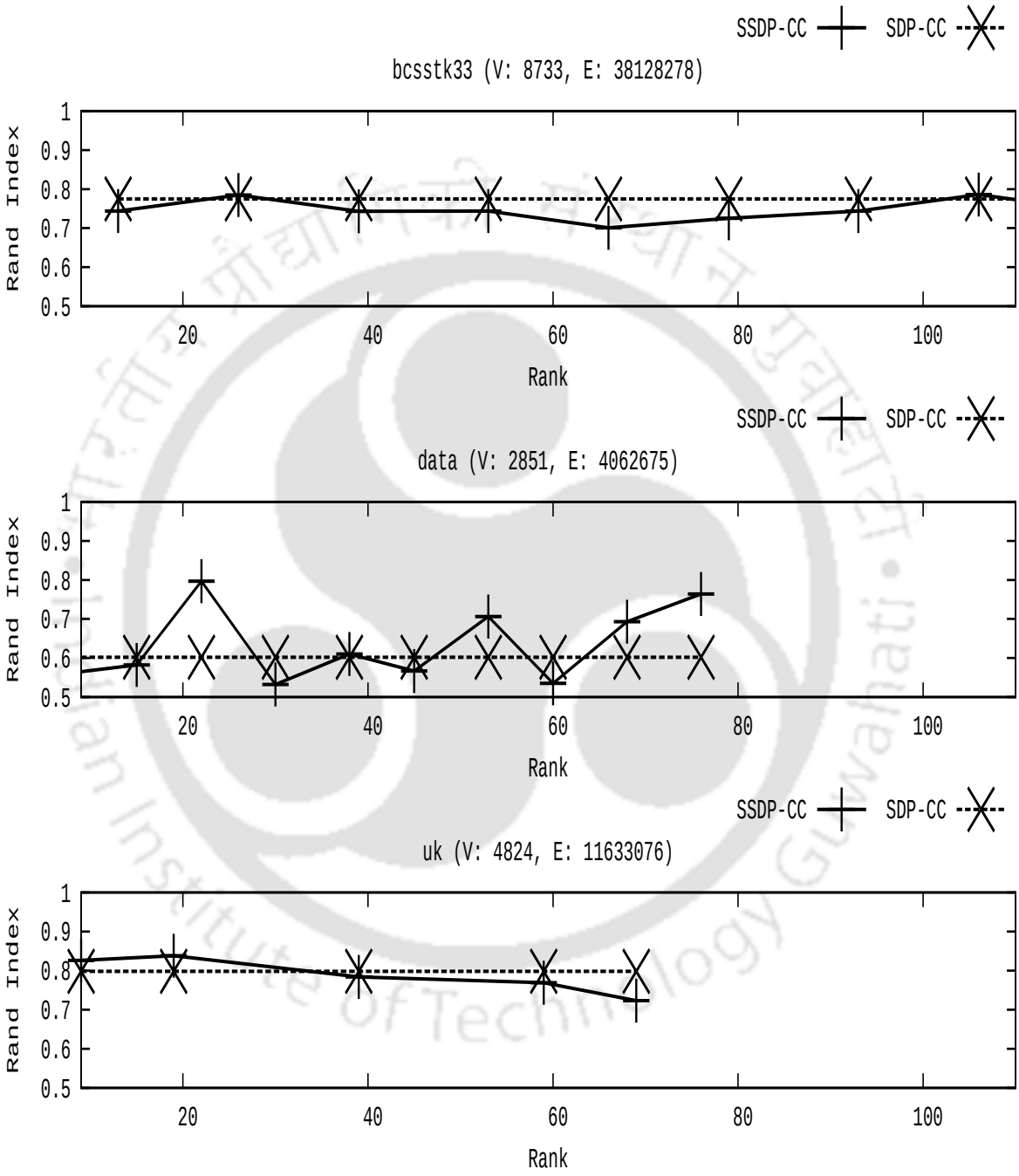

Figure 6.7: Graph Datasets (`bcsstk33`, `data` and `UK`): Rand Index

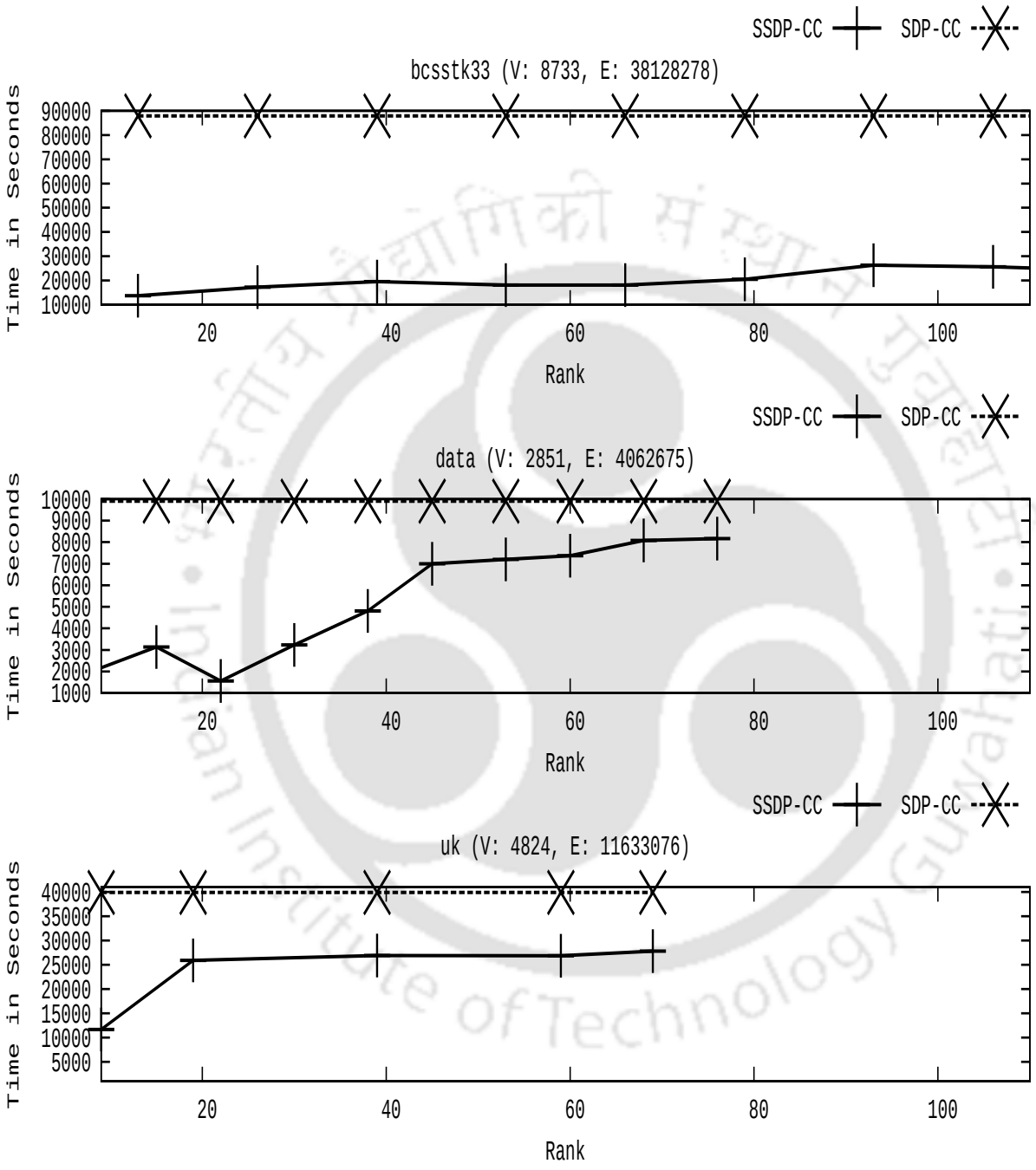

Figure 6.8: Graph Datasets (`bcsstk33, data` and `UK`): Time to Solve SSDP-CC/SDP-CC Formulations

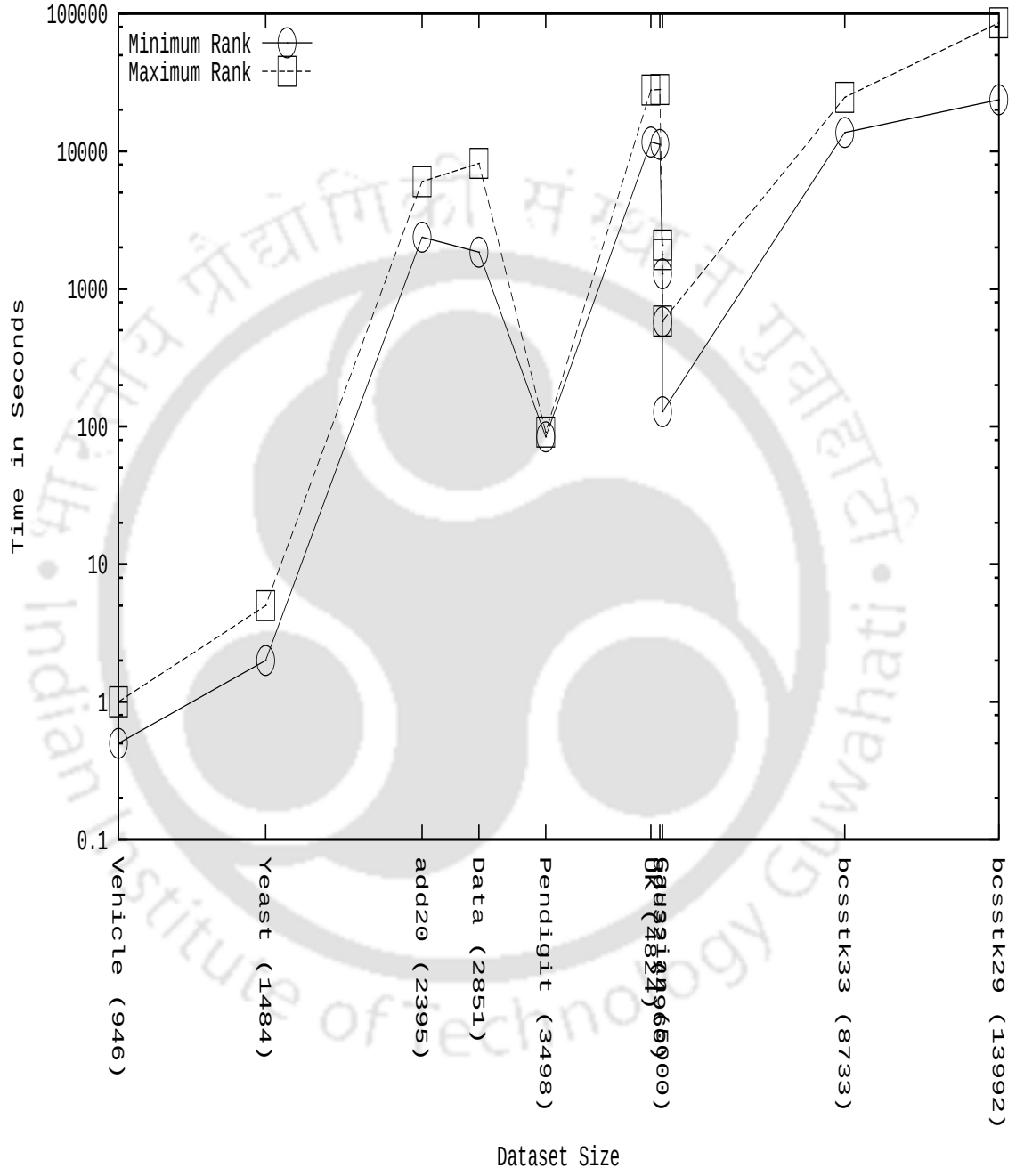

Figure 6.9: (Dataset Size vs Time Taken by SSDP-CC): Time to Solve SSDP-CC/Size of data

6.5.1 SSDP-CC and cSC - Rand Index Comparison

1. **Synthetic Datasets:** Figure 6.10 shows comparison of the SSDP-CC with cSC method on synthetic datasets. In case of well separated Gaussians data, both cSC and SSDP-CC perform well and obtains clusters that match the true labels. In case of overlapping Gaussians without-noise and with-noise, the SSDP-CC outperforms cSC. This result indicates that CC as a method of constraint clustering has a merit compared to other forms of constraint clustering when edge labels *alone* are utilized. This result can potentially change when one play with the parameter γ in the cSC method.

2. **Real World Datasets:** SSDP-CC has significant edge over cSC from the external quality perspective on all the three real world datasets. The results are depicted in the Figure 6.11.

3. **Graph Datasets:** When comparing the SSDP-CC to cSC on the benchmark graph datasets, SSDP-CC formulation is observed to outperform the later formulation in Rand Index as depicted in the Figure 6.12.

6.5.2 SSDP-CC and cSC - Time Comparison

Time complexity of the constraint spectral clustering (cSC) is same as the time complexity of the spectral clustering, known as $O(kn^2)$, where k is the number of steps in eigenvalue calculation; where as for solving SSDP-CC the time complexity is $O(kMnr+n^2)$ with k, M and r problem specific constants. The extra terms in the SSDP-CC time complexity play a role in increasing the overall running time as compared to cSC formulation. However considering quality index, one can afford the increase in the computational cost.

1. **Synthetic Datasets:** The time taken for the cSC is significantly lower compared to scalable CC formulation. The time comparison between SSDP-CC and cSC formulation is shown in Figure 6.13. From this figure, observe that cSC has an clear edge over the SSDP-CC formulation.

2. **Real World Datasets, Graph Datasets:** A similar trend in running times can be noted in both the real world and graph datasets. As the number of vertices increases, cSC outperforms scalable CC in terms of time. The time (in seconds) to obtain clusters using cSC and scalable CC are shown in Figures 6.14 and 6.15.

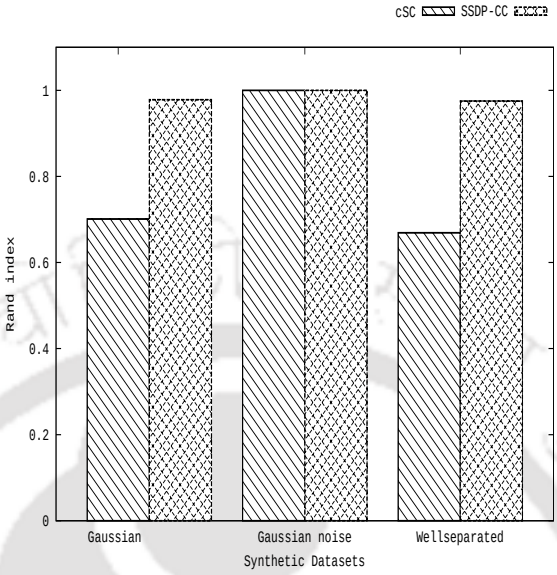

Figure 6.10: Synthetic Datasets: SSDP-CC Vs Constrained Spectral Clustering

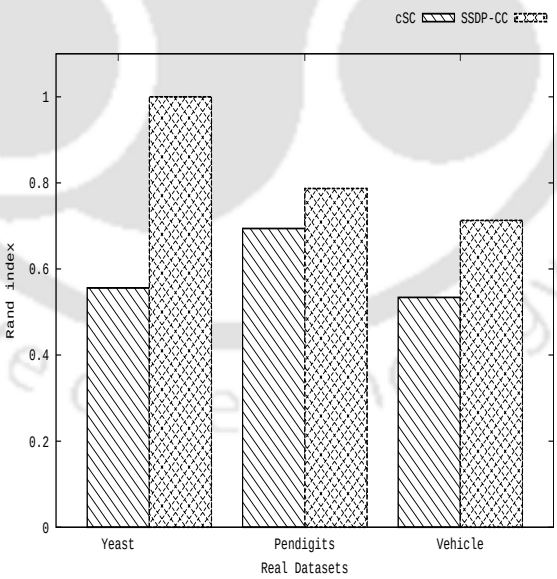

Figure 6.11: Real Datasets: SSDP-CC Vs Constrained Spectral Clustering

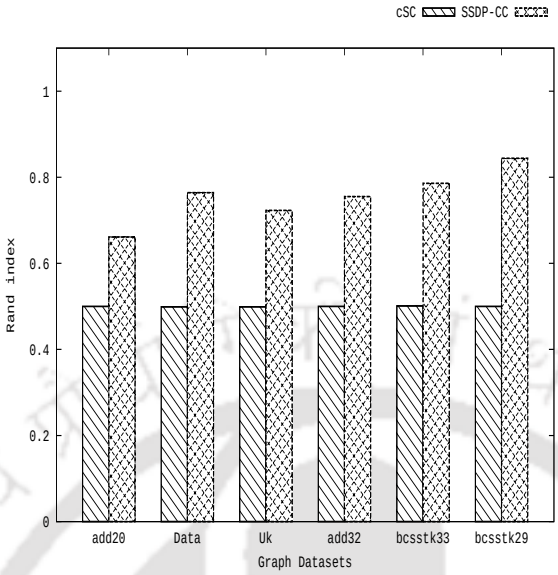

Figure 6.12: Graph Datasets: SSDP-CC Vs Constrained Spectral Clustering

6.6 Summary

In this chapter, the scalable MAX CUT low rank factorization is extended to CC. This method is noted to be a promising approach for scaling CC for large datasets. When chosen an appropriate rank for the low rank factorization, the proposed method yields better external quality index than the original formulation with reduction in time and memory requirement. When compared with cSC as a special case of involving only edge labels, CC exhibited edge over cSC. The community detection problem defines a community in terms of cluster. Therefore clustering methods' application is predominant in these problem domains. CC can potentially be used for community detection whose objective function is close to that of modularity maximization [1].

As described in section 6.1, scalability of CC is also achieved by reducing the number of constraints. Next chapter's contribution is towards reducing the number of constraints for achieving scalability of CC.

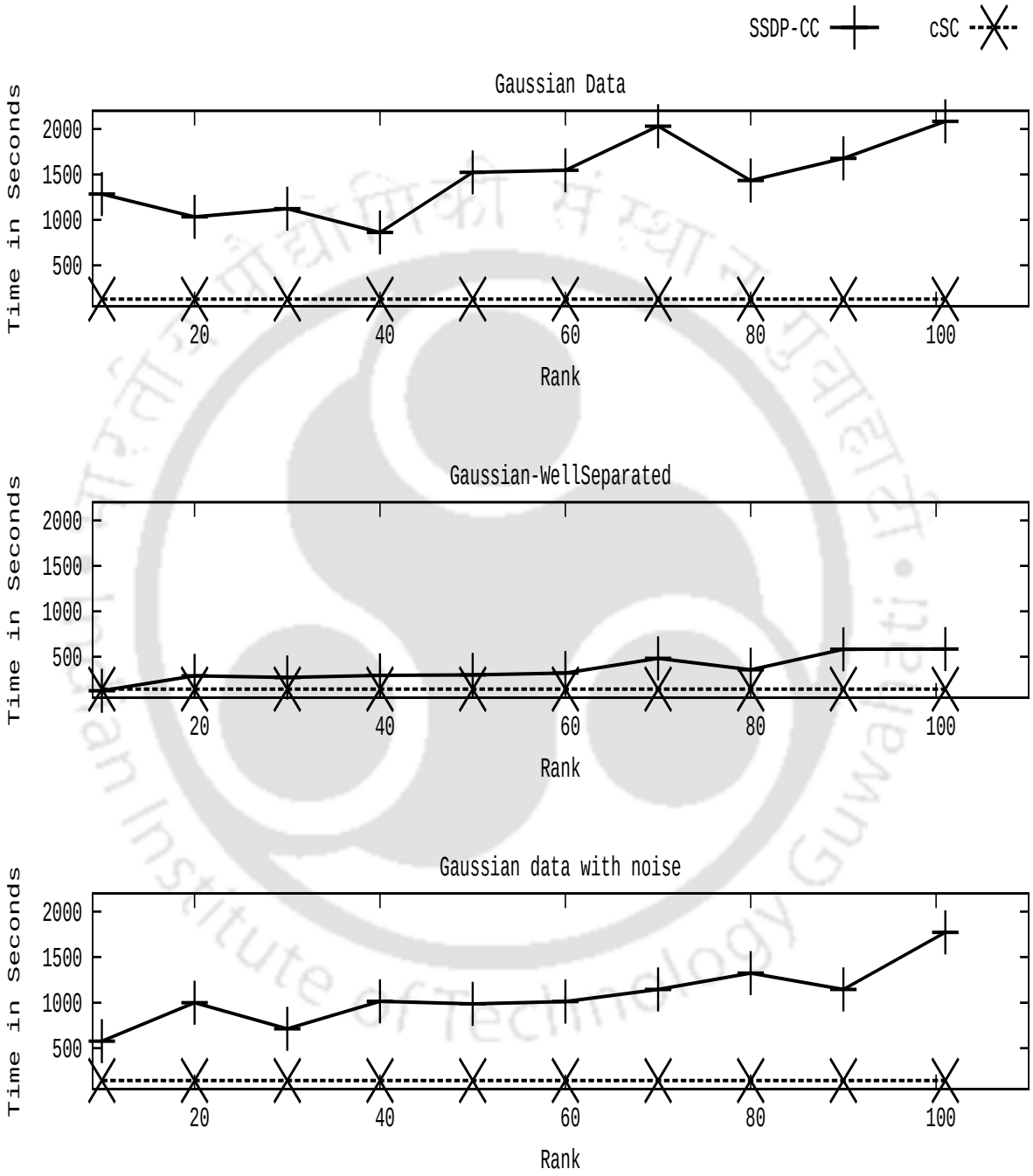

Figure 6.13: Time Comparison of SSDP-CC with cSC Formulation

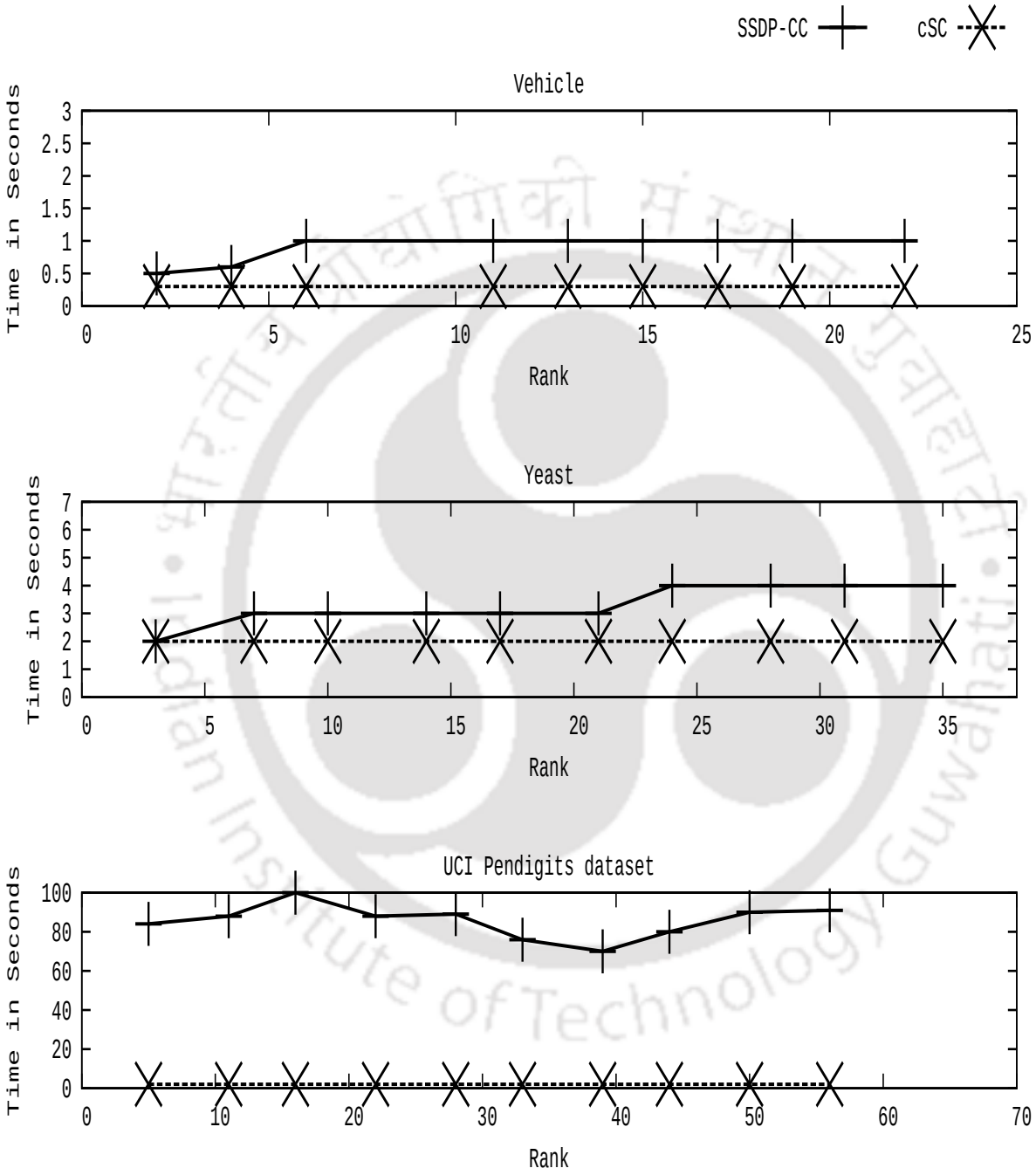

Figure 6.14: Real datasets: Time to Solve SSDP-CC and cSC Formulation

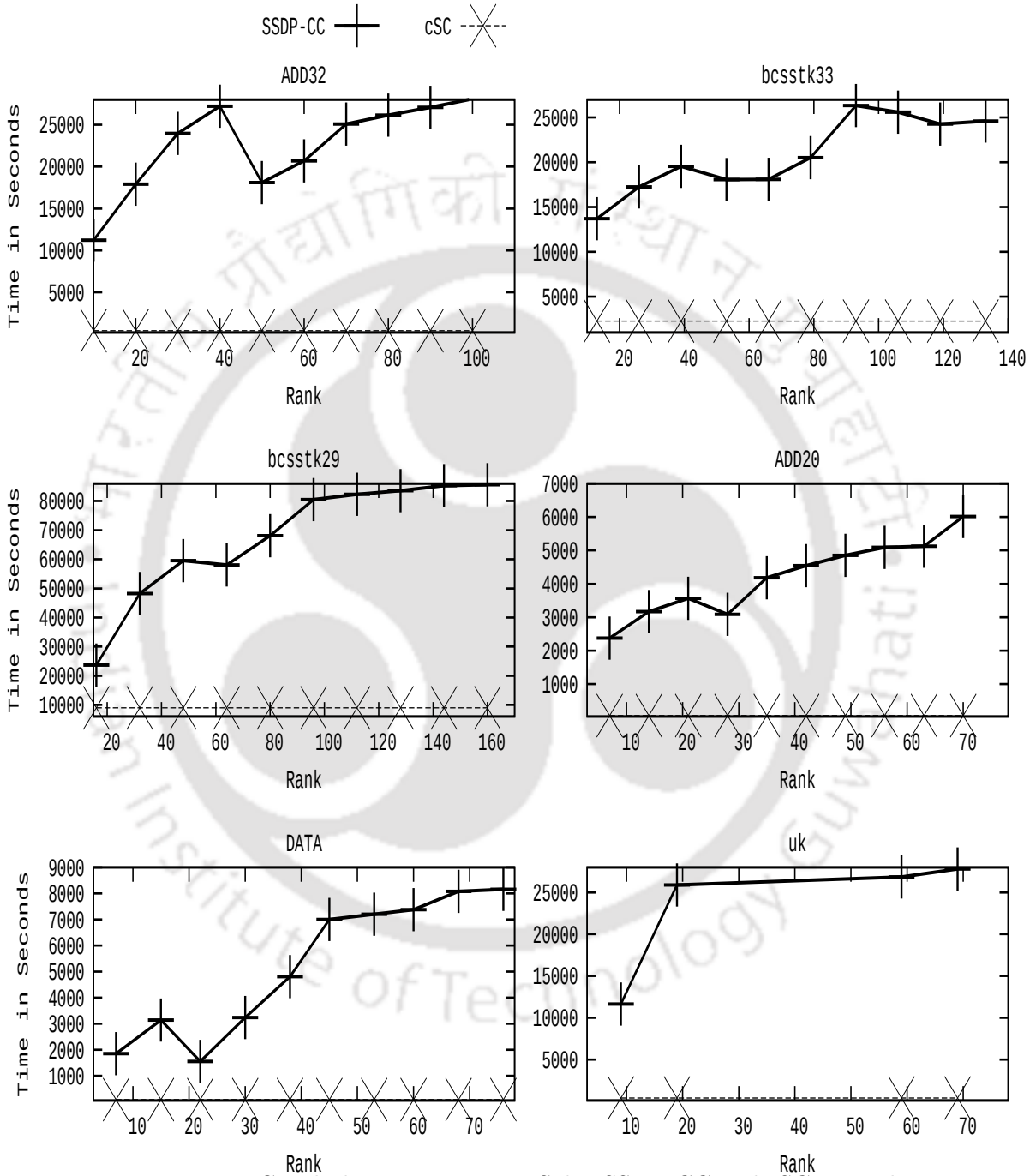

Figure 6.15: Graph datasets: Time to Solve SSDP-CC and cSC Formulation

Chapter 7

Scalability of CC Through Constraint Reduction

In chapter 6, a scalable CC formulation is discussed which reduces number of variables involved in the SDP formulation. This chapter presents a scalable solution for the SDP formulation of CC (SDP-CC) by reducing the number of constraints. Experiments are performed to demonstrate the merit of constraint reduction along:

1. Time required to solve the SDP in the light of large number of data points (or vertices).
2. The objective function value achieved through the relaxed formulation (which is empirically shown to be close to the original SDP formulation).

The key contributions in this chapter are:

1. Address scalability of by reducing the number of constraints in SDP-CC.
2. Apply CC in practice for **large datasets**.
3. Analyze CC's external quality on variety of datasets.
4. Compare the proposed scalable formulation with another well known variant in which scalability is achieved by reducing the number of variables involved in the SDP formulations as discussed in chapter 6.
5. Compare the CC's cluster quality with that of constrained spectral clustering method [78].

Experimental results on synthetic, real world datasets whose graph sizes range from 100 vertices to 13000 vertices are tested with both the scalable formulations. Large scale benchmark graph datasets are also tested whose sizes range from 2395 vertices to 13992 vertices. The proposed formulation is shown to have an edge over the original SDP-CC formulation, variable reduction variant of SDP-CC and a constraint clustering method, namely constrained spectral clustering.

7.1 Proposed Formulation

A fairly recent effort by [23] in achieving scalability is through introducing cascade of fast SDP relaxations. In this approach, number of constraints involved in the SDP formulation are reduced significantly. This help achieve speedup in computational time in the light of large number of vertices and edges of a given graph. The number of constraints are reduced by introducing an asymmetric matrix $A \in R^{n \times m}$ in the constraint of CC formulation (2.6):

$$diag(V) = \mathbf{e}.$$

as

$$A^T diag(V) = A^T \mathbf{e}.$$

The number of constraints in $diag(V) = \mathbf{e}$ is equal to n; where as the relaxed one contains as many constraints as the number of columns, $m \leq n$, in the A matrix. Also, this constraint will not be equivalent to the original constraint and hence it is a weaker form of the constraint.

Fast SDP relaxations are introduced in the CC framework for the first time in this work. To introduce these relaxations, the SDP-CC formulation given in equation (2.6) is re-written such that it involves the standard graph Laplacian quantity. Consider the first term in the objective function of equation (2.6):

$$\frac{1}{2} \sum_{(i<j)} w_{ij} \left(1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle\right) \tag{7.1}$$

As the affinity matrix is symmetric, equation (7.1) is re-written as:

$$\begin{aligned}
&\tfrac{1}{4}\sum_{(i,j)} w_{ij}(1-\langle \mathbf{v}_i, \mathbf{v}_j\rangle) \\
&= \tfrac{1}{4}\left\{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\langle \mathbf{v}_i, \mathbf{v}_j\rangle\right\} \\
&= \tfrac{1}{4}\left\{\sum_{i=1}^{n}(W\mathbf{e})_i {\mathbf{v}_i}^2 - \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\mathbf{v}_i^T\mathbf{v}_j\right\} \\
&= \tfrac{1}{4}\left\{\mathbf{v}^T Diag(W\mathbf{e})\mathbf{v} - \mathbf{v}^T W\mathbf{v}\right\} \\
&= \tfrac{1}{4}\mathbf{v}^T\left(Diag(W\mathbf{e}) - W\right)\mathbf{v} \\
&= \tfrac{1}{4}\mathbf{v}^T\left(D - W\right)\mathbf{v}
\end{aligned} \tag{7.2}$$

where $D = Diag(W\mathbf{e})$. Last expression of equation (7.2) is expressed in the form of inner product between (D - W) and V as follows:

$$\begin{aligned}
\tfrac{1}{4}\mathbf{v}^T\left(D - W\right)\mathbf{v} \quad &= \tfrac{1}{4}\text{trace}(\mathbf{v}^T\left(D - W\right)\mathbf{v}) \\
&= \tfrac{1}{4}\text{trace}(\mathbf{v}^T((D - W)\mathbf{v})) \\
&\quad [\because \text{trace(AB)=trace(BA)}] \\
&= \tfrac{1}{4}\text{trace}((D - W)\mathbf{v}\mathbf{v}^T) \\
&= \tfrac{1}{4}\text{trace}((D - W)V) \\
&= \tfrac{1}{4}\text{trace}(V(D - W)) \\
&= \tfrac{1}{4}\langle V, (D - W)\rangle \\
&\quad \text{where } V = \mathbf{v}\mathbf{v}^T
\end{aligned} \tag{7.3}$$

In a similar fashion, the second term of equation (2.6) is expressed a inner product between matrix V and matrix $(D + W)$ as given below:

$$\begin{aligned}
&\tfrac{1}{4}\sum_{(i,j)} w_{ij}(1+\langle \mathbf{v}_i, \mathbf{v}_j\rangle) \\
&= \tfrac{1}{4}\mathbf{v}^T\left(D + W\right)\mathbf{v} \\
&= \tfrac{1}{4}\langle V, (D + W)\rangle
\end{aligned} \tag{7.4}$$

Substituting equations (7.3) and (7.4) in equation (2.6) to obtain modified MAXA-

GREE2 formulation as given below:

$$\begin{aligned} \max_{V} \quad & \tfrac{1}{4}\{\langle V,(D-W)\rangle + \langle V,(D+W)\rangle\} \\ \text{subject to} \quad & V \succeq 0 \\ & diag(V) = \mathbf{e} \end{aligned} \tag{7.5}$$

Introduce a weaker constraint: $A^T diag(V) = A^T\mathbf{e}$ in the place of the constraint: $diag(V) = \mathbf{e}$ giving rise to equation (7.6); where the matrix A is to be constructed according the number of constraints to be reduced which is explained in section 7.1.1.

$$\begin{aligned} \max_{V} \quad & \tfrac{1}{4}\{\langle V,(D-W)\rangle + \langle V,(D+W)\rangle\} \\ \text{subject to} \quad & V \succeq 0 \\ & A^T diag(V) = A^T\mathbf{e} \end{aligned} \tag{7.6}$$

Henceforth the equation (7.6) is referred to as, reduced constraint SDP-CC or RC SDP-CC. The proposed RC SDP-CC is empirically evaluated for its merit. Note that the above formulation do not yet have a theoretical result bounding the objective function value of the relaxed formulation.

7.1.1 Construction of Matrix 'A'

The matrix $A \in \mathbb{R}^{n\times m}$ can be chosen freely with a constraint that A's column space is $\mathbf{e}$. The design specified in [23] is adopted for constructing this matrix. Given a graph consisting of n vertices, choose the number of constraints m to be reduced. Construct the matrix A as follows: initialize every element of A to 0. Divide the vertices into m groups. These m groups represent m columns of matrix A. All the vertices falling into first group are assigned a value 1 and rest of the vertices in this group are assigned a value 0. This process is repeated for all columns of the matrix A. For example let a graph having 6 vertices. To construct a matrix A consisting of 2 groups ($m = 2$), divide the 6 vertices into two groups. Let first three vertices, namely v_1, v_2, v_3 belong to the first group and remaining vertices belong to the second group. Then A takes the following form according to the above description:

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

The role of A on the obtained clusters through CC formulation is experimentally studied. In the experiments, data points are all assigned randomly to each of the m groups.

7.1.2 Time Complexity

Computational complexity to solve the basic SDP formulation in worst case is polynomial time. The worst case time complexity for solving an SDP optimization problem with n variables and n constraints is: $O(n^{4.5})$ [77]. Hence for SDP-CC, time complexity is $O(n^{4.5})$. In case of RC SDP-CC formulation the number of dual variable is reduced to m by introducing weaker constraint described in the above section. Therefore the time complexity for RC SDP-CC is $O(m^2 n^{2.5})$ where $m \leq n$.

7.2 Experimental Evaluation

The datasets employed for experimentation in chapter 6 are used for testing the proposed RC SDP-CC formulation in this section. Please refer to 6.3 for details on the datasets. The RC SDP-CC given in equation (7.6) is implemented using SDPNAL [90] solver. The matrix 'A' is constructed as described in section 7.1.1. Rand Index [64] is considered as a measure of similarity between obtained clusters (C_1) and true clusters (C_2) for measuring quality of the obtained clusters.

The proposed RC SDP-CC formulation's results (external quality, objective function value and time taken to obtain clusters) are compared with that of the original SDP-CC formulation. The RC SDP-CC formulation's results are also compared with that of variable reduction method described in section 6.4 by comparing the rand index value, the objective function value and time taken to obtain clusters.

Figure 7.1 shows comparison between RC SDP-CC, SDP-CC and variable reduction based CC on the Gaussian well separated synthetic dataset. Following are observations are made from the obtained figure:

1. Using RC SDP-CC formulation the external quality, namely rand index is competing with that of the original SDP-CC formulation. This is due to the fact that objective function value in case of RC SDP-CC is very close to that of the SDP-CC formulation. The variable reduction based CC has an edge over RC SDP-CC formulation on the external quality (top most sub-figure in Figure 7.1).

2. The objective function value is equal to or greater than that of original SDP-CC formulation as the number of constrains is reduced in the proposed formulation. In the case of SDP-CC and variable reduction based method, the objective function values go hand in hand.

3. The above two points suggest that objective function value does not have a direct bearing with external quality. Note that the approximation value which stand for the internal quality has a direct relation with external quality.

4. Time taken to obtain clusters significantly reduces as the number of constrains is reduced (middle sub-figure in Figure 7.1) when compared to SDP-CC and variable reduction based method.

5. Number of variables reduced in case of SSDP-CC formulation is significantly less compared to the number of constraints reduced in the proposed case. To compare these two methods, the number of constraints are reduced to match the number of variables. In the Figure 7.1, 155 on x-axis represents number of constraints reduced (equal to number of variables in SSDP-CC formulation). Number of variables reduced is same as the chosen rank. Rank is chosen in SSDP-CC based on the formula $rank = (\sqrt{2n} + 1)$, the value of which varies from $0.1 \times (\sqrt{2n} + 1)$ to $(\sqrt{2n} + 1)$ in steps of 0.1. From this one observes that constraint reduction has an edge over time and objective function value by retaining the external quality.

6. Note that when the constraints are reduced to one tenth in case of RC SDP-CC, the external quality has not been affected drastically. On the other hand, the objective function value improved over SDP-CC and variable reduction methods by gaining on computational time.

7. No explicit relationship is observed between the number of constraints and the external quality. However, a direct relationship can be observed from the computational time perspective and the objective function value. As the number of constraints is reduced, time taken to obtain clusters reduces significantly and objective function value is increased.

A similar observation is made on other synthetic datasets. In case of real world datasets, RC SDP-CC formulation has similar rand index values as that of SDP-CC formulation for all the considered datasets. Figure 7.2 presents results for `Pendigit` dataset for class labels 0 and 1.

1. Using RC SDP-CC formulation the external quality is competing with that of the original SDP-CC formulation. The variable reduction based CC is at par with the RC SDP-CC formulation on the external quality (top most sub-figure in Figure 7.2).

2. The objective function value is equal to or greater than that of original SDP-CC formulation as the number of constrains is reduced in the proposed formulation. In the case of SDP-CC and variable reduction based method, the objective function values go hand in hand.

3. Time taken to obtain clusters significantly reduces as the number of constrains is reduced (middle sub-figure in Figure 7.2) when compared to SDP-CC and variable reduction based method.

4. Number of variables reduced in case of SSDP-CC formulation for `Iris` dataset is 15 and in case of `Pendigit` dataset is 56. In the proposed formulation, number of constraints in `Iris` dataset is reduced to 15 while in `Pendigit` dataset it is reduced to 56. The middle subfigure in Figures 7.2 and 7.3 present results for these special cases. From these figures observe that constraint reduction has an edge over time and objective function value obtained by retaining the external quality.

5. In the case of real world datasets, the external quality and objective function value behavior is similar to that of point 6 of the synthetic dataset results presented earlier.

6. As in the case of synthetic dataset results presented in point 7, there is no explicit relationship is observed between the number of constraints and the external quality.

A similar observation is made on other real world datasets. Figure 7.3 show the results obtained for `Iris` dataset for class labels `Iris Setosa` and `Iris Versicolour`. The objective function value (as number of constraints is reduced) vary due to the fact that `Iris` dataset contains less number of data points.

Figure 7.4 show the results for the proposed formulation on benchmark graph dataset `add20` having 2395 vertices and 2866815 constraints. Observe from this figure that time taken to obtain cluster using constraint reduction is significantly less compared to SDP-CC formulation as well as variable reduction based scalable CC formulation. A 9 times gain is

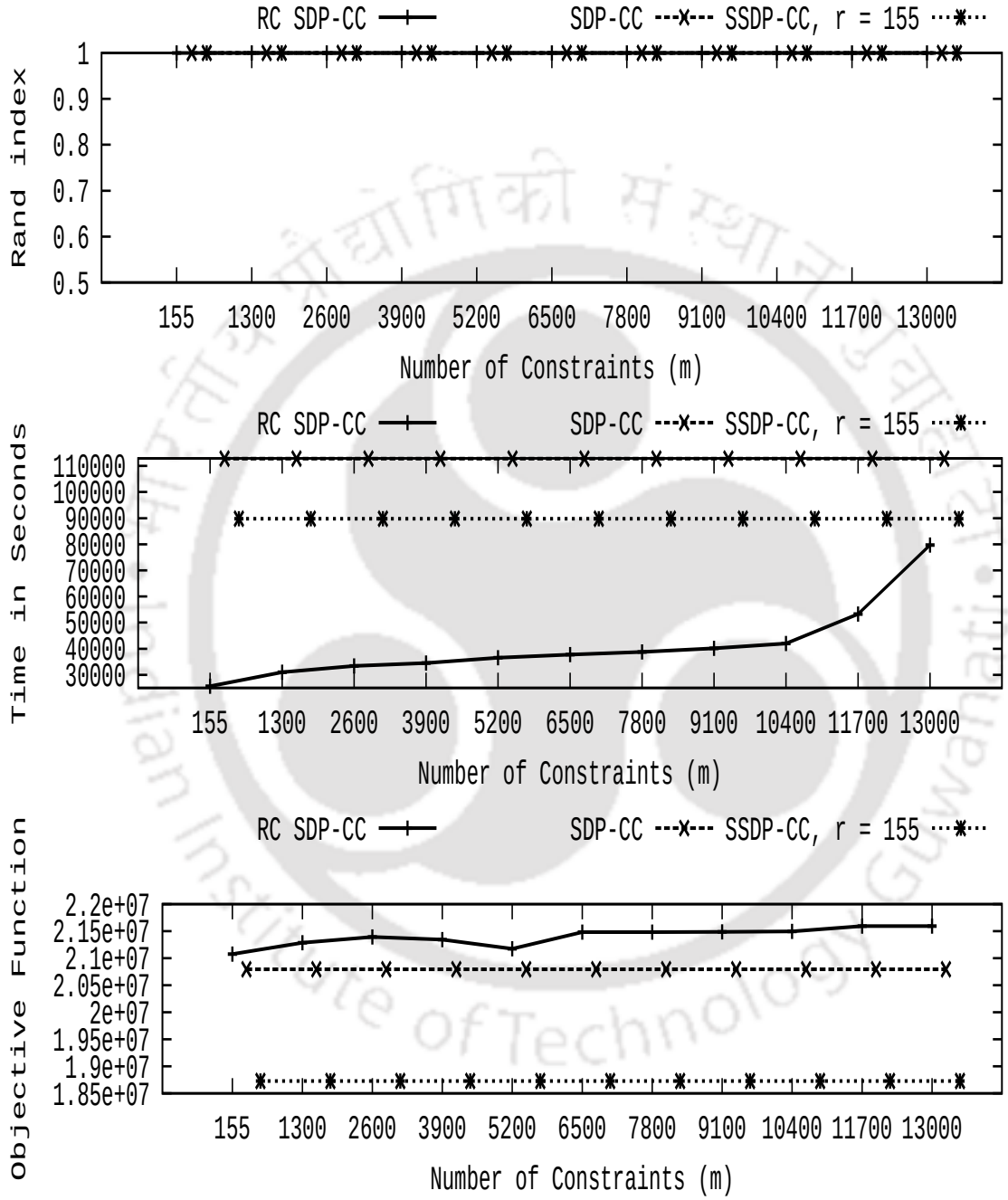

Figure 7.1: Synthetic Well Separated Dataset.

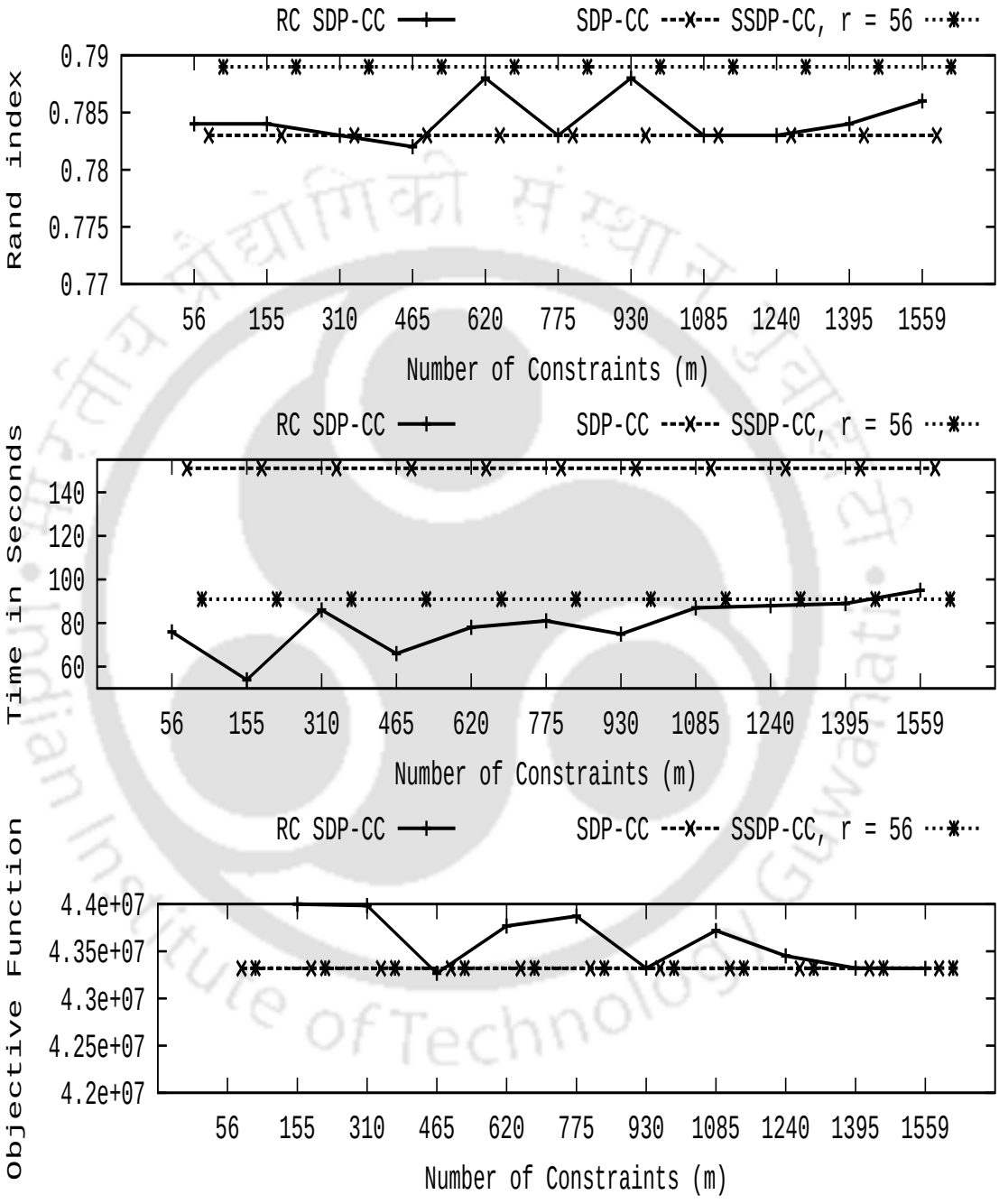

Figure 7.2: Real World Dataset: `Pendigit`.

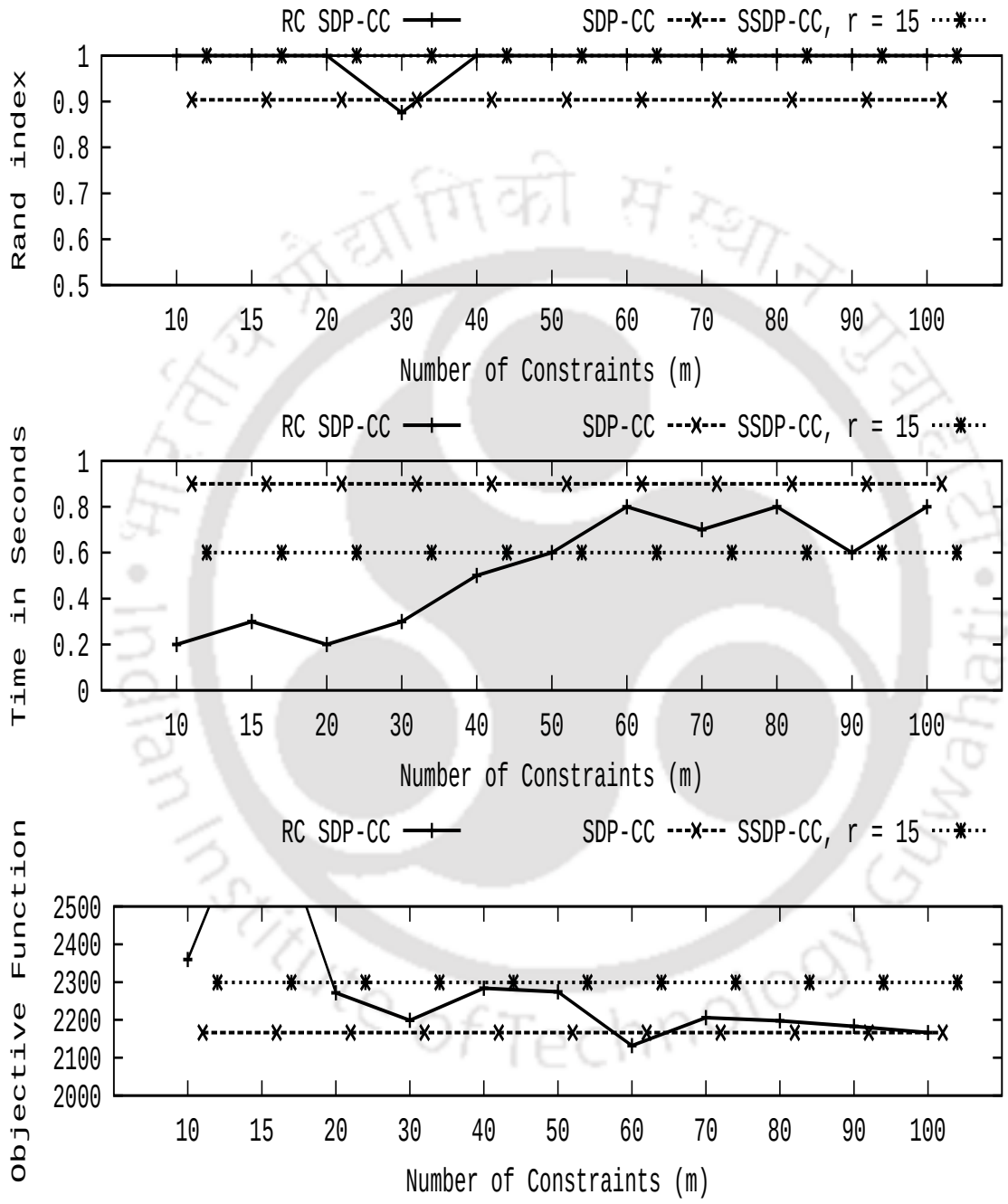

Figure 7.3: Real World Dataset: `Iris`

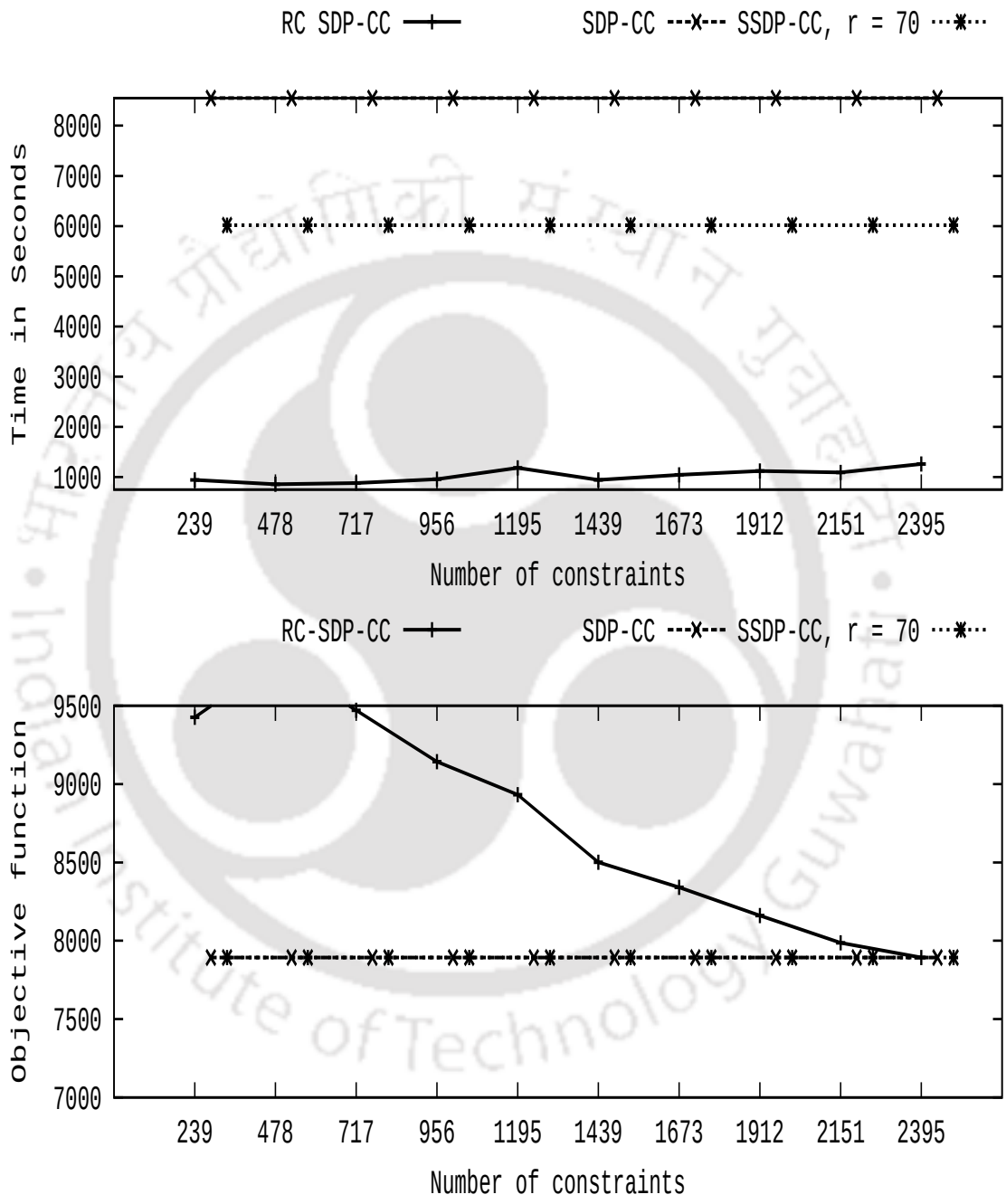

Figure 7.4: Graph Dataset: `add20`. $G = (V_g : 2395, E : 2866815)$.

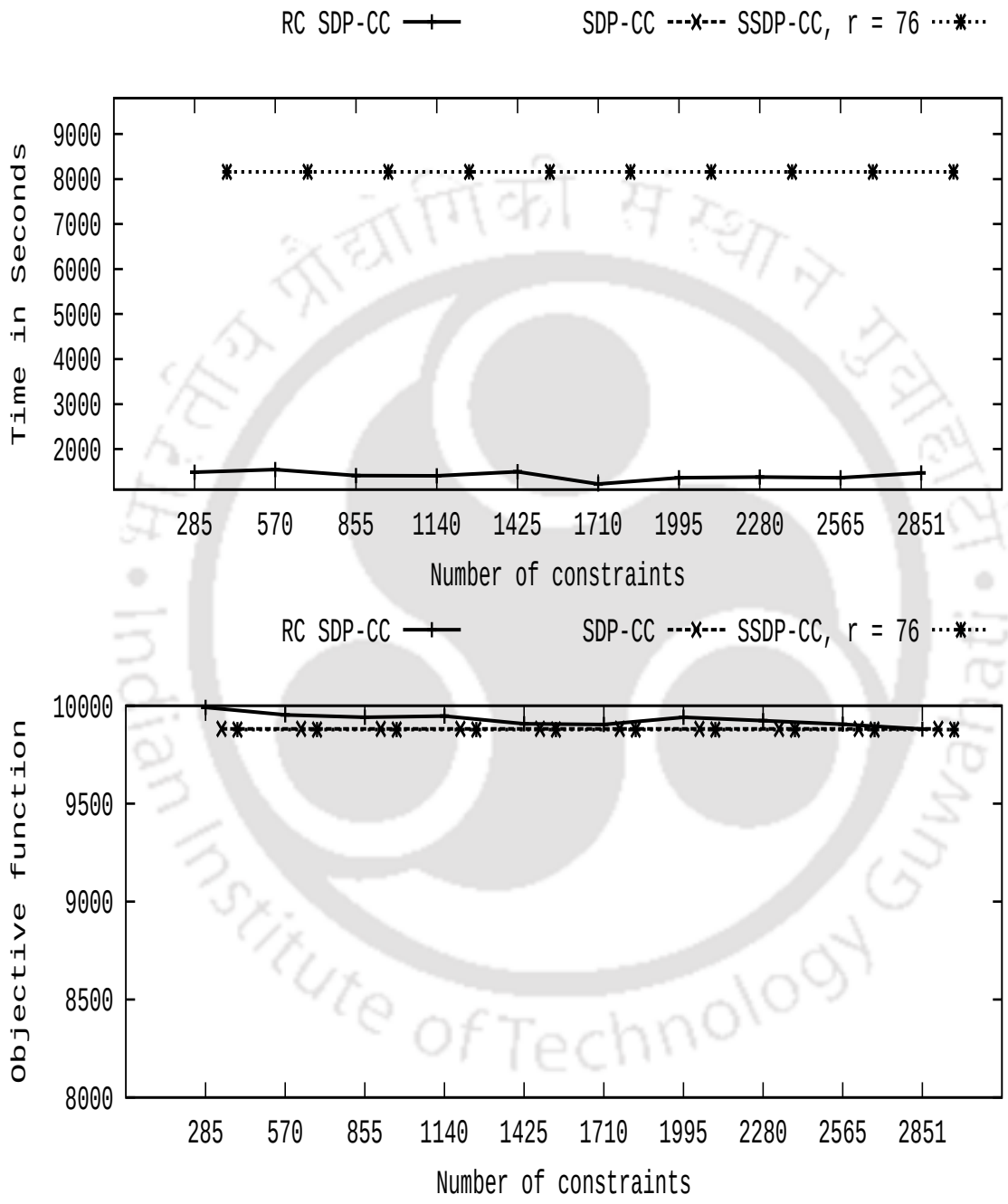

Figure 7.5: Graph Dataset: `Data`. $G = (V_g : 2851, E : 4062675)$.

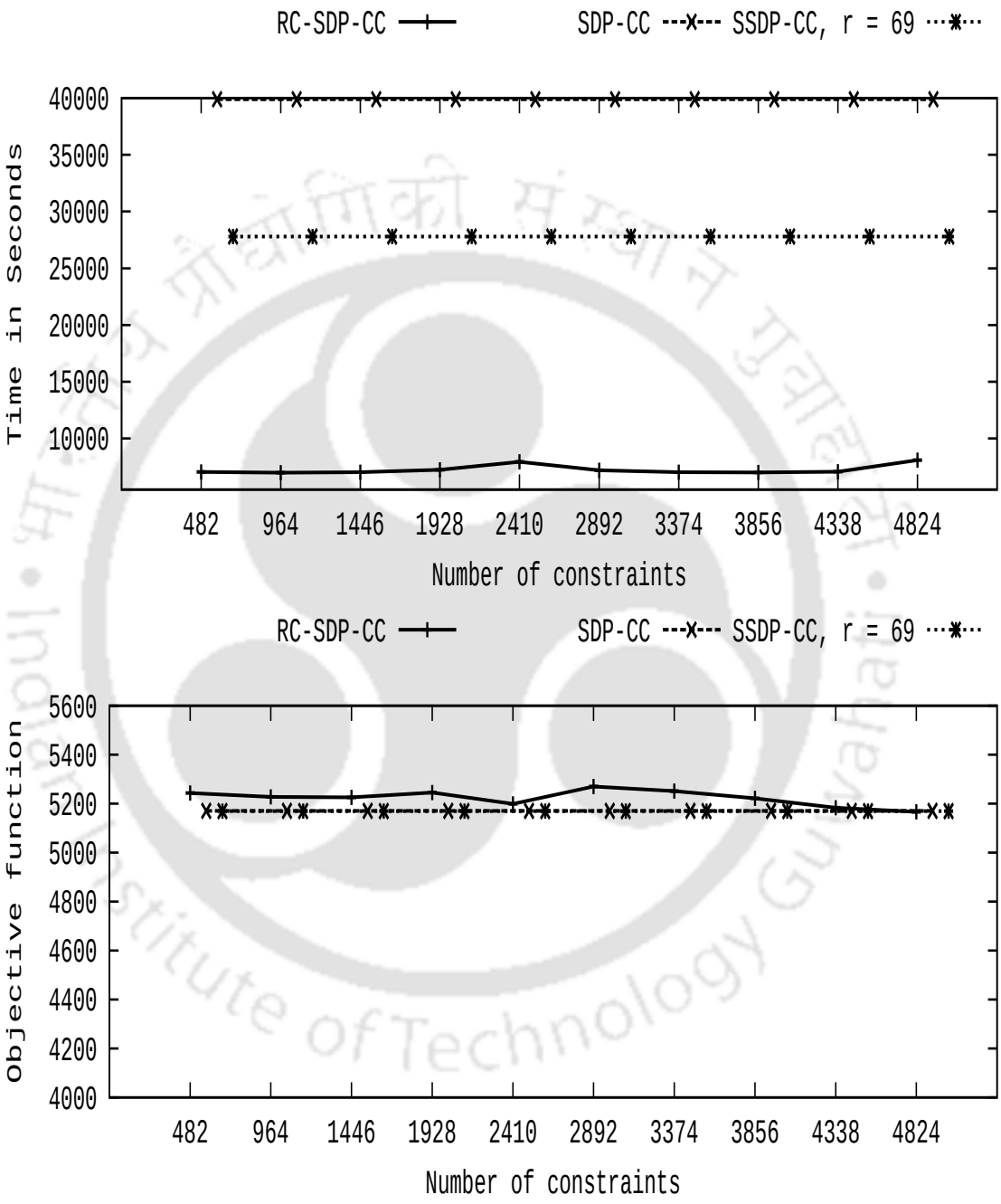

Figure 7.6: Graph Dataset: UK. $G = (V_g : 4824, E : 11633076)$.

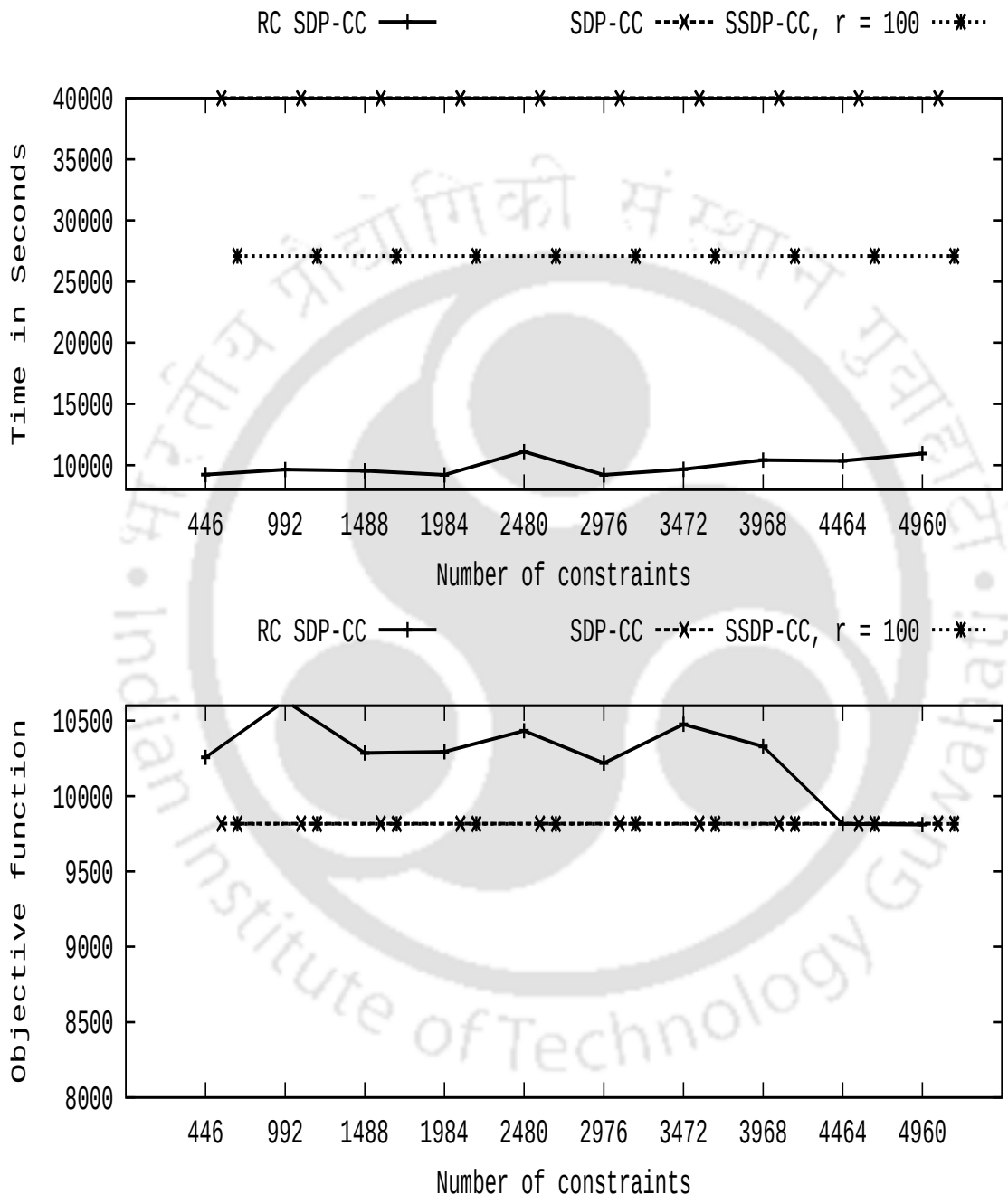

Figure 7.7: Graph Dataset: `add32`. $G = (V_g : 4960, E : 12298320)$.

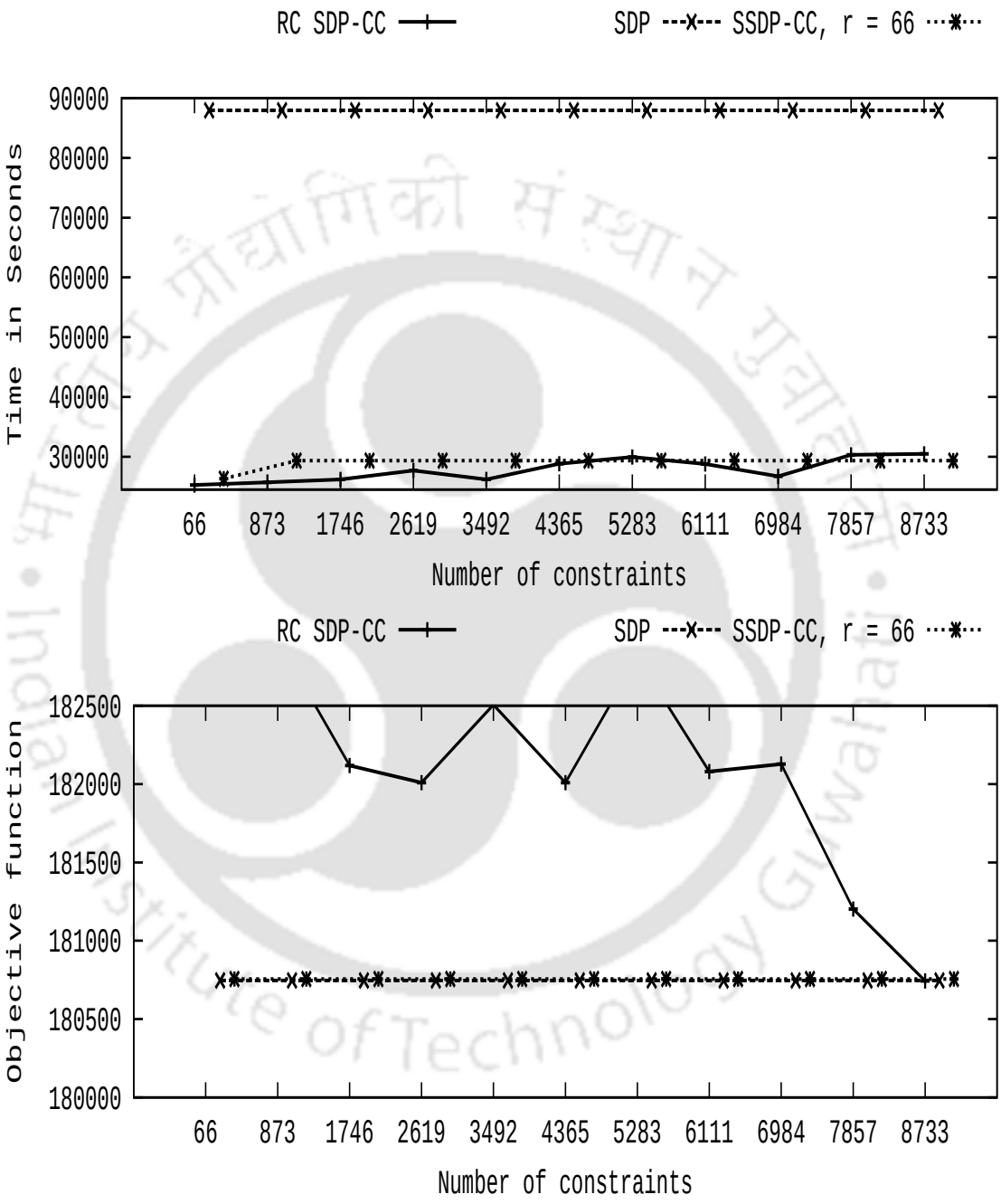

Figure 7.8: Graph Dataset: `bcsstk33`. $G = (V_g : 8733, E : 38128278)$.

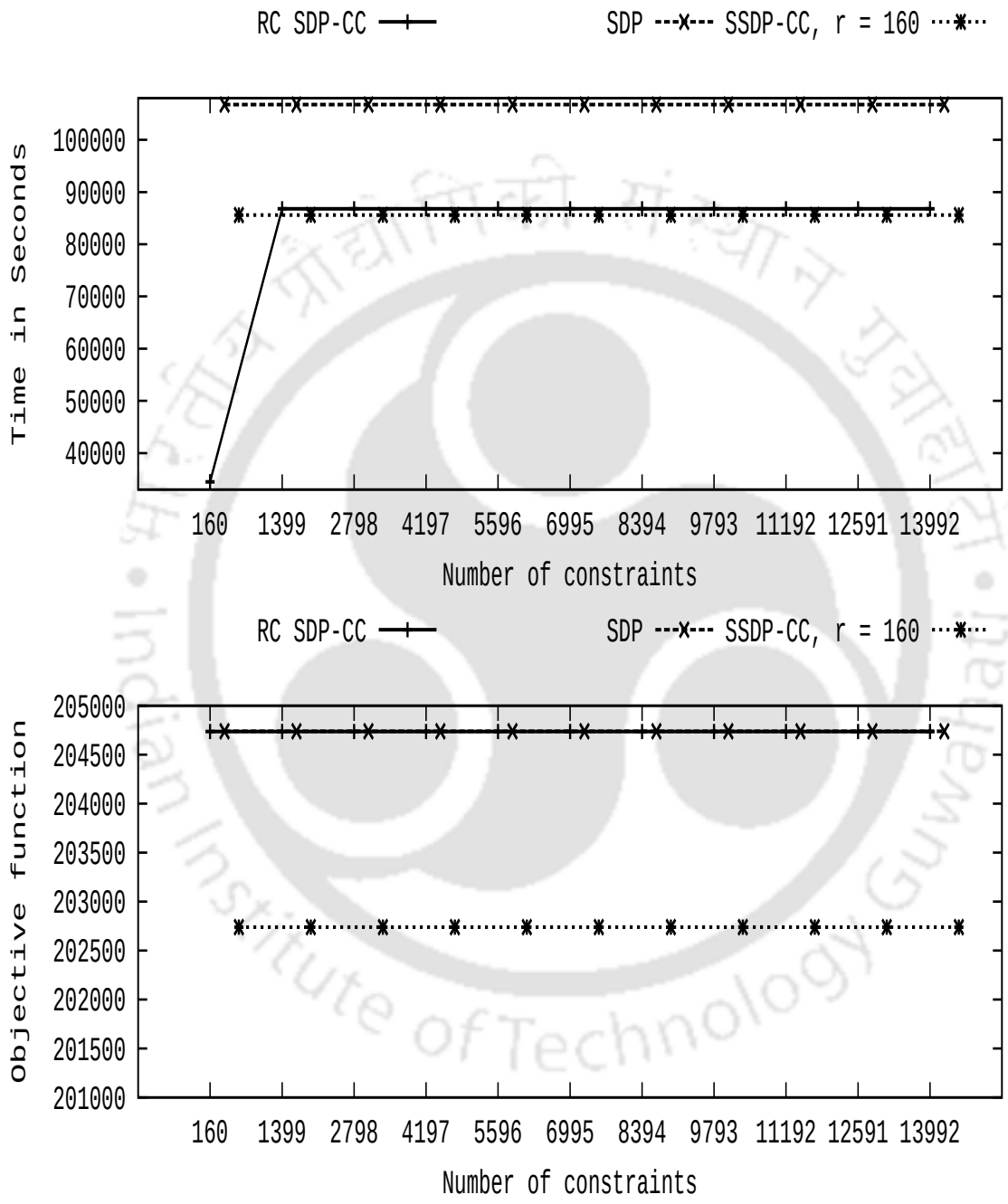

Figure 7.9: Graph Dataset: `bcsstk29`. $G = (V_g : 13992, E : 97880288)$.

noted on this graph dataset. Note that in case of variable reduction method, total number of variables considered is 70 which is significantly less than the number of constraints (239; 3 times that of the variable reduction) used in the proposed method; However, significant gains in the time to cluster and objective function value are observed for the proposed method.

In the case of objective function value, the proposed formulation outperforms the rest of the methods significantly. Note that in the case of large datasets the proposed method show its strength along all the comparing dimensions.

A similar observation is made from graph dataset: `data, UK, add32, bcsstk33` and `bcsstk29` whose results are presented in Figures 7.5, 7.6, 7.7, 7.8 and 7.9 respectively. For each of the datasets, the number of variables used are noted in the respective figures. In proposed method, the number of constraints employed are denoted on the x-axis of each of the figures. In each of these figures, time taken for solving the optimization formulation is less by a factor of 6 without degrading the objective function value (and hence internal quality).

For all these graph datasets where constraint reduction has shown superiority over time and objective function value, the number of constraints are not reduced to match variable reduction formulation. Only in case of `bcsstk33` and `bcsstk29` number of constraints are reduced to match the number of variables reduced. Figures 7.8 and 7.9 reflect these experiments. Note that along time and objective function value dimensions, the proposed formulation has an edge over SSDP-CC formulation.

7.2.1 RC SDP-CC Comparison with Constrained Spectral Clustering

As in the case of chapter 6, proposed constraint reduction method is compared with cSC method with $\gamma = 0$. The cSC method is compared with RC SDP-CC for the external quality measure. The objective function values are not compared as they differ in their objectives. As cSC involve obtaining eigenvectors of the modified graph Laplacian [78], time taken for this operations is: $O(kn^2)$ where k is the number of steps involved in eigenvalue computation which is significantly less compared to the SDP-CC or RC SDP-CC. The proposed formulation is therefore compared with that of the cSC along external quality dimension alone.

In the case of `Gaussian noise` dataset, cSC yields a rand index of 0.669 where as SDP-CC obtains a rand index of 0.889 and RC SDP-CC obtains a rand index of 0.905. In case of `Pendigit` dataset, cSC yields rand index of 0.694; SDP-CC obtains 0.787 and RC SDP-CC obtains 0.788. Superiority of the proposed RC SDP-CC formulation in the

graph datasets is demonstrated experimentally compared to constrained spectral clustering method.

7.3 Summary

In this chapter, the fast SDP relaxation is adopted in the CC for achieving scalability. The introduced weaker constraints help in reducing the number of constraints in the SDP-CC formulation. Experimental results on variety of datasets and graphs show the merit of the constraint reduction on (a) time taken to obtain clusters (b) objective function value of the obtained clusters and (c) external quality of the obtained clusters. The proposed scalable formulation is also compared with variable reduction scalable formulation. Trade off among the two scalable variants are experimentally analyzed. The proposed formulation is able to scale to graph data set having 13992 vertices and 97 million edges. The constraint reduction is a promising approach in achieving best objective function value by significantly reducing the computational time. A hybrid technique which reduces constraints and variables in the SDP formulation is a promising direction for retaining the external quality as well.

Summary of the contributions of this thesis is presented in the next chapter along with future research question in CC.

Chapter 8

Summary and Future work

8.1 Summary

This dissertation analyzed the quality of CC. The work on CC in the literature has focused on the theoretical developments. To the best of our knowledge very few attempts are made in applying CC in practice. This thesis contributed in four distinct research directions which are not considered in the literature. These are as detailed below.

- **Role of Rounding Techniques:** Several rounding procedures exist for rounding the solutions obtained by SDP formulations. In this thesis, impact of three rounding techniques namely `hyperplane rounding`, `outward rotation rounding` and RPR^2 `rounding`, are employed to understand the relative strengths and weaknesses of each of these rounding techniques on the quality of two cluster CC. Contradictions to the theoretical results are observed experimentally on diverse dataset characteristics.

- **Role of Optimal Graph Construction Methods:** In graph based clustering methods one needs to compute graph from the given vector dataset. Sensitivity of the input graph on the quality of the obtained clusters has been analyzed theoretically. This thesis provides the role of **optimal graph** on the quality of obtained clusters and convergence of the quality indexes. Two optimal graph construction methods are considered. One method yields sparse weighted general optimal graph and the other method yields optimal complete graph. Empirically the influence of the optimal graphs on CC's cluster quality is examined. Effectiveness of the optimal graphs is observed from the experimental results.

- **Comparison of Constrained Clustering Methods:** A comparative study is carried out to understand the quality of obtained clusters through CC with class of clustering algorithms that take into account domain knowledge in terms of pair wise constraint widely known as constrained clustering. To the best of our knowledge, the thesis made the first time comparison of CC with constraint clustering as CC being a variant of constraint clustering. Four constraint clustering methods are considered, namely constrained K-means, constraint spectral clustering, constraint spectral clustering with local proximity measure and flexible constrained spectral clustering. From the experiments it is observed that CC has an edge over constraint clustering methods when only the constraints are considered for clustering the input data.

- **Scalability:** As CC formulation involve solving an expensive SDP problem, CC on large scale datasets pose challenge. In this thesis, scaling of the CC formulation for clustering large datasets is addressed. In particular two variants of scalable solutions are presented:

  1. Scalable formulation in which number of variables are reduced from the equation (2.6). A non-linear optimization problem is formulated while reducing the number of variables. This non-linear optimization formulation is solved with the help of standard optimization methods, namely limited memory BFGS.
  2. Scalable formulation in which number of constrains are reduced from the equation (2.6). Number of linear equality constraints are reduced to a specified number and the SDP formulation is solved with the reduced number of constraints.

  Both variants are observed to have their merit in achieving scalability. Benchmark datasets having a maximum number of vertices approximately 14000 and maximum number of edges 97 million could be solved by the proposed solution without compromising on the quality of the obtained clusters.

8.2 Future Work

The following are some of the further research directions that are of particular importance pertaining to CC.

1. **MINDISAGREE/MAXCORR:** This thesis is devoted to understanding the CC through MAXAGREE formulation. Other objective function variants of CC are MINDISAGREE and MAXCORR. It will be interesting to understand the influence of these objective functions on the quality of CC.

2. **Multi-cluster CC:** This thesis is confined to obtaining two cluster solutions. However, multi-cluster solutions are preferred in practice. Extending the quality analysis from two cluster solutions to multi-cluster solutions is an interesting direction. In particular rounding techniques play a vital role in obtaining multiple clusters. Hence examining their strengths in the multi-cluster formulations is a natural extension.

3. **One-class CC:** In this class of problems, given a dataset belonging to one particular class say '+' examples, the objective is to learn the notion of negativeness from the positively related examples [74]. The one-class classification/clustering problems gained popularity due to the robustness in identifying the negativeness notion accurately. In the context of CC, Giotis *et al.* obtained a CC formulation involving fixed number of clusters. The minimum number of clusters to be obtained is at least 2. It is interesting to combine the ideas from one class classification and the fixed number of clusters in CC formulation to obtain one-cluster solution.

4. **Overlapping CC:** In the CC, each data point (or vertex) is mapped uniquely to a distinct cluster. In the overlapping CC, a data point potentially belong to multiple clusters simultaneously. In this setting, the objective function of CC is changed to incorporate the information that a data point belong to multiple clusters. To achieve this, a similarity function between sets of cluster labels is defined. That is if a vertex u belong to a set of cluster labels say C_u and vertex v belong to a set of cluster labels say C_v, then similarity, H(C_u, C_v) between C_u and C_v is defined. The objective in overlapping CC is to minimizing the difference between the H(C_u, C_v) and similarity between (u, v) over all possible pairs of (u, v) [11]. This cost function is shown to be NP-Hard. A heuristic is proposed to solve this optimization formulation. One needs to re-look at how to solve the overlapping CC optimization formulation efficiently.

5. **Incremental CC:** Incremental clustering methods consider that the data points arrive in an online fashion. In the incremental version the new data point is assigned to a cluster based on the edge labels of the new data point with all the existing data points. In this case it may create a new singleton cluster or add the new data point

to existing cluster or can combine existing clusters. Chekuri *et al.* proposed the incremental clustering in [55] for information retrieval application. In the context of CC, incremental methods are another handle for scalability.

References

[1] Gaurav Agarwal and David Kempe, *Modularity-maximizing graph communities via mathematical programming*, Eur. Phys. J. B., 66 (2008), pp. 409–418.

[2] Nir Ailon, Noa Avigdor-Elgrabli, Edo Liberty, and Anke van Zuylen, *Improved approximation algorithms for bipartite correlation clustering.*, SIAM J. Comput., 41 (2012), pp. 1110–1121.

[3] Miguel F. Anjos and Henry Wolkowicz, *Strengthened semidefinite relaxations via a second lifting for the max-cut problem*, Discrete Appl. Math., 119 (2002), pp. 79 – 106.

[4] A. Arasu, C. Ré, and D. Suciu, *Large-scale deduplication with constraints using dedupalog*, in International conference on Data Engineering, 2009, pp. 952 – 963.

[5] A. Asuncion and D.J. Newman, *UCI machine learning repository*, 2007.

[6] Nikhil Bansal, Avrim Blum, and Shuchi Chawla, *Correlation clustering*, in Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2002), IEEE Computer Society, 2002, pp. 238–247.

[7] Stephen T. Barnard and Horst Simon, *Fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems*, in In Proceedings of the 6th SIAM conference on Parallel Processing for Scientific Computing, 1993, pp. 711 – 718.

[8] S. Basu, I. Davidson, and K. Wagstaff, *Constrained Clustering: Advances in Algorithms, Theory and Applications*, Data Mining and Knowledge Discovery Series, Chapman & Hall/CRC, 1 ed., 2008.

[9] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini, *Clustering gene expression patterns*, J. Comput. Biol., 6 (1999), pp. 281–297.

[10] M. Bilenko, S. Basu, and R. J. Mooney, *Integrating constraints and metric learning in semi-supervised clustering*, in 21st International conference on machine Learning (ICML), 2004, pp. 81–88.

[11] Francesco Bonchi, Aristides Gionis, and Antti Ukkonen, *Overlapping correlation clustering*, Knowledge and Information Systems, 35 (2013), pp. 1 – 32.

[12] P. S. Bradley and O. L. Mangasarian, *Massive data discrimination via linear support vector machines*, Optim. Methods Softw., 13 (2000), pp. 1–10.

[13] Samuel Burer and Renato D. C. Monteiro, *A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization*, Math. Program., 95 (2003), pp. 329–357.

[14] N. Cesa-Bianchi, C. Gentile, F. Vitale, and G. Zappella, *A correlation clustering approach to link classification in signed networks*, Journal of Machine Learning Research, 23 (2012), pp. 1 – 34.

[15] Swamy Chaitanya, *Correlation clustering: maximizing agreements via semidefinite programming*, in Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms(SODA), 2004, pp. 526–527.

[16] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth, *Clustering with qualitative information*, J. Comput. System Sci., 71 (2005), pp. 360–383.

[17] Y. Chen, S. Sanghavi, and H. Xu, *Clustering sparse graphs*, in NIPS, 2012.

[18] Flavio Chierichetti, Nilesh Dalvi, and Ravi Kumar, *Correlation clustering in mapreduce*, in KDD '14: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014.

[19] Eden Chlamtac and Madhur Tulsiani, *Handbook on Semidefinite, Conic and Polynomial Optimization*, Springer, 2012, ch. Convex Relaxations and Integrality Gaps, pp. 139 – 169.

[20] Tom Coleman, James Saunderson, and Anthony Wirth, *A local-search 2-approximation for 2-correlation-clustering*, in Proceedings of the 16th annual European symposium on Algorithms, Springer-Verlag, 2008, pp. 308–319.

[21] Samuel I. Daitch, Jonathan A. Kelner, and Daniel A. Spielman, *Fitting a graph to vector data*, in 26th International conference on machine Learning(ICML), 2009, pp. 201–208.

[22] I. Davidson and S. S. Ravi, *Clustering with constraints: Feasibility issues and the K-means algorithm*, in 5th SIAM Data Mining Conferences, SIAM, 2005, pp. 138 – 149.

[23] Tijl De Bie and Nello Cristianini, *Fast SDP relaxations of graph cut clustering, transduction, and other combinatorial problems*, J. Mach. Learn. Res., 7 (2006), pp. 1409–1436.

[24] Dotan Emanuel and Amos Fiat, *Correlation clustering—minimizing disagreements on arbitrary weighted graphs*, in Algorithms–ESA, vol. 2832 of Lecture Notes in Comput. Sci., 2003, pp. 208–220.

[25] Uriel Feige and Michel X. Goemans, *Aproximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT*, in ISTCS, 1995, pp. 182–189.

[26] Uriel Feige, Marek Karpinski, and Michael Langberg, *Improved approximation of Max-Cut on graphs of bounded degree*, J. Algorithms, 43 (2002), pp. 201–219.

[27] Uriel Feige and Michael Langberg, *The* RPR^2 *rounding technique for semidefinite programs*, J. Algorithms, 60 (2006), pp. 1–23.

[28] Maurizio Filippone, Francesco Camastra, Francesco Masulli, and Stefano Rovetta, *A survey of kernel and spectral methods for clustering*, Pattern Recognition, 41 (2008), pp. 176 – 190.

[29] Douglas H. Fisher, *Knowledge acquisition via incremental conceptual clustering*, Mach. Learn., (1987), pp. 139–172.

[30] Alan M. Frieze and Mark Jerrum, *Improved approximation algorithms for MAX k CUT and MAX BISECTION.*, in Integer Programming and Combinatorial Optimization, vol. 920 of Lecture Notes in Comput. Sci., Springer, 1995, pp. 1–13.

[31] Ioannis Giotis and Venkatesan Guruswami, *Correlation clustering with a fixed number of clusters*, Theory of Computing. An Open Access Journal, 2 (2006), pp. 249–266.

[32] Michel X. Goemans and David P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. Assoc. Comput. Mach., 42 (1995), pp. 1115–1145.

[33] Eran Halperin and Uri Zwick, *A unified framework for obtaining improved approximation algorithms for maximum graph bisection problems*, in Integer programming and combinatorial optimization, vol. 2081 of Lecture Notes in Comput. Sci., 2001, pp. 210–225.

[34] Anil K. Jain, *Data clustering: 50 years beyond K-means*, Pattern Recogn. Lett., 31 (2010), pp. 651–666.

[35] Anil K. Jain, M. N. Murty, and P. J. Flynn, *Data clustering: A review*, ACM Comput. Surv., 31 (1999), pp. 264–323.

[36] Hongjie Jia, Shifei Ding, Xinzheng Xu, and Ru Nie, *The latest research progress on spectral clustering*, Neural Computation and Applications, (2013).

[37] T. Joachims and J. Hopcroft, *Error bounds for correlation clustering*, in Proceedings of the 22nd International Conference on Machine Learning(ICML), 2005, pp. 385–392.

[38] David S. Johanson., *Approximation algorithms for combinatorial problems*, Journal of comput. and Sys. Sci, (1974), pp. 256–278.

[39] Sepandar D. Kamvar, Dan Klein, and Christopher D. Manning, *Spectral learning*, in Proceedings of the 18th international joint conference on Artificial intelligence, 2003, pp. 561–566.

[40] Ashish Kapoor, Yuan (Alan) Qi, Hyungil Ahn, and Rosalind Picard, *Hyperparameter and kernel learning for graph based semi-supervised classification*, in Advances in Neural Information Processing Systems 18, 2005, pp. 627–634.

[41] David Karger, Rajeev Motwani, and Madhu Sudan, *Approximate graph coloring by semidefinite programming*, J. ACM, 45 (1998), pp. 246–265.

[42] Howard Karloff and Uri Zwick, *A 7/8-approximation algorithm for MAX 3SAT?*, in Proceedings of the 38th Annual Symposium on Foundations of Computer Science(SFCS), 1997, pp. 406–415.

[43] George Karypis and Vipin Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM J. Sci. Comput., 20 (1999), pp. 359–392.

[44] Sungwoong Kim, Sebastian Nowozin, Pushmeet Kohli, and Chang D. Yoo, *Higher-order correlation clustering for image segmentation*, in Advances in Neural Information Processing Systems 24, Curran Associates, Inc., 2011, pp. 1530–1538.

[45] Dan Klein, Sepandar D. Kamvar, and Christopher D. Manning, *From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering*, in Proceedings of the 19th International Conference on Machine Learning(ICML), 2002, pp. 307–314.

[46] Michael Langberg, *Coping with NP-hardness: Approximation algorithms based on semidefinite programming*, PhD thesis, The Weizmann Institute of Science, Israle, 2003.

[47] Michael Lewin, Dror Livnat, and Uri Zwick, *Improved rounding techniques for the MAX 2-SAT and MAX DICUT problems*, in Integer programming and combinatorial optimization, vol. 2337 of Lecture Notes in Comput. Sci., 2002, pp. 67–82.

[48] Karl J. Lieberherr and Ernst Specker, *Complexity of partial satisfaction.*, J. ACM, 28 (1981), pp. 411–421.

[49] Dong C. Liu and Jorge Nocedal, *On the limited memory BFGS method for large scale optimization*, Math. Programming, 45 (1989), pp. 503–528.

[50] Xianmin Liu and Jianzhong Li, *Algorithms and complexity results for labeled correlation clustering problem*, J. Comb. Optim., 29 (2015), pp. 488–501.

[51] Ulrike Luxburg, *A tutorial on spectral clustering*, Statistics and Computing, 17 (2007), pp. 395–416.

[52] J. MacQueen, *Some methods for classification and analysis of multivariate observations*, in Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[53] Markus Maier, Ulrike von Luxburg, and Matthias Hein, *Influence of graph construction on graph-based clustering measures.*, in Neural Information Processing Systems (NIPS), 2008, pp. 1025–1032.

[54] Markus Maier, Ulrike von Luxburg, and Matthias Hein, *How the result of graph clustering methods depends on the construction of the graph*, ESAIM: Probability and Statistics, 17 (2013), pp. 370 – 418.

[55] Claire Mathieu, Ocan Sankur, and Warren Schudy, *Online correlation clustering*, in Symposium on Theoretical Aspects of Computer Science(STACS), 2010, pp. 573–584.

[56] Mariá C.V. Nascimento and André C.P.L.F. de Carvalho, *Spectral methods for graph clustering a survey*, European J. Oper. Res., 211 (2011), pp. 221 – 231.

[57] J. Saketha Nath, Chiranjib Bhattacharyya, and M. Narasimha Murty, *Clustering based large margin classification: a scalable approach using socp formulation*, in ACM International Conference on Knowledge Discovery and Data Mining(SIGKDD), 2006, pp. 674 – 679.

[58] M. E. J. Newman, *Detecting community structure in networks*, Eur. Phys. J. B, 38 (2004), pp. 321–330.

[59] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss, *On spectral clustering: Analysis and an algorithm*, in Advances In Neural Information Processing Systems, 2001, pp. 849–856.

[60] Rabia Nuray-Turan, Dmitri V. Kalashnikov, and Sharad Mehrotra, *Exploiting web querying for web people search*, ACM Trans. Database Syst., 37 (2012), pp. 7:1–7:41.

[61] Christos H. Papadimitriou and Mihalis Yannakakis, *Optimization, approximation, and complexity classes*, J. Comput. System Sci., 43 (1991), pp. 425–440.

[62] D. Pelleg and D. Baras, *K-means with large and noise constraint sets*, in Eighteenth European Conference on Machine Learning, 2007, pp. 674 – 682.

[63] ——, *K-means with large and noise constraint sets*, tech. report, IBM Research Divison, 2007.

[64] William M. Rand, *Objective criteria for the evaluation of clustering methods*, Journal of the American Statistical Association, 66 (1971), pp. 846–850.

[65] Nicola Rebagliati and Alessandro Verri, *Spectral clustering with more than k eigenvectors*, Neurocomputing, 74 (2011), pp. 1392–1401.

[66] Mohammad Hossein Rohban and Hamid R. Rabiee, *Supervised neighborhood graph construction for semi-supervised classification*, Pattern Recogn., 45 (2012), pp. 1363 – 1372.

[67] HéCtor Ruiz, Terence A. Etchells, Ian H. Jarman, José D. MartíN, and Paulo J. G. Lisboa, *A principled approach to network-based classification and data representation*, Neurocomputing, 112 (2013), pp. 79–91.

[68] Sartaj Sahni and Teofilo Gonzalez, *P-complete approximation problems*, J. ACM, 23 (1976), pp. 555–565.

[69] Satu Elisa Schaeffer, *Survey: Graph clustering*, Comput. Sci. Rev., 1 (2007), pp. 27–64.

[70] D. F. Shanno and K. H. Phua, *Minimization of unconstrained multivariate functions*, ACM Trans. Math. Software, 6 (1980), pp. 618 – 622.

[71] J. B. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Transactions On Pattern Analysis And Machine Intelligence, 22 (2000), pp. 888–905.

[72] H. Shin, N. J. Hill, and G. Rätsch, *Graph based semi-supervised learning with sharper edges*, in European Conference on Machine Learning, 2006, pp. 402 – 412.

[73] R. Tapia, Y.Zhang, M. Saltzman, and A. Weiser, *The mehrotra predictor-corrector interior-point methoh as a perturbed composite newton method*, Tech. Report TR90-17, Rice University, July 1990.

[74] D. M. J. Tax and R. P. W. Duin, *Support vector domain description*, Pattern Recogn. Lett., 20 (1999), pp. 1191 – 1199.

[75] M. J. Todd, *Semidefinite optimization*, Acta Numer., 10 (2001), pp. 515–560.

[76] Jurgen Van Gael and Xiaojin Zhu, *Correlation clustering for crosslingual link detection*, in Proceedings of the 20th International Joint Conference on Artifical Intelligence, IJCAI'07, Morgan Kaufmann Publishers Inc., 2007, pp. 1744–1749.

[77] Lieven Vandenberghe and Stephen Boyd, *Semidefinite programming*, SIAM Review., 38 (1996), pp. 49–95.

[78] G. Wacquet, í. Poisson Caillault, D. Hamad, and P. A. HéBert, *Constrained spectral embedding for k-way data clustering*, Pattern Recogn. Lett., 34 (2013), pp. 1009–1017.

[79] Kiri Wagstaff and Claire Cardie, *Clustering with instance-level constraints*, in Proceedings of the 17th International Conference on Machine Learning (ICML), 2000, pp. 1103–1110.

[80] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl, *Constrained K-means clustering with background knowledge*, in Proceedings of the 18th International Conference on Machine Learning (ICML), 2001, pp. 577–584.

[81] Xiang Wang and Ian Davidson, *Flexible constrained spectral clustering*, in KDD '10: Proceedings of the 16th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2010, pp. 563–572.

[82] Xiang Wang, Buyue Qian, and Ian Davidson, *On constrained spectral clustering and its applications*, Data Mining and Knowledge Discovery, 28 (2014), pp. 1–30.

[83] David P. Williamson and David B. Shmoys, *The Design of Approximation Algorithms*, Cambridge University Press, first ed., 2011.

[84] Anthony Ian Wirth, *Approximation Algorithms for Clustering*, PhD thesis, Princeton University, 2005.

[85] Dachuan Xu, *On approximation of MAX $\frac{n}{2}$ UNCUT problem*, Journal of Systems Science and Complexity, 16 (2003), pp. 260–267.

[86] Qianjun Xu and Marie Desjardins, *Constrained spectral clustering under a local proximity structure assumption*, in Proceedings of the 18th International Conference of the Florida Artificial Intelligence Research Society, 2005, pp. 866–867.

[87] Rui Xu and Donald Wunsch, *Survey of clustering algorithms*, IEEE Trans. Neural Netw., 16 (2005), pp. 645 – 678.

[88] Stella X. Yu and Jianbo Shi, *Grouping with bias*, in Advances in Neural Information Processing Systems (NIPS), 2001, pp. 1327–1334.

[89] Zhenya Zhang, Hongmei Cheng, Shuguang Zhang, Wanli Chen, and Qiansheng Fang, *Clustering aggregation based on genetic algorithm for documents clustering*, in IEEE Congress on Evolutionary Computation, 2008, pp. 3156–3161.

[90] Xin-Yuan Zhao, Defeng Sun, and Kim-Chuan Toh, *A newton-cg augmented lagrangian method for semidefinite programming*, SIAM J. on Optimization, 20 (2010), pp. 1737–1765.

[91] X. Zhu, J. Kandola, Z. Ghahramani, and J. Lafferty, *Nonparametric transforms of graph kernels for semi-supervised learning*, in Advances in Neural Information Processing Systems, 2006, pp. 1641–1648.

[92] Uri Zwick, *Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems*, in Annual ACM Symposium on Theory of Computing, 1999, pp. 679–687.

Publications Related to Thesis

Published:

1. Mamata Samal, V. Vijaya Saradhi and S. Nandi, Correlation clustering: Quality Analysis and Scalability. 6^{th} Workshop for Women in Machine Learning (WiML) 2011, Co-located with **NIPS 2011**, Granada, Spain.

2. Mamata Samal, V. Vijaya Saradhi and S. Nandi, Scalability of Correlation Clustering Through Constraint Reduction. **Conference on Data Sciences 2014** (CoDS 2014).

Communicated:

Journals

1. Mamata Samal, V. Vijaya Saradhi and S. Nandi, Scalability of Correlation clustering. Revised and resubmitted to **Neural Computing and Applications** journal.

Brief Biography of the Author

Mamata Samal has completed her M.Sc. in Mathematics and M.Tech. in Computer Science from Utkal University, Bhubaneswar, Odisha, India in 2002 and 2004, respectively. Currently she is pursuing her Ph.D. and she is currently working as a senior research fellow at the Indian Institute of Technology Guwahati, Her research interests include Machine learning, Pattern Recognition, Data Mining and Optimization techniques.

Department of Computer Science and Engineering

Indian Institute of Technology Guwahati

Guwahati 781039, India