

University of Louisville

## ThinkIR: The University of Louisville's Institutional Repository

---

Electronic Theses and Dissertations

---

12-2013

### An inter-domain supervision framework for collaborative clustering of data with mixed types.

Artur Abdullin  
*University of Louisville*

Follow this and additional works at: <https://ir.library.louisville.edu/etd>



Part of the [Computer Engineering Commons](#)

---

#### Recommended Citation

Abdullin, Artur, "An inter-domain supervision framework for collaborative clustering of data with mixed types." (2013). *Electronic Theses and Dissertations*. Paper 6.  
<https://doi.org/10.18297/etd/6>

This Doctoral Dissertation is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact [thinkir@louisville.edu](mailto:thinkir@louisville.edu).

AN INTER-DOMAIN SUPERVISION FRAMEWORK FOR  
COLLABORATIVE CLUSTERING OF DATA WITH MIXED  
TYPES

By

Artur Abdullin  
M.S., Perm State University, 2007  
M.S., University of Louisville, 2009

A Dissertation  
Submitted to the Faculty of the  
J. B. Speed School of Engineering of the University of Louisville  
in Partial Fulfillment on the Requirements  
for the Degree of

Doctor of Philosophy

Department of Computer Engineering and Computer Science  
University of Louisville  
Louisville, Kentucky

December 2013

Copyright 2013 by Artur Abdullin

All rights reserved



AN INTER-DOMAIN SUPERVISION FRAMEWORK FOR COLLABORATIVE CLUSTERING  
OF DATA WITH MIXED TYPES

By

Artur Abdullin  
Master of Science in Computer Science  
Master of Science in Physics

A Dissertation Approved

on December 2nd, 2013

by the following Dissertation Committee:

---

Dr. Olfa Nasraoui - Dissertation Director

---

Dr. Jacek M. Zurada

---

Dr. Hichem Frigui

---

Dr. Adel Elmaghraby

---

Dr. Amir Amini

## DEDICATION

I dedicate my dissertation work to my family and many friends. To my parents, Rinat and Faniya Abdullin, who made all of this possible, for their endless encouragement, support, and patience. A special feeling of gratitude to my lovely wife, Kristina Aksenova, and my sister, Albina Abdullina, who has never left my side and is very special.

## ACKNOWLEDGMENTS

I wish to thank my committee members who were more than generous with their expertise and precious time. A special thanks to Dr. Olfa Nasraoui, my committee chair for her excellent guidance, caring, countless hours of reflecting, reading, encouraging, and most of all, patience throughout the entire process. Thank you, Dr. Nasraoui. Thank you Dr. Jacek M. Zurada, Dr. Hichem Frigui, Dr. Adel Elmaghraby, and Dr. Amir Amini for agreeing to serve on my committee and your valuable feedback.

I would like to thank Dr. Aleksey Fadeev, who as a good friend, was always willing to help and give his best suggestions. Many thanks to Dr. Aleksey Ashikhmin, Dr. Roman Yampolskiy, Nicole and Andrei Radzionau for their friendship, support and encouragement. My research would not have been possible without their help.

This work was supported by US National Science Foundation Data Intensive Computation Grant IIS-0916489.

## ABSTRACT

# AN INTER-DOMAIN SUPERVISION FRAMEWORK FOR COLLABORATIVE CLUSTERING OF DATA WITH MIXED TYPES

Artur Abdullin

December 2nd, 2013

We propose an Inter-Domain Supervision (IDS) clustering framework to discover clusters within diverse data formats, mixed-type attributes and different sources of data. This approach can be used for combined clustering of diverse representations of the data, in particular where data comes from different sources, some of which may be unreliable or uncertain, or for exploiting optional external concept set labels to guide the clustering of the main data set in its original domain. We additionally take into account possible incompatibilities in the data via an automated inter-domain compatibility analysis. Our results in clustering real data sets with mixed numerical, categorical, visual and text attributes show that the proposed IDS clustering framework gives improved clustering results compared to conventional methods, over a wide range of parameters. Thus the automatically extracted knowledge, in the form of seeds or constraints, obtained from clustering one domain, can provide additional knowledge to guide the clustering in another domain. Additional empirical evaluations further show that our approach, especially when using selective mutual guidance between domains, outperforms common baselines such as clustering either domain on its own or clustering all domains converted to a single target domain. Our approach also outperforms other specialized multiple clustering methods, such as the fully independent ensemble clustering and the tightly coupled multiview clustering, after they were adapted to the task of clustering mixed data. Finally, we present a real

life application of our IDS approach to the cluster-based automated image annotation problem and present evaluation results on a benchmark data set, consisting of images described with their visual content along with noisy text descriptions, generated by users on the social media sharing website, Flickr.

# TABLE OF CONTENTS

<b>ACKNOWLEDGMENTS</b>	<b>iv</b>
<b>ABSTRACT</b>	<b>v</b>
<b>LIST OF TABLES</b>	<b>ix</b>
<b>LIST OF FIGURES</b>	<b>xii</b>
<b>1 INTRODUCTION AND MOTIVATION</b>	<b>1</b>
1.1 Motivations . . . . .	1
1.2 Objectives . . . . .	2
1.3 Summary of contributions . . . . .	2
1.4 Organization of this Dissertation . . . . .	4
<b>2 BACKGROUND AND RELATED WORK</b>	<b>5</b>
2.1 Introduction and Chapter Organization . . . . .	5
2.2 Clustering . . . . .	5
2.2.1 Different attribute types . . . . .	6
2.3 Clustering Data with Mixed Attribute Types . . . . .	8
2.3.1 Conversion and Splitting . . . . .	8
2.3.2 Clustering a Combined Dissimilarity Matrix . . . . .	11
2.3.3 Algorithms for Mixed-Type Data Clustering . . . . .	13
2.3.4 Multiview Clustering (MC) . . . . .	14
2.3.5 Ensemble Clustering (EC) . . . . .	15
2.3.6 Collaborative Clustering (CC) . . . . .	17
2.4 Semi-Supervised Clustering . . . . .	18
2.5 Limitations of Existing Methods and Comparison with the Proposed Work . . . . .	19
2.6 Clustering Evaluation . . . . .	21

2.6.1	Internal index metrics . . . . .	22
2.6.2	External index metrics . . . . .	24
2.7	Chapter Summary and Discussion . . . . .	25
<b>3</b>	<b>METHODOLOGY</b>	<b>31</b>
3.1	Seed-based Inter-Domain Supervision (Seed-based IDS) . . . . .	33
3.1.1	The Case of an Equal Number of Clusters in Each Data Type or Domain . .	33
3.1.2	Different Seed Exchange Mechanisms . . . . .	34
3.1.3	Computational Complexity . . . . .	36
3.1.4	The Case of a Different Number of Clusters or Different Cluster Partitions in each Data Type or Domain . . . . .	37
3.1.5	Computational Complexity . . . . .	39
3.2	Constraint-based Inter-Domain Supervision . . . . .	40
3.2.1	Mutual Inter-Domain Supervision using Hidden Markov Random Fields (HMRF): HMRF-KMeans . . . . .	40
3.2.2	Algorithm Flow . . . . .	43
3.2.3	Computational Complexity . . . . .	44
3.3	How to use Other Existing Clustering Paradigms for the Purpose of Clustering Het- erogeneous Data . . . . .	45
3.3.1	Ensemble Clustering . . . . .	45
3.3.2	Multiview Clustering . . . . .	46
3.4	Discovering Domain Compatibility in Heterogeneous Data . . . . .	48
3.5	Summary of the Chapter . . . . .	49
<b>4</b>	<b>EXPERIMENTAL RESULTS AND APPLICATION TO IMAGE ANNOTA- TION</b>	<b>50</b>
4.1	Real-Life Data Sets . . . . .	51
4.2	Results for the Inter-Domain Supervised Clustering . . . . .	58
4.2.1	Results for the Seed-based IDS Clustering: Effect of the Seed Exchange Mech- anism and Convergence . . . . .	58
4.2.2	Results for the Constraint-based IDS Clustering: Studying the Impact of the Number of Constraints . . . . .	66
4.2.3	Comparison of the Proposed IDS Framework with Other Clustering Methods	70
4.3	Results of the Compatibility Analysis Experiments . . . . .	80

4.4	Application: Image Auto-Annotation . . . . .	87
4.5	Summary of the Chapter . . . . .	94
<b>5</b>	<b>CONCLUSIONS AND FUTURE WORK</b>	<b>96</b>
5.1	Summary . . . . .	96
5.2	Current Status and Future Prospects . . . . .	97
	<b>REFERENCES</b>	<b>97</b>
	<b>CURRICULUM VITAE</b>	<b>107</b>

## LIST OF TABLES

2.1	Different attribute types. . . . .	7
2.2	Overview of clustering approaches. . . . .	26
3.1	List of notations . . . . .	32
4.1	Overview of the clustering evaluation measures. . . . .	50
4.2	Experimental plan overview for Section 4.2. . . . .	52
4.3	Experimental plan overview for Section 4.3. . . . .	53
4.4	Experimental plan overview for Section 4.4: first set of the experiments. . . . .	54
4.5	Experimental plan overview for Section 4.4: second set of the experiments. . . . .	55
4.6	Real-life data set properties. . . . .	56
4.7	Adult data set attribute description. . . . .	56
4.8	Heart disease data set attribute description. . . . .	56
4.9	Credit card approval data set attribute description. . . . .	57
4.10	Clustering results of seed-based IDS for the Adult data set with different seed exchange mechanisms (10 runs, $k = 2$ clusters per domain). . . . .	64
4.11	Clustering results of seed-based IDS for the Heart disease data set with different seed exchange mechanisms (50 runs, $k = 2$ clusters per domain). . . . .	64
4.12	Clustering results of seed-based IDS for the Credit card approval data set with different seed exchange mechanisms (50 runs, $k = 2$ clusters per domain). . . . .	65
4.13	Clustering results of seed-based IDS for the MIRFlickr data set with different seed exchange mechanisms (10 runs, $k = 16$ clusters per domain). . . . .	66
4.14	Clustering results for the Adult data set (10 runs, $k = 2$ clusters per domain). We run the Seed-based IDS with the following parameters: $k = 2$ number of seeds, and the constraint-based IDS: $n_{T_1} = n_{T_2} = 5$ , $t_{T_1} = t_{T_2} = 1$ . . . . .	75

4.15	Clustering results for the Heart disease data set (50 runs, $k = 2$ clusters per domain). We run the seed-based IDS with the following parameters: $k = 2$ clusters and number of seeds, and the constraint-based IDS: $n_{T_1} = 5$ , $n_{T_2} = 11$ , and $t_{T_1} = t_{T_2} = 1$ . . . . .	76
4.16	Clustering results for the Credit card approval data set (50 runs, $k = 2$ clusters per domain). We run the seed-based IDS with the following parameters: $k = 2$ clusters and number of seeds, and the constraint-based IDS: $n_{T_1} = 5$ , $n_{T_2} = 11$ , and $t_{T_1} = t_{T_2} = 1$ . . . . .	78
4.17	Clustering results for the MIRFlickr data set (10 runs, $k = 16$ clusters per domain). We run the seed-based IDS with the following parameters: $k = 2$ clusters and number of seeds, and the constraint-based IDS: $n_{T_1} = n_{T_2} = 5$ , $t_{T_1} = t_{T_2} = 1$ . . . . .	79
4.18	Clustering results of the seed-based IDS (with DB-exchange) for the mixed, incom- patible, and compatible sets (10 runs, $k = 16$ clusters per domain). . . . .	83
4.19	Clustering results of the constraint-based IDS for the mixed, incompatible, and com- patible sets (10 runs, $k = 16$ clusters per domain, $n_{T_1} = 11$ , $n_{T_2} = 5$ , and $t_{T_1} = t_{T_2} = 1$ ). . . . .	83
4.20	Clustering results of the conversion algorithm for the mixed, incompatible, and com- patible sets. (10 runs, $k = 16$ clusters per domain). . . . .	88
4.21	Clustering results of the splitting algorithm for the mixed, incompatible, and com- patible sets (10 runs, $k = 16$ clusters per domain). . . . .	89
4.22	Clustering results of the ensemble voting algorithm for the mixed, incompatible, and compatible sets (10 runs, $k = 16$ clusters per domain, 2 instances for the text domain, and 3 instances for the text domain). . . . .	89
4.23	Clustering results of multiview clustering for the mixed, incompatible, and compatible sets (10 runs, $k = 16$ clusters per domain). . . . .	90
4.24	Value of the $MAP_{f_{max}=3}$ and $MAP_{f_{max}=5}$ for the image auto-annotation of the MIR- Flickr data set for the different validation schemes. . . . .	92
4.25	Value of $MAP_{f_{max}=3}$ and $MAP_{f_{max}=5}$ for the MIRFlickr data set with the compatible and mixed training sets for the the different validation schemes. . . . .	93

## LIST OF FIGURES

1.1	An example that illustrates a typical scenario for the IDS clustering framework. . . .	3
2.1	Overview of the clustering approaches, typically used for heterogeneous data sets, and the proposed IDS clustering framework. . . . .	8
2.2	Seed-based and constraint-based semi-supervised clustering approaches . . . . .	19
2.3	Multiview clustering. . . . .	20
2.4	Ensemble clustering. . . . .	21
2.5	Collaborative clustering. . . . .	22
3.1	Overview of the Seed-based Inter-Domains Supervised clustering algorithm. . . . .	35
3.2	Seed-based IDS, stage 3: best seeds combination selection. . . . .	36
3.3	Different number of clusters per domain. . . . .	38
3.4	An illustration of the split-domain clustering stage. . . . .	38
3.5	An illustration of the Hidden Markov Random Fields framework for the constrained cluster label assignments. . . . .	40
3.6	An illustration of the HMRF-Kmeans [Basu et al., 2004] objective function. Blue arrows represent the distortion between data records and cluster centroids, green arrows represent the must-link constraints, and red arrows represent the cannot-link constraints. . . . .	42
3.7	Outline of the mutual inter-domain supervision based heterogeneous data clustering using HMRF-KMeans. . . . .	45
3.8	Ensemble clustering framework for clustering data with mixed attributes types. . . .	46
3.9	Multiview clustering framework for clustering data with mixed attributes types. . . .	47
3.10	Domain Compatibility Analysis. . . . .	48
4.1	Representation of a data record from the MIRFlickr data set. . . . .	58

4.2	Value of the objective functions (with number of exchange seeds) for seed-based IDS with different exchange mechanisms for the Adult data set (dashed line: baseline splitting algorithm with no exchange). . . . .	60
4.3	Value of the objective functions (with number of exchange seeds) for seed-based IDS with different exchange mechanisms for the Heart disease data set (dashed line: baseline splitting algorithm with no exchange). . . . .	61
4.4	Value of the objective functions (with number of exchange seeds) for seed-based IDS with different exchange mechanisms for the Credit card approval data set (dashed line: baseline splitting algorithm with no exchange). . . . .	62
4.5	Value of the objective functions (with number of exchange seeds) for seed-based IDS with different exchange mechanisms for the MIRFlickr data set (dashed line: baseline splitting algorithm with no exchange). . . . .	63
4.6	Constraint-based IDS ( $k_{T_1} = k_{T_2} = 2$ , $t_{T_1} = t_{T_2} = 1$ ): effect of the number of constraints in the Adult data set. Warm colors indicate improvement and cold colors indicate decline over the baseline splitting algorithm. . . . .	68
4.7	Constraint-based IDS ( $k_{T_1} = k_{T_2} = 2$ , $t_{T_1} = t_{T_2} = 1$ ): effect of the number of constraints in the Heart disease data set. Warm colors indicate improvement and cold colors indicate decline over the baseline splitting algorithm. . . . .	69
4.8	Constraint-based IDS ( $k_{T_1} = k_{T_2} = 2$ , $t_{T_1} = t_{T_2} = 1$ ): effect of the number of constraints in the Credit card approval data set. Warm colors indicate improvement and cold colors indicate decline over the baseline splitting algorithm. . . . .	70
4.9	Value of the objective functions for seed-based IDS (red diamonds), constraint-based IDS (green stars), splitting clustering (dashed black squares), and multiview clustering (dotted black circles). See the value of the validity indices in Table 4.14 for the Adult data set and Table 4.15 for the Heart disease data set. The number of clusters is set to $k = 2$ for both data sets. . . . .	72
4.10	Value of the objective functions for seed-based IDS (red diamonds), constraint based IDS (green stars), splitting clustering (dashed black squares), and multiview clustering (dotted black circles). See the value of the validity indices in Table 4.16 for the Credit card approval data set and Table 4.17 for the MIRFlickr data set. The number of cluster is $k = 2$ for the Credit card approval data set, and $k = 16$ for the MIRFlickr data set. . . . .	73
4.11	Sample data from the compatible clusters. . . . .	80

4.12	Sample data from the incompatible clusters. . . . .	81
4.13	Value of the objective function (and number of exchange seeds) within seed-based IDS for the compatible (solid blue line) and incompatible (dashed blue line) sets. Red diamonds represent a seed exchange between domains. The maximum number of seed exchanged between domains is 16, same as the number of clusters. . . . .	82
4.14	Constraint-based IDS ( $k_{T_1} = k_{T_2} = 16$ , $t_{T_1} = t_{T_2} = 1$ ): effect of the number of constraints in the compatible set of the MIRFlickr data set. Warm colors indicate improvement and cold colors indicate decline over the baseline splitting algorithm. .	85
4.15	Constraint-based IDS ( $k_{T_1} = k_{T_2} = 16$ , $t_{T_1} = t_{T_2} = 1$ ): effect of the number of constraints in the compatible set of the MIRFlickr data set. Warm colors indicate improvement and cold colors indicate decline over the mixed set. . . . .	86
4.16	Constraint-based IDS ( $k_{T_1} = k_{T_2} = 16$ , $t_{T_1} = t_{T_2} = 1$ ): effect of the number of constraints in the compatible set of the MIRFlickr data set. Warm colors indicate improvement and cold colors indicate decline over the incompatible set. . . . .	87
4.17	Value of the objective function for constraint-based IDS for the compatible (solid green line) and incompatible (dashed dark green line) sets ( $k = 16$ clusters per domain, $n_{T_1} = 5$ , $n_{T_2} = 5$ , and $t_{T_1} = t_{T_2} = 1$ ). . . . .	88
4.18	An example illustrating the two cluster-based annotation schemes, with final number of tags, $f_{max} = 2$ . . . . .	90
4.19	Value of $MAP_{f_{max}=3}$ for the image auto-annotation of the MIRFlickr data set for the different validation options: seed-based IDS with normal seed exchange (solid red diamonds), constraint-based IDS (solid green stars), conversion clustering (dotted black circles), multiview clustering (solid magenta circles), ensemble clustering (solid blue circles), and splitting clustering (dashed black squares). See the value of $MAP_{f_{max}=3}$ and $MAP_{f_{max}=5}$ in Table 4.24. . . . .	91
4.20	Value of the $MAP_{f_{max}=3}$ for the image auto-annotation with the compatibility analysis of the MIRFlickr data set for the different validation options: seed-based IDS with DB index-based seed exchange (solid red diamonds), constraint-based IDS (solid green stars), conversion clustering (dotted black circles), multiview clustering (solid magenta circles), ensemble clustering (solid blue circles), and splitting clustering (dashed black squares). See the value of the $MAP_{f_{max}=3}$ and $MAP_{f_{max}=5}$ in Table 4.20. . . . .	94

# CHAPTER 1

## INTRODUCTION AND MOTIVATION

### 1.1 Motivations

Advances in sensing, storage technology and dramatic growth in applications such as Internet search, e-commerce, social media sites, and digital imaging have created large, high-dimensional data sets. Most of this data is stored digitally in electronic media, thus providing a huge potential for the development of automatic data analysis, classification, and retrieval techniques. In addition to the growth in the amount of data, the *variety* of available data (text, image, and video) has also increased especially on social media sites such as Flickr and Youtube. The availability of large data collections with no or limited information concerning the membership of data items to a predefined class, has turned increasing attention toward the need for *unsupervised* and *semi-supervised learning*. In unsupervised learning or *clustering*, there are no explicit labels; instead cluster analysis groups data based only on information found in the data that describes the objects and their relationships. The goal is to assign objects such that objects within the same group are similar to one another and different from the objects in other groups [Tan et al., 2005]. In semi-supervised learning, only a small portion of the data is labeled, and the goal is to exploit both labeled and unlabeled data for better learning [Basu et al., 2002b].

Recent years have seen an increasing interest in clustering data comprising multiple domains or modalities, such as categorical, numerical, text, transactional, and visual modalities. This kind of data is sometimes found within the context of clustering *multiview*, *heterogeneous*, or *multimodal* data. Traditionally each of these different types of data has been best clustered with a different specialized clustering algorithm or with a specialized dissimilarity measure [Dhillon and Modha, 2001, Banerjee et al., 2005, Huang, 1998a]. A very common approach to cluster data with mixed types has been to either convert all data types to the same type (e.g: from categorical to numerical or vice-versa) and then cluster the data with a standard clustering algorithm that is suitable for that target domain; or to use a different dissimilarity measure for each domain, then combine them into

one dissimilarity measure and cluster this dissimilarity matrix. However, there are many different contexts in which the plurality of data exist. For example, in multiview data, the features of the data can naturally be divided into subsets (views), such that each of which is sufficient to learn a target concept. In multimodal data, there exist more than one modality in the data. One example is online images (on Flickr or Facebook) with visual content features and tags. Another example is data consisting of ratings, clickstreams, and transactions by users relating to items purchased or viewed online. A third example would be user feedback in the form of user ratings and textual reviews/comments. Generally, due to the abundance of user-generated diverse data formats, there is an increasing need for clustering algorithms that can exploit all or part of the diverse data descriptions in a way that best combines the knowledge that can be extracted from each source.

## 1.2 Objectives

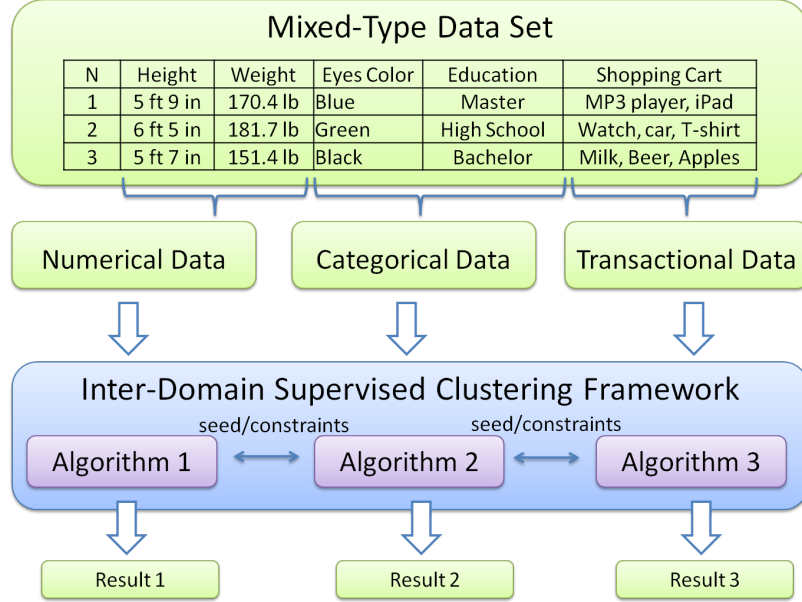
The objective of this work is to develop an unsupervised learning approach for combined clustering of diverse representations of the data, in particular where data representatives come from different sources or domains, consisting of possibly different types, and where the different sources of data may disagree or be incompatible in how they delineate the groups or clusters.

## 1.3 Summary of contributions

We propose a new methodology for clustering data comprising multiple domains or parts, in such a way that the separate domains mutually supervise each other within a framework that is similar to semi-supervised learning. However, unlike semi-supervised learning, our methodology does not assume the presence of any external labels from any part of the data; rather, each of the different domains of the data separately undergoes an unsupervised learning process, while receiving some guidance or supervision in the form of data constraints or seeds that are discovered from clustering the other domains. As illustrated with an example in Figure 1.1, the entire process can be considered to be very similar to the alternation of semi-supervised learning stages in the different data domains, with each domain receiving selective guidance or supervision that is automatically discovered from clustering the other domain. The same approach can also be used for multi-source data regardless of the type of data in each source, since each source of data can be considered as a separate domain.

Our contributions can be summarized as follows:

- We propose a *seed-based* Inter-Domain Supervision approach to transfer knowledge from the



**Figure 1.1:** An example that illustrates a typical scenario for the IDS clustering framework.

clustering in one domain to the other domains (Section 3.1).

- We propose different seed exchange mechanisms for the seed-based IDS, in order to control the *selectivity* of the exchanged knowledge, based on linear-complexity unsupervised internal cluster validity indices (Section 3.1.2).
- We propose a *constraint-based* Inter-Domain Supervision approach to handle inconsistent partitions between different domains, which can now be combined into a consistent clustering result (Section 3.2).
- We propose a *domain compatibility analysis* approach for a more effective clustering of heterogeneous data, by exploiting the synergy between the different domains, even when some inter-domain incompatibility exists in the descriptions of parts of the data (Section 3.4).
- We outline a general methodology to utilize a variety of other clustering approaches (ensemble, multiview and collaborative clustering) to the problem of mixed data type clustering, although some of them were generally designed for different purposes.
- We perform an exhaustive evaluation of the proposed methods for a variety of real data sets with varying sizes, dimensionality, and number of clusters, and study the effect of the parameters governing the clustering process on the quality of the results. The data is composed of a variety of types: numerical, categorical, visual image features, and text descriptions.

- We propose an cluster-based automated annotation methodology that exploits both the image visual content, and the associated text of a set of training images from Flickr, and furthermore demonstrate the benefit of the proposed inter-domain compatibility analysis.
- Our empirical evaluations on clustering a variety of mixed type data sets show that our proposed IDS framework can achieve a significant improvement over two baseline methods and two sophisticated methods, based on most validity metrics, and can provide significant improvement in Mean Average Precision (MAP) on the automated annotation task for the MIRFlickr data, consisting of visual and text domains.

## 1.4 Organization of this Dissertation

The rest of this dissertation is organized as follows. Chapter 2 gives an overview of the pertinent background and related work. Chapter 3 presents a new Inter-Domain Supervision (IDS) clustering framework to cluster heterogeneous data with compatibility analysis. Chapter 4 presents experimental results that evaluate our proposed approach in comparison with **(1)** two commonly used approaches, treated as baselines: (1.a) independent clustering of split-domains, thus with no inter-domain exchange of guidance, and (1.b) clustering a combined data obtained by a conversion of the multiple domains into a single domain; **(2)** two alternative competitive approaches adapted to the problem of clustering multiple data domains: (2.a) multiview clustering, and (2.b) ensemble clustering. Finally, Chapter 5 presents our conclusions.

## CHAPTER 2

### BACKGROUND AND RELATED WORK

#### 2.1 Introduction and Chapter Organization

We start this chapter with a short introduction to the clustering problem and review different types of data attributes. We then present the most common approaches to clustering heterogeneous data, including conversion, splitting and combined dissimilarity measure approaches. We also discuss several approaches that can be used for the purpose of clustering mixed data, although most of them have been proposed for different purposes. In this respect, we review the related areas of multiview clustering, ensemble clustering, and collaborative clustering, and explain how each one can be modified for the specific purpose of clustering heterogeneous data. We follow our review of clustering algorithms with a review of the cluster validity metrics that are typically used to evaluate the results of clustering algorithms. Finally, we conclude with a comparison of the discussed methods.

#### 2.2 Clustering

Data clustering is also known as cluster analysis, Q-analysis, typology, clumping, and taxonomy depending on the field where it is applied [Jain and Dubes, 1988]. The goal of clustering is to discover the *natural* groupings of a set of patterns, points, or objects. The problem of clustering, in general, is to partition a set  $O = \{o_1, o_2, \dots, o_n\}$  of objects embedded in a  $d$ -dimensional space into  $k$  distinct sets of clusters  $C = \{C_1, C_2, \dots, C_k\}$  based on a measure of *similarity* such that the similarities between objects in the same cluster are high, while the similarities between objects in different clusters are low. Clusters can differ in terms of their shape, size, and density. An ideal cluster can be defined as a set of objects which is compact and separated from other clusters.

Traditionally, data clustering has been used for the following main purposes.

- Discovering an underlying structure: to gain insight into data, generate hypotheses, detect anomalies, and identify salient features [Lakhina et al., 2005, Gal and Cohen-Or, 2006, Boley et al., 1999].

- Natural classification: to identify the degree of similarity among forms or organisms (phylogenetic relationship) [Remm et al., 2001].
- Compression: as a method for organizing the data and summarizing it through cluster prototypes [Equitz, 1989].
- Recent applications of clustering: information retrieval, customer segmentation, recommendation systems, visualization, etc [Frakes and Baeza-Yates, 1992, Espinoza et al., 2005, Ungar et al., 1998].

Existing clustering algorithms can be broadly classified into partitional, hierarchical, and density-based [Jain and Dubes, 1988]. A hierarchical clustering is a sequence of partitions in which each partition is nested into the next partition in the sequence. The result is a hierarchical structure of groups known as *dendrogram*. Hierarchical clustering algorithms [Johnson, 1967, Fisher, 1987, Steinbach et al., 2000] recursively find nested clusters in either an agglomerative mode or a divisive mode. Agglomerative hierarchical clustering starts with every single object in a single cluster. Then it repeats merging the closest pair of clusters according to some similarity criteria until all of the data are in a single cluster. In contrast to agglomerative mode, the divisive mode starts with all data points in the same cluster and repeats splitting each cluster into smaller clusters. Input to a hierarchical clustering algorithm is an  $n \times n$  similarity matrix, where  $n$  is the number of objects to be clustered. Partitioning clustering methods [MacQueen, 1967, Bezdek et al., 1984, Krishnapuram and Keller, 1993] try to obtain a single partition of data without any other subpartition like hierarchical algorithms do, and are often based on the optimization of an appropriate objective function [Gan et al., 2007]. As an input, a partitional clustering algorithm can use either an  $n \times d$  pattern matrix, where  $n$  objects are embedded in  $d$ -dimensional feature space, or an  $n \times n$  similarity matrix.

Hard (or crisp) clustering algorithms assign each object to a single cluster. On the other hand, fuzzy (or soft) clustering algorithm assign every object to every cluster with a membership weight that is between 0 (absolutely does not belong to the cluster) and 1 (absolutely belongs to the cluster) [Bezdek, 1981]. Density-based clustering methods such as DBSCAN [Ester et al., 1996] seek clusters by relying on the notion of dense regions of space that are separated by relatively vacuous areas. Some of the algorithms are reviewed in Section 2.3.1.

### 2.2.1 Different attribute types

Each data object in a data set is described by a set of attributes. An attribute is a property of an object that may vary, either from one object to another or from one time to another [Tan et al., 2005].

Attribute Type		Description	Examples
Categorical	Nominal	The values of a nominal attribute are just different names or codes.	zip codes, eye color, gender
	Ordinal	The values of an ordinal attribute provide enough information to order objects.	grades, street numbers, quality (poor, good, better)
Numerical	Interval	For interval attributes, the difference between values are meaningful, e.i, a unit of measurement exists.	calendar dates, temperature in Fahrenheit.
	Ratio	For ratio variables, both differences and ratios are meaningful	temperature in Kelvin, counts, age, mass, length
Transactional		In transactional data each data record or transaction consists of a set of items	web user sessions, clickstreams, items in a shopping cart, documents

**Table 2.1:** Different attribute types.

For example, eye color varies from person to person, while the age of the same person varies over time. The eye color is a *categorical* attribute with a small number of possible values (blue, brown, green, etc), while age is a *numerical* attribute with a limited number of values. A useful way to specify the type of the attribute is to identify the properties of the values that correspond to the underlying properties of the attribute. For example, an attribute such as age has many of the properties of numbers. The following properties of numbers are typically used to describe attributes:

1. Distinctness = and  $\neq$
2. Order  $<$ ,  $\leq$ ,  $>$ , and  $\geq$
3. Addition  $+$  and  $-$
4. Multiplication  $*$  and  $/$

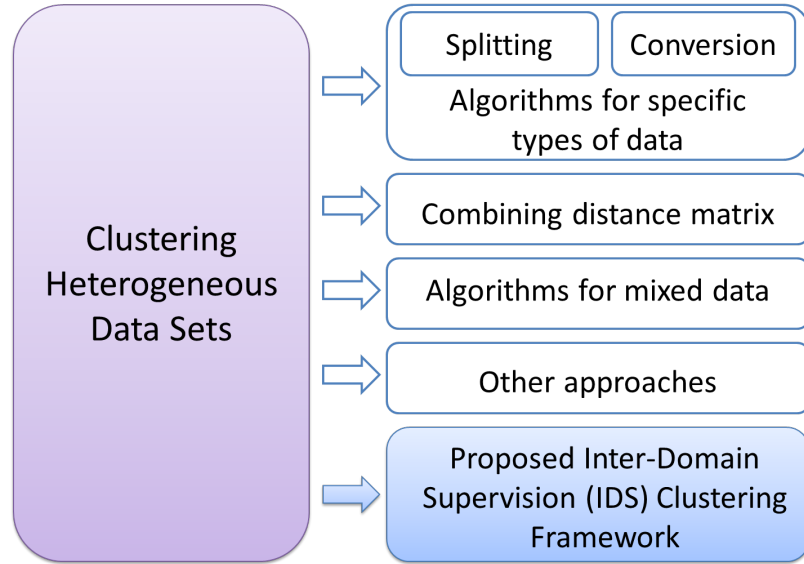
Given these properties, we can define four types of attributes: nominal, ordinal, interval, and ratio (see Table 2.1).

Nominal and ordinal attributes are collectively referred to as *categorical* attributes and interval and ratio attributes are called *numerical* attributes. Although some categorical attributes like zip codes or IDs are represented by numbers, they do not share properties of these numbers and should

be treated more like symbols. Numerical attributes are represented by numbers and have most of the properties of numbers, they could be integer valued or continuous. In addition, to the numerical and categorical attributes there is a third type - *transactional*. Transactional data is a special type of data, where each data record or transaction consists of a set of items. Examples of this data include (1) user activity records such as web user sessions or clickstreams, where the items are the set of actions or pages that can be clicked and (2) text documents, where the items are the words or tokens. Transactions could be represented by binary vectors with each dimension corresponding to one item, in which an entry denotes the presence or absence of the corresponding item. Usually transactional data sets have a high number of dimensions and in many cases, such as online transactions, in particular web user sessions, are extremely sparse.

## 2.3 Clustering Data with Mixed Attribute Types

In order to cluster data consisting of mixed types, there are several approaches which will be described in the following subsections in the order of sophistication level, ranging from simple data conversion, distance measure combination and then dedicated mixed data type clustering algorithm, and finally ensemble, multiview and collaborative clustering (see Figure 2.1).



**Figure 2.1:** Overview of the clustering approaches, typically used for heterogeneous data sets, and the proposed IDS clustering framework.

### 2.3.1 Conversion and Splitting

There are different ways to handle data consisting of multiple domain with different types of attributes, for the purpose of clustering. The first and most popular approach to clustering data with

mixed attributes is convert all data to the same target domain and cluster it with a specially design algorithm. We call this method the *conversion* approach. This approach is very common because it is a convenient and a fast solution that does not discard any of the attributes, and because most successful clustering algorithms are specialized for one specific target type of attributes.

The conversion algorithm requires data type conversion, there are different ways to convert one data type to another. For example, to convert numerical type attribute  $z$ , ranging in  $[z_{min}, z_{max}]$ , to a categorical type attribute  $y$ , also known as “discretization” [Gan et al., 2007], three strategies can be used:

1. mapping the  $n$  numerical values,  $z_i$ , to  $N$  categorical values  $y_i$  using direct categorization. The categorical value is defined as  $y_i = \lfloor \frac{N(z_i - z_{min})}{(z_{max} - z_{min})} \rfloor + 1$ , where  $\lfloor \cdot \rfloor$  denotes the largest integer less than or equal to  $z$ . Obviously, if  $z_i = z_{max}$ , we get  $y_i = N + 1$ , and we should set  $y_i = N$ .
2. mapping the  $n$  numerical values to  $N$  categorical values using a histogram binning based method.
3. clustering the  $n$  numerical values into  $N$  clusters using any numerical clustering algorithm (e.g. k-means). The optimal number of clusters  $N$  can be chosen based on some validation criteria.

There are also several methods to convert a categorical type attribute to the numerical domain:

1. by mapping the  $n$  values of a nominal attribute to binary values using 1-of- $n$  encoding, resulting in transactional-like data, with each nominal value becoming a distinct binary attribute
2. by mapping the  $n$  values of an ordinal nominal attribute to integer values in the range of 1 to  $n$ , resulting in numerical data with  $n$  values

Transaction data can be thought of as a special type of categorical or numerical data having boolean values, with all the possible items as attributes.

The second classical approach is to run a specially designed clustering algorithm independently on each domain, respectively, and then take the best clustering result into account. We call this method the *splitting* approach.

There are many specialized clustering algorithms for different types of data. For instance, categorical attributes have been handled using k-modes [Huang, 1998a], ROCK [Guha et al., 2000] or CAC-TUS [Ganti et al., 1999]. The main idea of the k-modes algorithm is to select  $k$  initial modes, followed by allocating every object to the nearest mode. The k-modes algorithm uses the matching dissimilarity measure to measure the distance between categorical objects [Kaufman and Rousseeuw, 1990].

ROCK is an adaptation of an agglomerative hierarchical clustering algorithm, which heuristically optimizes a criterion function defined in terms of the number of "links" between tuples, where the number of links between two tuples is the number of common neighbors. Starting with each tuple in its own cluster, the algorithm repeatedly merges the two closest clusters until the required number of clusters remain. The central idea behind CACTUS is that a summary of the entire data set is sufficient to compute a set of "candidate" clusters which can then be validated to determine the actual set of clusters. The CACTUS algorithm consists of three phases: computing the summary information from the data set, using this summary information to discover a set of candidate clusters, and then determining the actual set of clusters from the set of candidate clusters.

The spherical k-means algorithm is a variant of the k-means algorithm that uses the cosine similarity instead of the Euclidean distance. The algorithm computes a disjoint partitioning of the document vectors, and for each partition, computes a centroid normalized to have unit Euclidean norm [Dhillon and Modha, 2001]. This algorithm was successfully used for clustering transactional and text data (text documents are often represented as high-dimensional and sparse vectors). LargeItem [Wang et al., 1999] is an optimization algorithm designed for clustering transaction data based on the notion of large items without using any measure of pairwise similarity. The LargeItem algorithm consists of two phases: the allocation phase and refinement phase. Given a user-specified minimum support  $\theta$  ( $0 < \theta < 1$ ), an item  $i$  is large in a cluster  $C$  if its support, or number of transactions containing the item, is at least  $\theta|C|$ . Otherwise, item  $i$  is small in  $C$ . The criterion of a good clustering is that there are many large items within a cluster and there is little overlapping of such items across clusters. The objective function or cost function is defined in terms of the intracluster cost and intercluster cost. CLOPE is an algorithm designed for clustering transactional or categorical data [Yang et al., 2002]. Like most partitional clustering approaches, CLOPE has a criterion function that guides the algorithm to approximate the best partition by iteratively scanning the data set. This *global* criterion function tries to increase the intracluster overlapping of transaction items by increasing the height-to-width ratio of the cluster histogram. Different numbers of clusters can be obtained by varying a user-specified parameter  $r$ , which controls the tightness of the cluster.

Numerical data has been clustered using k-means [MacQueen, 1967], DBSCAN [Ester et al., 1996] and others [Nasraoui and Krishnapuram, 2002, Nasraoui and Krishnapuram, 1996]. The k-means algorithm [MacQueen, 1967] is a partitional or non-hierarchical clustering method, designed to cluster numerical data in which each cluster has a center called mean or centroid. The k-means algorithm proceeds as follow: for a given set of  $k$  initial clusters, the data are assigned to the nearest cluster center and the cluster centers are recomputed. The two previous steps are repeated until the

objective function (sum of distances from the data to their corresponding cluster centers) does not significantly change or the memberships of the clusters no longer change. DBSCAN is a density-based clustering algorithm designed to discover arbitrary shaped density-based clusters. A point  $\mathbf{x}$  is directly density reachable from a point  $\mathbf{y}$  if it is not farther away from it than a given distance  $\epsilon$  (i.e., is a part of its  $\epsilon$ -neighborhood), and if the  $\epsilon$ -neighborhood of  $\mathbf{y}$  has more points than a user-specified threshold parameter  $N_{min}$ , such that one may consider  $\mathbf{y}$  and  $\mathbf{x}$  to be part of a cluster.

The limitations of all the above approaches are as follows:

- Specialized clustering algorithms can fall short when they must handle different data types for which they are not specialized.
- Data type conversion can result in the loss of information (e.g: when a numerical range is discretized into a small number of levels), waste of storage (e.g: categorical attributes are typically transform into a large number of dimensions), or creation of artefacts in the data (e.g: an unfortunate discretization of a numerical attribute can map a majority of data to a single value).
- Different data sources can be hard to combine for the purpose of clustering because of the problem of duplication of data and the problem of missing data from one of the sources, in addition to the problem of heterogeneous types of data from multiple sources that are incompatible with one another. This means that combining data may be harmful to the knowledge discovery!

### 2.3.2 Clustering a Combined Dissimilarity Matrix

Besides data conversion, another common approach to clustering data with mixed attribute types is to pre-compute a specially designed distance measure for each subset of same-type attributes, then combine them into one dissimilarity measure and finally cluster the resulting dissimilarity matrix using a relational or kernel clustering algorithm [Frigui et al., 2007]. Relational clustering is more general in the sense that it is applicable to situations in which the objects to be clustered cannot be represented by numerical features [Nasraoui et al., 1999, Nasraoui and Frigui, 2000]. There are several well-known relational clustering algorithms in the literature. One of the most popular is the sequential agglomerative hierarchical nonoverlapping (SAHN) model, which is a bottom-up approach that generates crisp clusters by sequentially merging pairs of clusters that are closest to each other in each step [Sheath and Sokal, 1973]. Depending on how “closeness” between clusters is defined, the SAHN model gives rise to single, complete, or average linkage algorithms. A variation of this

algorithm can be found in [Guha et al., 1998]. Another well-known relational clustering algorithm is partitioning around medoids (PAM) [Kaufman and Rousseeuw, 1990]. This algorithm is based on finding representative objects from the data set in such a way that the sum of the within cluster dissimilarities is minimized. A modified version of PAM, called CLARA (clustering large applications) to handle large data sets relies on a sampling approach to handle large data sets.

Regarding kernel clustering methods, several clustering methods have been modified to incorporate kernels, this includes modifications of the: k-means [Muller et al., 2001, Girolami, 2002], fuzzy c-means [Zhang and Chen, ], SOM [Inokuchi and Miyamoto, 2004, MacDonald and Fyfe, 2000], and Neural gas [Qin and Suganthan, 2004]. Kernel-based learning algorithms are based on Cover’s theorem [Cover, 1965]. By nonlinearly transforming a set of complex and nonlinearly separable patterns into a higher-dimensional feature space, we can obtain the possibility to separate these patterns linearly. Kernel clustering methods can be broadly divided in three categories [Filippone et al., 2008], which are based, on:

- Kernelization of the metric. Methods based on kernelization of the metric look for centroids in the input space and the distances between patterns and centroids is computed by means of kernels;
- Clustering in the feature space. Clustering in the feature space is made by mapping each pattern using a nonlinear transformation  $\Phi$  and then computing the centroids in the feature space. Calling  $v_i^\Phi$  the centroids in the feature space, it is possible to compute the distance between a data sample and its cluster centroid in feature space by means of the kernel trick;
- Description via support vectors. The description via support vectors makes use of One Class SVM to find a minimum enclosing hypersphere in feature space able to enclose almost all data in feature space excluding outliers [Ben-Hur et al., 2002, Ben-Hur et al., 2001]. Data points are mapped from the input space to a high dimensional feature space using a kernel. In the feature space, we look for the smallest hypersphere that encloses the data. This hypersphere is mapped back to the input space, where it forms a set of contours which enclose the data points. These contours are interpreted as nonlinear arbitrary shaped cluster boundaries. Finally, the support vector clustering algorithm assigns the same label to the data that are enclosed by the same surface in the input space.

### 2.3.3 Algorithms for Mixed-Type Data Clustering

One approach to cluster data with mixed attribute types without any data conversions or pre-computation of a combined distance measure, is to use specialized clustering algorithms, which were designed to handle mixed-type data. Several algorithms for mixed data attributes exist, for instance the k-prototypes [Huang, 1998b], INCONCO [Plant and Böhm, 2011], k-means-mixed [Ahmad and Dey, 2007], and CAVE [Hsu and Chen, 2007]. The k-prototype algorithm integrates the k-means [MacQueen, 1967] and the k-modes [Huang, 1998a] algorithms to allow for clustering objects described by mixed numerical and categorical attributes. The k-prototypes works by simply combining the Euclidean distance and categorical distance measures in a weighted sum. The choice of the weight parameter and the weighting contribution of the categorical versus numerical domains cannot vary from one cluster to another, and this can be considered as a limitation. The INCONCO algorithm extends the Cholesky decomposition [Kershaw, 1978] to model dependencies in heterogeneous data and, relying on the principle of Minimum Description Length [Rissanen, 1978], integrates numerical and categorical information in clustering. The limitations of the INCONCO algorithm include that it assumes a known probability distribution model for each domain. Also, it assumes that the number of clusters must be identical and it is limited to two domains, specifically, categorical and numerical features. The k-means-mixed clustering algorithm is based on the k-means paradigm and works with mixed numerical and categorical features [Ahmad and Dey, 2007]. It uses a cost function and distance measure that are based on the co-occurrence of values. The distance measure also takes into account the significance of an attribute towards the clustering process. The definition of a cluster center contains the proportional distribution of different categorical values in the cluster. Hence, when the cost function computes the distance of an object from the existing cluster centers, the function inherently considers the significance of each attribute and is based on the probability of an element to be pulled towards a cluster depending on the distribution of the different attribute values present in the cluster. CAVE is a clustering algorithm based on variance and entropy, that is able to mine mixed data [Hsu and Chen, 2007]. The algorithm uses variance for measuring the similarity of numerical values and integrates entropy with distance hierarchies for measuring the similarity between categorical values. In particular, a distance hierarchy is composed of concept nodes and links; where higher-level nodes represent more general concepts while lower-level nodes represent more specific concepts. In addition, each link is associated with a weight representing a distance. The algorithm then aggregates the similarity quantities from the categorical and the numerical parts to compute the similarity values between the mixed data.

### 2.3.4 Multiview Clustering (MC)

The multiview setting typically applies to supervised learning problems that have a natural way to divide their features into subsets (views) each of which are sufficient to learn the target concept. Multiview algorithms train two independent hypotheses with bootstrapping by providing each other with labels for the unlabeled data [Blum and Mitchell, 1998]. The training algorithms tend to maximize the agreement between the two independent hypotheses and optimally combine the multiple views. In the rest of this section, we will use the terms *graph* and *view* interchangeably.

Bickel and Scheffer developed a multiview version of mixture-of-multinomials model based clustering for text data [Bickel and Scheffer, 2004]. For the estimation of mixture-of-multinomials model parameters, they use an Expectation Maximization (EM) approach. A drawback of the mixture-of-multinomials is that documents with equal composition of words but with different word counts yield different posteriors. To deal with this problem, they also introduce the multiview version of spherical k-means algorithm which normalizes each document vector to unit length. They start from randomly initialized concept vectors for each cluster and assign the documents that are closest to its concept vector to the corresponding partition in the first view. In the next step, they estimate the new concept vectors in the second view based on the clustering partition from the first view. Then based on the new concept vectors, they compute a new clustering partition in the second view. These steps keep alternating until the algorithm converges. Thus, at each step, a clustering partition from one view is replaced by a clustering partition from another view.

Aside from Bickel and Scheffer [Bickel and Scheffer, 2004], the remaining multiview clustering algorithms have been based on graph clustering. Besides the direct combination of graphs, [Abhishek and Hal, 2011] and [Abhishek et al., 2011] proposed to maximize the agreement between different views. Relying on the central idea that the clustering from one view should agree with the clustering from another view, they extended spectral clustering to multiple views based on the co-training idea [Blum and Mitchell, 1998]. Their approach is based on the assumption that the true underlying clustering would assign corresponding points in each view to the same cluster. First, they perform spectral clustering on individual graphs to get the discriminative eigenvectors in each view. Then they iteratively find a projection of the similarity matrix of the first view along the eigenvectors of the second view and vice-versa. Then, using the projections of the first and second views as the new graph of similarities, they compute the Laplacian and find updated values for the discriminative eigenvectors in both views. After the final values of the eigenvectors of both views are obtained, they select the most informative view and cluster the eigenvectors of the selected view

with the k-means algorithm.

A completely different approach was proposed in [Zhou and Burges, 2007], that first “combines” the two views/graphs and then proceeds with spectral clustering. They use a Markov random walk model to combine multiple graphs. Assuming a random walk with the current position being at a vertex in one graph, in the next step, the walker may continue her random walk in the same graph with a certain probability, or jump to the other graph with the remaining probability and continue her random walk there. A subset of vertices is regarded as a cluster if during the random walk, the probability of leaving this subset is small while the stationary probability mass of the same subset is large.

Another approach that was proposed in [de Sa, 2005] uses an algorithm for spectral clustering in the multiview setting where there are two independent subsets of dimensions, each of which could be used for clustering. The algorithm clusters the data in each view so as to minimize the disagreement between the clusterings. The main idea is that two (or more) networks receiving data from different views, but with no explicit supervisory label, should cluster the data in each view so as to minimize the disagreement between clusterings. Both views are combined into a bipartite graph, where the strength of the weight (Gaussian weighted normalized distance) between two nodes (patterns) in different views depends on the number of co-occurring pairs of patterns that are sufficiently close in both views. Using those weights, they define an affinity matrix which is then clustered by spectral graph clustering [Ng et al., 2001].

Finally, [Tang et al., 2009] presented a Linked Matrix Factorization (LMF) algorithm to find a shared partition of different views in both unsupervised and semi-supervised settings. In LMF, each graph is approximated by matrix factorization with a graph-specific factor and a factor common to all graphs, where the common factor provides features for all vertices. Then, vertices are clustered in the new feature space common for all views with a spectral clustering algorithm.

### **2.3.5 Ensemble Clustering (EC)**

The success of ensemble-based methods for supervised learning has motivated the development of ensemble methods for unsupervised learning. The basic idea of clustering ensembles is to combine multiple partitions into a single clustering solution. Clustering ensembles can go beyond what is typically achieved by a single clustering algorithm in several respects: (i) robustness: better average performance across the data sets; (ii) novelty: finding a combined solution unattainable by any single clustering algorithm; (iii) stability and confidence estimation: clustering solutions with lower sensitivity to noise, outliers, or sampling variations. This is because clustering uncertainty

can be assessed better from ensemble distributions; (iv) parallelization and scalability: the ability to integrate solutions from multiple distributed sources of data or features [Topchy et al., 2004a, Topchy et al., 2005].

Ensemble clustering must tackle three major problems which are specific to combination design:

- Consensus function: Unlike supervised classification, the patterns are unlabeled and therefore, there is no explicit correspondence between the labels delivered by different clusterings. An extra complexity arises when different partitions contain different numbers of clusters, often resulting in an intractable label correspondence problem. The optimal correspondence can be obtained using the Hungarian method for the minimal weight bipartite matching problem with  $O(k^3)$  complexity for  $k$  clusters [Kuhn, 1955, Frank, 2005].
- Diversity of clusterings: There are many different ways of generating a clustering ensemble and then combining the partitions. Multiple data partitions could be generated by: (i) applying different clustering algorithms, (ii) applying the same clustering algorithm with different values of parameters (different number of clusters, different number of neighbors, etc.) or initializations, and (iii) combining different data representations (different sets of features or different subsets of the original data) and clustering algorithms [Strehl and Ghosh, 2003], [Topchy et al., 2004b], [Hore et al., 2009].
- Cluster ensemble selection: Given a large library of clustering solutions, the goal of cluster ensemble selection is to choose a subset from the library to form a smaller cluster ensemble that performs as well as, or better than, using all available clustering solutions [Fern and Lin, 2008].

Clustering ensembles can also be used in multiobjective clustering as a compromise between individual clusterings with conflicting objective functions and plays an important role in distributed data mining [Strehl and Ghosh, 2003]. In [He et al., 2005], the authors proposed a divide and conquer technique to cluster data with mixed types of attributes. First, the original mixed data set is divided into two subsets: the *pure* categorical data set and the *pure* numerical data set. Next, an existing clustering algorithm designed to cluster a specific type of data is employed to cluster each subset separately and produce the corresponding clusterings. Last, the clustering results of the categorical and numerical data sets are combined as a categorical data set, on which the categorical data clustering algorithm is used to produce a final clustering.

The Weighted Cluster Ensembles method [Domeniconi and Al-Razgan, 2009] performs multiple clustering of the data in multiple subspaces of the input space, thus creating diverse partitions

that are later combined in an ensemble of weighted clusters. However, the goal of this method is not to tackle different domains or mixed data types, but rather to perform ensemble clustering in different subspaces of conventional data of the same type. Moreover, following the desiderata of all ensemble learning methods, this method actually strives to combine individual clustering results that are independent of one another, and thus are as diverse as possible. Also like all ensemble learning methods, there is no interaction or cooperation between the multiple domains during the cluster optimization process. This is exactly the opposite of our goal, which does not aim at ensemble clustering, but rather aims at performing clustering in each domain, but where the different domains actually do interact with each other to send and receive mutual guidance, "while" striving to obtain a better clustering in "each" domain, not only in the combined domains.

### 2.3.6 Collaborative Clustering (CC)

The problem of collaborative clustering can be defined as follows: "Given a finite number of disjoint data sites with data patterns defined in the same or different feature spaces, develop a scheme of collective development and reconciliation of a fundamental cluster structure across the sites that is based on exchange and communication of local findings where the communication needs to be realized at some level of information granularity" [Pedrycz and Rai, 2008]. One important feature is that sharing the raw data together is not allowed given restrictions of privacy or other technical reasons. However, some findings at the higher conceptual level of information granules could be shared between the collaborating data sites. Usually, the information granules are cluster membership partition matrices, constructed through fuzzy clustering [Dunn, 1973].

The main goal of collaboration is to give an ability for each node to benefit other nodes based on their needs. It is important to note that the collaborative approach aims only at enriching the *local* clustering solution of each individual node based on recommendations from other nodes. Thus, no "combined" solution is desired. This means that the goal of collaborative clustering is distinct from the goal of providing a clustering solution for the entire heterogeneous data set. In other words, collaborative clustering is centered on data being *distributed* over multiple sites.

Pedrycz [Pedrycz and Rai, 2009] proposed an algorithm where two underlying processes are run consecutively. It starts with fuzzy clustering procedures (FCM) [Bezdek et al., 1984] that are run independently at each data site for a certain number of iterations until convergence. Next, the data sites exchange the findings by transferring partition matrices, and afterward, an iterative process which optimizes the objective function takes place. After convergence, the partition matrices are exchanged between the data sites and the iterative computing of the partition matrices and the pro-

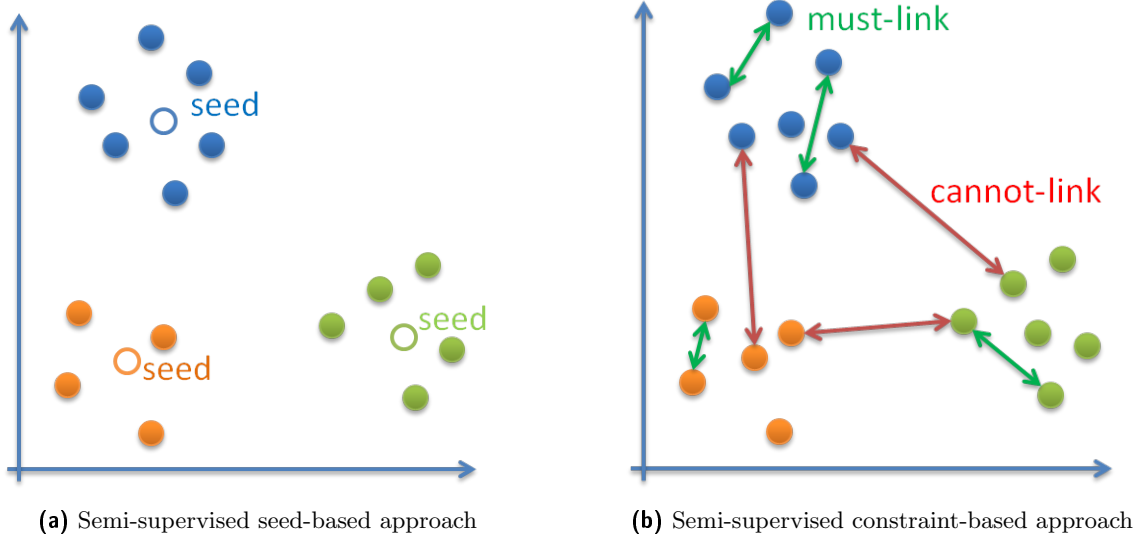
totypes resume. Another interesting CC approach was presented in [Hammuoda and Kamel, 2006] that proposed a distributed collaborative approach for document clustering. The main objective of this paper was to allow peers in a network to form independent opinions of local document grouping, followed by an exchange of cluster summaries in the form of key-phrase vectors. The nodes then expand and enrich their local solution by receiving recommended documents from their peers based on the peer judgement of the similarity of the local documents to the exchanged cluster summaries.

## 2.4 Semi-Supervised Clustering

Apart from clustering algorithms, which are unsupervised learners in the sense that they use *unlabeled* data, recent years have seen increasing interest in another direction, known as *semi-supervised learning (SSL)* which takes advantage of both labeled and unlabeled data. Many semi-supervised algorithms have been proposed including co-training, transductive support vector machines, entropy minimization, semi-supervised Expectation Maximization, graph-based approaches, and clustering-based approaches. In semi-supervised clustering, labeled data can be used in the form of

- *initial seeds* [Basu et al., 2002a],
- *constraints* [Wagstaff et al., 2001],
- *feedback* [Cohn et al., 2003].

All these existing approaches are based on model-based clustering [Zhong and Ghosh, 2003] where each cluster is represented by its centroid. *Seed-based* approaches use labeled data *only to help initialize* cluster centroids, while *constrained* approaches keep the grouping of labeled data unchanged throughout the clustering process, and *feedback-based* approaches start by running a regular clustering process and finally adjusting the resulting clusters based on labeled data (see Figure 2.2). Finally, it is worth mentioning that, although rooted in ideas of SSL, our IDS clustering framework is distinct. Semi-supervised clustering relies on *user-supplied labels*, whereas our proposed approach is completely unsupervised and thus does not rely on any external labels. Instead, it relies on selective, soft mutual guidance between the different domains of the data, while clustering.

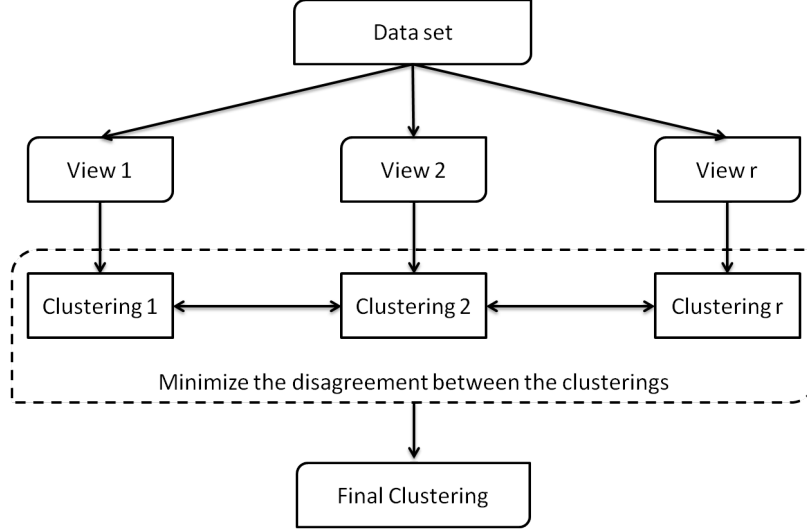


**Figure 2.2:** Seed-based and constraint-based semi-supervised clustering approaches

## 2.5 Limitations of Existing Methods and Comparison with the Proposed Work

### Using Multiview Clustering to Cluster Heterogeneous Data and Relationship to Our proposed Framework

It is clear that most of the multiview methods, reviewed above, could be used for clustering heterogeneous data, and in most cases for data that is expressed as a graph. In those cases, the graphs are combined either before or during clustering, based on the assumption that they are combinable (see Figure 2.3). However what if the graphs are not compatible on certain parts of the data? Such a situation is never hypothesized in MC algorithms, and therefore it cannot be handled. One exception to the graph-based MC is [Bickel and Scheffer, 2004] which works directly on document objects, not graphs, expressed in two views. However, one limitation of this approach is that the entire partition membership matrix is transferred to the other view after its convergence in its own view. It is easy to show that in case of incompatibility between views, this blind exchange will lead to instability, leading to an infinite cycle of exchanges of partitions between the different views without any improvement resulting from such an exchange. Thus, one limitation of existing MC methods is the insistence on enforcing “agreement” between the different aspects of the data. Such an assumption, when violated, may force incorrect results. In this dissertation, we propose an inter-domain compatibility analysis to improve the clustering of heterogeneous data.

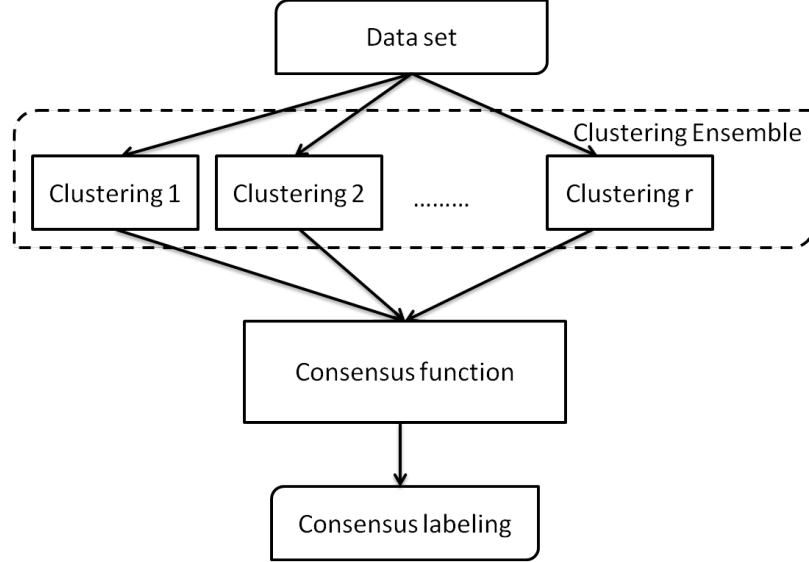


**Figure 2.3:** Multiview clustering.

### Using Ensemble Clustering to Cluster Heterogeneous Data and Relationship to our proposed Framework

EC can handle heterogeneous data by dedicating a different clustering process to each domain and aggregating the results within an ensemble framework (see Figure 2.4), which by intuition, emphasizes a consensus or agreement between the different domains, as was done in [He et al., 2005]. One limitation of EC methods would be in dealing with incompatibilities between the different domains. Our proposed Inter-Domain Supervision (IDS) approach may appear to be similar to ensemble-based clustering. However, one main distinction is that our approach enables the different algorithms running in each domain to reinforce or supervise each other *during* all the stages until the final clustering is obtained. In other words, our approach is more collaborative. Ensemble-based methods, on the other hand, were not intended to provide collaborative exchange of knowledge between different data “domains” *while* algorithms are still running, but rather to combine the *end* results of several runs or algorithms.

Also, even if the base clustering algorithms were distributed over different domains, EC methods do not provide any reliable individual clustering result from each domain on its own during the clustering process, but would rather require all the single-domain clusterings to complete and then be combined before having any viable clustering result that is ready for use. In contrast, our proposed IDS clustering approach works on producing reliable clustering in each domain from the very beginning of the clustering process; thus it is able to provide a reliable result even at intermediate stages, before all the clustering processes over all the domains are completed.



**Figure 2.4:** Ensemble clustering.

### Using Collaborative Clustering (CC) for Heterogeneous Data Clustering and Relationship to Our Proposed Framework

Although this was not the purpose of CC, one way to harness CC to cluster heterogeneous data is to consider each site as dedicated to only one pure domain of the data (see Figure 2.5). However, CC does not provide a “combined” clustering result, and similarly to MC and EC, makes an implicit assumption of necessary agreement between the different domains. To summarize, the main differences between collaborative clustering and our proposed IDS approach are:

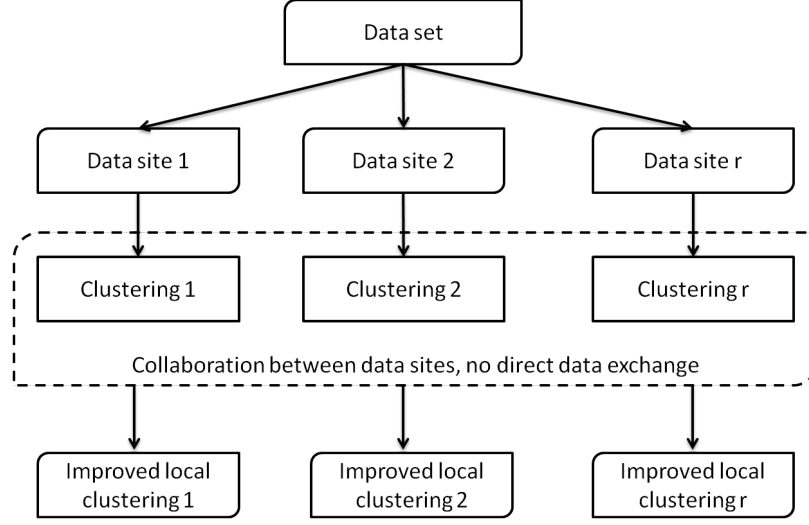
- in collaborative clustering, the data is physically distributed across different nodes or sites, and in fact, this is the main assumption that guides the clustering strategy,
- the data sets at the different sites have the same type of features,
- collaborative clustering seeks to improve the local clustering solution at each node or site and no final combined solution is desired.

Therefore, it is clear from the above distinctions that CC was designed to solve a problem that is distinct from our heterogeneous data clustering problem.

## 2.6 Clustering Evaluation

The procedure of evaluating the results of a clustering algorithm is often referred to as cluster validity.

In general terms, there are three approaches to investigate cluster validity [Halkidi et al., 2002].



**Figure 2.5:** Collaborative clustering.

The first approach is based on external criteria that assess the quality of the results of a clustering algorithm based on a pre-specified or ground-truth structure, reflecting our intuition or knowledge about the actual clustering structure of the data. Typically the ground-truth comes in the form of known data labels.

The second approach is based on internal criteria, that evaluate the clustering results based only how they fit the vectors of the data set themselves (e.g. distance or similarity matrix).

The third approach of clustering validity is based on relative criteria, meaning the evaluation of a clustering structure by comparing it to other clustering schemes, resulting by the same algorithm but with different input parameter values.

The first two approaches typically rely on statistical tests, or a computing validity score or index, and their major drawback is their high computational cost. Moreover, a typical validity index aims at measuring the degree to which a data set confirms same assumed distribution or structure. On the other hand, the third approach aims at finding the best clustering result that a clustering algorithm can define under certain assumptions and parameters. For example, it can be used to automatically determine an optimal number of clusters.

### 2.6.1 Internal index metrics

- The Davies-Bouldin (DB) index is a function of the ratio of the sum of within-cluster scatter to between-cluster separation [Davies and Bouldin, 1979]. Hence the ratio is small if the clusters are compact and far from each other. That is, the DB index will have a small value for a good

clustering. The DB index is defined as:

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{j:i \neq j} \frac{\delta_i + \delta_j}{\Delta_{ij}},$$

where  $\delta_i$  is the mean distance of the points belonging to cluster  $i$  to their centroid  $\mu_i$  and  $\Delta_{ij}$  is the distance between the centroids  $\mu_i$  and  $\mu_j$ .

- The Silhouette index is calculated based on the average silhouette width for each sample  $s(i)$ , average silhouette width,  $S_k$ , for each cluster and overall silhouette width,  $S$ , for the entire data set [Rousseeuw, 1987]. The average silhouette width for each sample  $s(i)$  is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

where  $a(i)$  is the average dissimilarity of data point  $x_i$  with the data within the same cluster and  $b(i)$  is the minimum over all clusters of the average dissimilarity of  $x_i$  with the data from each other cluster. The mean of the silhouette widths for a given cluster  $I_r$  is called cluster mean silhouette width and is defined as

$$S_r = \frac{1}{n_r} \sum_{i \in I_r} s(i),$$

where  $n_r$  is the number of data points in  $I_r$ . Finally, the global silhouette width or index for the entire data set is defined as the average of the mean silhouettes of all the clusters, as follows:

$$S = \frac{1}{k} \sum_{r=1}^k S_r.$$

Using this approach, each cluster can be represented by its silhouette, which is based on the comparison of its compactness and separation from other clusters. A silhouette value  $s(i)$  close to 1 means that the data sample is well-clustered and assigned to an appropriate cluster. A silhouette value close to zero means that the data sample could be assigned to another cluster, and the data sample lies halfway between both clusters. A silhouette value close to -1 means that the data sample is misclassified and is located somewhere in between the clusters.

- The Dunn index is based on the concept of cluster sets that are compact and well separated [Dunn, 1974]. The main goal of the measure is to maximize the inter-cluster distances and minimize the intra-cluster distances. The size or diameter of a cluster  $\Delta_r$  can be defined as

maximum distance between any two points inside a cluster  $r$ :

$$\Delta_r = \max_{x_i, x_j \in I_r} D(x_i, x_j).$$

Let  $\delta_{rr'}$  be the distance between clusters  $r$  and  $r'$  and defined as follows:

$$\delta_{rr'} = \min_{x_i \in I_r, x_j \in I_{r'}, r \neq r'} D(x_i, x_j).$$

Then the Dunn index with the  $k$  clusters is defined as:

$$DI = \frac{\min_{r \neq r'} \delta_{rr'}}{\max_{1 \leq r \leq k} \Delta_r}.$$

A higher value of the Dunn index means a better clustering.

- The Xie-Beni (XB) index is an index of fuzzy clustering, but it also can be used in crisp clustering [Xie and Beni, 1991]. It is defined as the ratio of the mean quadratic distance between every point and its cluster centroid to the minimum distance between cluster centroids:

$$XB = \frac{1}{N} \frac{\sum_{i=1}^N c_{il_i}^2 D(x_i, \mu_{l_i})^2}{\min_{l \neq l'} D(\mu_l, \mu_{l'})^2},$$

where  $c_{il_i}$  is the fuzzy membership (or in case of crisp clustering, crisp membership) of data point  $i$  and  $\mu_{l_i}$  is the cluster centroid of cluster  $l_i$ . A lower value of the XB index means a better clustering.

## 2.6.2 External index metrics

External metrics are only used if the external ground-truth class labels are available with the data.

- Purity is a simple evaluation measure that assumes that an external class label is available to evaluate the clustering results. First, each cluster is assigned to the class which is most frequent in that cluster, then the accuracy of this assignment is measured by the ratio of the number of correctly assigned data samples to the number of data points. A bad clustering has purity close to 0, and a perfect clustering has a purity of 1. Purity is very sensitive to the number of clusters; in particular, purity is 1 if each point gets its own cluster [Manning et al., 2008].
- Entropy is a commonly used external validation measure that measures the purity of the

clusters with respect to the given class labels [Shannon, 1948]. To find the entropy of the clustering, we compute the probability,  $p_{lr} = n_{lr}/n_r$ , that a member of cluster  $r$  belongs to class  $l$ , where  $n_r$  is the number of data points in cluster  $r$  and  $n_{lr}$  is the number of data points of class  $l$  in the cluster  $r$ . Then using the class distribution, the entropy of each cluster  $r$  is calculated using the standard entropy formula  $e_r = -\sum_{l=1}^L p_{lr} \log_2 p_{lr}$ , where  $L$  is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster:

$$E = \frac{1}{N} \sum_{r=1}^k n_r e_r,$$

where  $k$  is the number of clusters, and  $N$  is the total number of data points. A perfect clustering has an entropy close to 0 which means that every cluster consists of points with only one class label. A bad clustering has an entropy close to 1.

- Normalized mutual information (NMI) estimates the quality of the clustering with respect to a ground-truth class membership [Strehl et al., 2002]. It measures how closely the clustering algorithm could reconstruct the underlying label distribution in the data and is defined as follows

$$NMI = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}},$$

where  $I(X; Y) = H(X) - H(X|Y)$  is the mutual information between random variables  $X$  and  $Y$ ,  $H(X)$  and  $H(Y)$  are the marginal entropies, and  $H(X|Y)$  is the conditional entropy of  $X$  given  $Y$ . The minimum NMI is 0 if the clustering assignment is random with respect to class membership.  $X$  and  $Y$  represent the class label and cluster label, respectively. The maximum NMI is 1 if the clustering algorithm perfectly recreates the class memberships.

## 2.7 Chapter Summary and Discussion

In Table 2.2, we summarize the existing approaches for clustering mixed data types and give an overview of the proposed IDS framework. We distinguish them based on the following criteria:

- whether there is knowledge exchange between domains during the clustering process,
- the way they handle the clustering of mixed type data,
- the advantages of the clustering approach,
- the disadvantages or limitations of the clustering approach.

In the next chapter, we present the proposed IDS approach.

**Table 2.2:** Overview of clustering approaches.

Approach	Domain Integration	How they handle clustering of mixed-type data	Pros	Cons
Algorithms for specific types of data, Subsection 2.3.1.	<ul style="list-style-type: none"> <li>• Not integrated</li> <li>• No interaction between the domains</li> </ul>	Splitting into different domains. Conversion to one type or domain	Very simple and fast	<ul style="list-style-type: none"> <li>• Limited to a specific data type</li> <li>• Potential loss of information</li> <li>• Possible creation of artifacts in the data</li> <li>• Assumes the same number of clusters in all domains</li> </ul>

Approach	Domain Integration	How they handle clustering of mixed-type data	Pros	Cons
Combining distances into a single distance matrix, Subsection 2.3.2.	Fully integrated, but domains do not interact during the clustering process	Similarity function which combines different types of data	<ul style="list-style-type: none"> <li>• Simple</li> <li>• Can use existing algorithms for clustering the distance matrix</li> </ul>	<ul style="list-style-type: none"> <li>• Must devise a specialized distance or similarity function that can adequately combine both domains</li> <li>• Assumes the same number of clusters in all domains</li> <li>• Must worry about weighting the contribution of each domain to the distance computation</li> <li>• Assumes all domains are compatible with each other</li> </ul>

Approach	Domain Integration	How they handle clustering of mixed-type data	Pros	Cons
Algorithms for mixed data, Subsection 2.3.3.	Fully integrated	Unified model that combines the clustering objectives into one cost function	Work only on specific combinations of data types	<ul style="list-style-type: none"> <li>• Limited to a specific data types and clustering algorithm</li> <li>• Limited to only two types or source of data</li> <li>• Assumes the same number of clusters</li> <li>• Assumes all domains are compatible one another</li> </ul>
Multiview clustering (MC), Subsection 2.3.4.	Fully integrated	In most cases, data from each domain is expressed as a graph and then graphs are combined together	<ul style="list-style-type: none"> <li>• A broad variety of existing MC methods</li> <li>• Can use an existing algorithms</li> </ul>	<ul style="list-style-type: none"> <li>• MC methods enforce “agreement” between the different aspects of the data</li> <li>• Assumes the same number of clusters</li> <li>• Assumes all domains are compatible one another</li> </ul>

Approach	Domain Integration	How they handle clustering of mixed-type data	Pros	Cons
Ensemble clustering (EC), Subsection 2.3.5.	Not integrated	By dedicating a different clustering process to each domain and aggregating the results within an ensemble framework	<ul style="list-style-type: none"> <li>• A broad variety of existing EC methods</li> <li>• Allows a different number of clusters</li> <li>• Can use any existing algorithms as a base clustering algorithm</li> </ul>	<ul style="list-style-type: none"> <li>• Ensemble size has to be at least 3</li> <li>• Assumes all domains are compatible one another to form a consensus</li> </ul>
Collaborative clustering (CC), Subsection 2.3.6.	Fully integrated via a combined objective function	Considers each site as dedicated to only one pure domain of the data	Can use an existing algorithm as a base clustering	<ul style="list-style-type: none"> <li>• CC only seeks to improve the local clustering solution at each node</li> <li>• Assume the same type of data at each site</li> <li>• Assumes all domains are compatible one another</li> </ul>

Approach	Domain Integration	How they handle clustering of mixed-type data	Pros	Cons
Proposed Inter-Domain Supervision (IDS) Framework, Chapter 3.	Can be selective about whether and for which part of the data to integrate	Separate domains mutually supervise each other within a SSL framework	<ul style="list-style-type: none"> <li>• Can adapt a broad variety of existing SS methods (constraint or seed)</li> <li>• Can handle a different number of clusters per domain</li> <li>• Can use any existing algorithm as the base learner</li> <li>• Performs selective integration of the domains in different data subsets depending on their compatibility for each subset</li> </ul>	To be determined

## CHAPTER 3

### METHODOLOGY

As we have concluded from the previous chapter, most of the current clustering approaches are limited to a particular data type or rely on a specific similarity function, which usually comes from a domain expert. During the clustering process they often assume the same number of clusters in each domain and assume a specific distribution model for each data domain [Plant and Böhm, 2011, Hsu and Chen, 2007, Ahmad and Dey, 2007, Huang, 1998b]. Most of the methods were not intended to provide a collaborative exchange of knowledge between the different data “domains” during the progression of the clustering algorithms, but rather combine the end results.

In this chapter, we propose a new methodology for clustering data comprising multiple domains or parts, in such a way that the separate domains mutually supervise each other within a semi-supervised learning framework. We call our approach Inter-Domain Supervision Clustering (IDS Clustering). Unlike current uses of semi-supervised learning, our methodology does not assume the presence of labels for part of the data; rather, that each of the different domains of the data separately undergoes an unsupervised learning process, while sending and receiving guidance information in the form of data constraints or seeds to/from the other domains. The entire process can be considered as an alternation of semi-supervised learning stages in the different data domains.

Our proposed IDS framework can use specifically designed clustering algorithms which can be distinct and specialized for each domain or type of data, however all the algorithms are bound together within a collaborative scheme:

1. For categorical data types, the algorithms k-modes [Huang, 1998a], ROCK [Guha et al., 2000], CACTUS [Ganti et al., 1999], etc, can be used.
2. For transactional or text data, the spherical k-means algorithm [Dhillon and Modha, 2001], or other specialized algorithm can be used.
3. For numerical data types, one can use the k-means [MacQueen, 1967], DBSCAN [Ester et al., 1996], or any other clustering algorithm for such data.

Symbol	Description
$T$	One single source or domain of the data (e.g: attribute of one type)
$\mathbf{M}_T$	The cluster membership matrix of domain $T$
$v_{M_T}^T$	A validity index vector computed for each cluster in the data domain $T$ using $M_T$
$xb_{M_T}^T$	The Xie-Beni index vector computed for each cluster in the data domain $T$ using $M_T$
$db_{M_T}^T$	The Davies-Bouldin index vector computed for each cluster in the data domain $T$ using $M_T$
$t$	The number of iterations in which there is no supervision between domains
$k$	The number of clusters
$k_T$	The number of clusters in domain $T$
$\mathbf{J}$	The Jaccard coefficient matrix
$N$	The number of objects in the data set
$X$	The set of data objects
$\mathbf{x}$	A data object or record
$U_{T,j}$	The set of points that belong to cluster $j$ in domain $T$
$D$	The distortion measure between the data points or objects
$M$	The set of must-link constraints
$C$	The set of cannot-link constraints
$W$	The set of violation costs for must-link constraints
$\bar{W}$	The set of violation costs for cannot-link constraints
$nc_T$	Number of constraints in domain $T$
$n_T$	Number of exchange points in domain $T$ , from which pairwise constraints would be send to another domain
$\mu$	A cluster representative or centroid
$L$	The set of cluster labels
$l$	A cluster label
$I()$	Indicator function, $I(x) = 1$ , iff $x$ is true and $I(x) = 0$ , otherwise

**Table 3.1:** List of notations

4. For graph data, one can use KMETIS [Karypis and Kumar, 1998], spectral clustering [Shi and Malik, 2000], or any other specialized algorithm for graphs.

In Section 3.1 and 3.2, we propose two different models for mutual supervision between different domains of the data: (i) via seed exchange and (ii) via constraints, respectively.

Then in section 3.4, we explore the role of compatibility between the different domains in heterogeneous data before applying our Inter-Domain Supervision clustering. Our findings indicate that a preliminary domain compatibility analysis step sets the stage for a more effective clustering of heterogeneous data that can exploit the synergy between the different domains in a more selective manner.

Table 3.1 lists the important notation that will be used throughout the rest of the chapter.

### 3.1 Seed-based Inter-Domain Supervision (Seed-based IDS)

#### 3.1.1 The Case of an Equal Number of Clusters in Each Data Type or Domain

The proposed seed-based IDS framework, can handle data records composed of two parts of data of any type, for example: numerical and categorical, numerical and transactional, text and visual, and etc. For the sake of simplicity, let's assume that first part of data consists of attribute of numerical type and the second part consists of attributes of categorical type. Our semi-supervised inspired framework consists of the following stages, as shown in Figure 3.1:

1. **Splitting Across Domains:** The first stage consists of dividing the set of attributes into two subsets: one subset, called domain  $T_1$ , with only attributes of numerical type (age, income, etc), and another subset, called domain  $T_2$ , with attributes of categorical type (eyes color, gender, etc).
2. **Baseline Clustering Per Domain:** The next stage is to cluster each subset using a specifically designed algorithm for that particular data type. In our experiments, we used k-means [MacQueen, 1967] for numerical type attributes  $T_1$ , and k-modes [Huang, 1998a] for categorical type attributes  $T_2$ . Both algorithms start from the same random initial seeds and run for a small number of iterations ( $t_n$  and  $t_c$  for k-means and k-modes, respectively), yielding (data-cluster) membership matrices  $M_{T_1}$  and  $M_{T_2}$ , respectively.
3. **Best Cluster Selection from All Domains:** In the third stage, we compare the cluster centroids obtained in the first domain,  $T_1$ , and the second domain,  $T_2$ , and find the best combination of both for each of the domains.
  - (a) **Cluster Matching:** First, we solve a cluster correspondence problem between the two domains using the Hungarian matching method [Frank, 2005, Kuhn, 1955] using as weight matrix, the entry-wise reciprocal of the Jaccard coefficient matrix, which is computed using the cluster memberships  $M_{T_1}$  and  $M_{T_2}$  of the  $T_1$  and  $T_2$  domains respectively.
  - (b) **Cluster Validation Across Domains:** Then using the membership matrices  $M_{T_1}$  and  $M_{T_2}$ , we compute cluster validity indices  $v_{M_{T_1}}^{T_1} \in \mathbb{R}^k$  and  $v_{M_{T_2}}^{T_1} \in \mathbb{R}^k$  in data domain  $T_1$  for each cluster centroid obtained respectively, from clustering the data in domain  $T_1$  and from clustering the data in domain  $T_2$  from the previous stage 2. Similarly, we also compute the same validity indices  $v_{M_{T_1}}^{T_2}$  and  $v_{M_{T_2}}^{T_2}$  in data domain  $T_2$  for each cluster

centroid obtained respectively, from clustering the data in domain  $T_1$  and from clustering the data in domain  $T_2$ . Note that we compute the cluster validity index  $vi$  not for the entire clustering but for each cluster centroids (seed) separately in each domain. In Section 3.1.2 we explore the role of different validity measures in the seed exchange process.

- (c) **Best Cluster Selection Across Domains:** To find the best combination of centroids for domain  $T_1$ , we compare  $v_{M_{T_1}}^{T_1}$  and  $v_{M_{T_2}}^{T_1}$  for each centroid resulting from clustering the data in domain  $T_1$  and resulting from clustering the data in domain  $T_2$ , and then take only those centroids which score a lower (or higher, see Section 2.6 for details) value in the validity index  $v$ , thus forming better clusters in one domain compared to the other. We then perform a similar operation for domain  $T_2$ . The outputs of this stage are two sets, each consisting of the best combination of cluster centroids or prototypes for each of the data domains  $T_1$  and  $T_2$ , respectively.

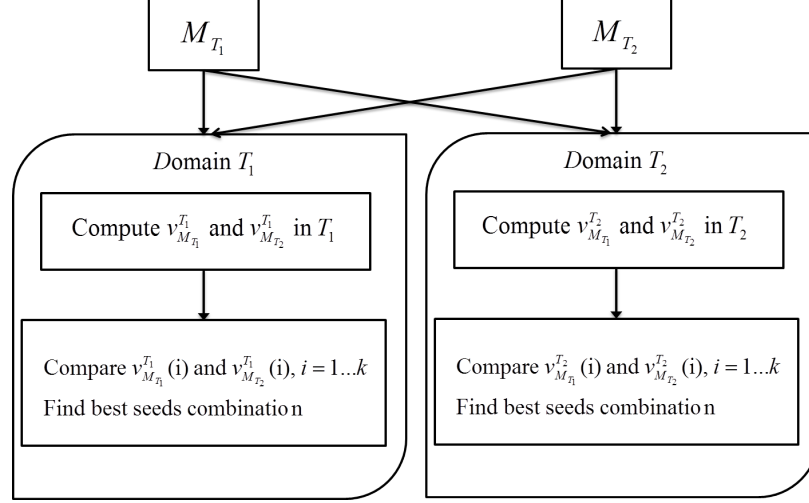
4. **Inter-Domain Supervised Clustering in Domain 1:** In this stage, we use the best seeds obtained from stage 3 to recompute the cluster centroids in the first domain by running k-means for a small number ( $t_n$ ) of iterations; then compare these recomputed centroids against the cluster centroids that were computed in the second domain in the previous iteration (as explained in detail in stage 3) and find the best cluster centroids' combination for the second domain ( $T_2$ ).
5. **Inter-Domain Supervised Clustering in Domain 2:** In this stage, we use the best seeds obtained from stage 4 to initialize the k-modes algorithm in domain  $T_2$ , and run it for  $t_c$  iterations. Then again, we compare these recomputed centroids against the cluster centroids computed in the first domain in the previous iteration (as explained in detail in stage 3) and find the best cluster centroids' combination for the first domain ( $T_1$ ).
6. We repeat stages 4 and 5 until both algorithms converge or the number of exchange iterations exceeds a maximum number.

### 3.1.2 Different Seed Exchange Mechanisms

In the previous section we presented an overview of the seed-based IDS approach. We now look at stage 3 in detail and consider there mechanisms for seed exchange:

- Normal or “blind” exchange. In this type of exchange mechanism, we do not look for the best possible seeds combination, instead, we blindly exchange seeds between domains. At every





**Figure 3.2:** Seed-based IDS, stage 3: best seeds combination selection.

obtained respectively, from clustering the data in domain  $T_1$  and from clustering the data in domain  $T_2$  from the previous stage, and repeat the same kind of procedure for domain  $T_2$ . Note that computing a DB index for every cluster centroid is essentially the same as computing the original overall DB index but without taking the sum over all centroids. To find the best combination of centroids for domain  $T_1$ , we compare  $db_{M_{T_1}}^{T_1}$  and  $db_{M_{T_2}}^{T_1}$  for each centroid resulting from clustering the data in domain  $T_1$  and resulting from clustering the data in domain  $T_2$ , and then take only those centroids which score a lower value in the DB index, thus forming better clusters in one domain compared to the other. We then perform a similar operation for domain  $T_2$ . The outputs of this stage are two sets, each consisting of the best combination of cluster centroids or prototypes for each of the data domains  $T_1$  and  $T_2$ , respectively, according to the DB validity index.

The general procedure of the seed exchange mechanism using a generic validity index ( $v$ ) is presented in Figure 3.2. In Section 4.2.1, we present experiments for the different exchange mechanisms using real life data sets.

### 3.1.3 Computational Complexity

The complexity of the proposed approach is mainly determined by the complexity of the embedded base algorithms used in each domain. In addition, there is an overhead complexity resulting from the coordination and alternating seed exchange process between the different domains during the mutual supervision process. The main overhead computation in the latter step is the cluster matching, validity scoring, and comparison performed in stage 3 (which is then repeatedly invoked at the

end of the subsequent stages 4 and 5). Stage 3 involves the following computations: first, the computation of the Jaccard coefficient matrix using the cluster memberships of the domains in time  $O(k^2N)$  (assuming the number of clusters to be of similar order  $k$ ), then solving the correspondence problem between the two domains using the Hungarian method in time  $O(k^3)$ , and finally, computing the DB (or XB) validity indices for each cluster centroid in both domains in time  $O(k^2N)$ . Thus, the total overhead complexity of stage 3 is  $O(k^2N)$  since  $k \ll N$ . With the k-means and k-modes as the base algorithms, the total computational complexity of the proposed approach is  $O(N)$ .

### 3.1.4 The Case of a Different Number of Clusters or Different Cluster Partitions in each Data Type or Domain

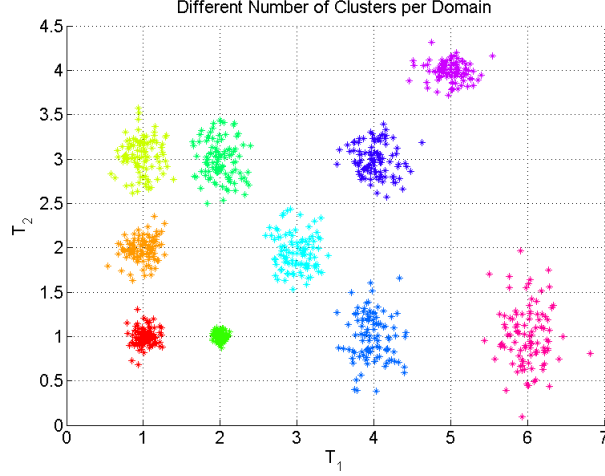
In our current design above, the number of clusters is assumed to be the same in each domain. This can be considered as the most basic default approach, and has the advantage of being easier to design. However, for clustering real life data, there are two challenges:

- Case 1: The first challenge is when each data domain naturally gives rise to a different number of clusters, which is simple to understand.
- Case 2: The second challenge is when regardless of whether the number of clusters are similar or different in the different domains, their nature is actually completely different, and this will be illustrated with the following example.

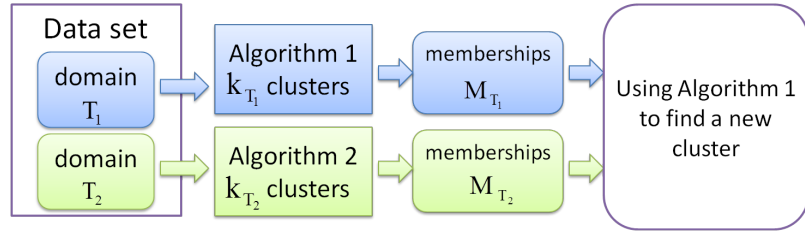
How do we combine the results of clustering in different domains if the numbers of clusters are different? Let us look at the example shown in Figure 3.3, which for visualization purposes, artificially splits two numerical features into two distinct domains, thus illustrating the difficulties with mixed domains. Here we have two domains or (artificially different) data types  $T_1$  and  $T_2$ . In total, taking into account both data domains or types  $T_1$  and  $T_2$ , we have ten distinct clusters, however if we cluster each domain separately, we see that in  $T_1$ , we have six clusters, while in  $T_2$ , we have only four clusters. This illustrates Case 2 and gives rise to the problem of judiciously combining the clustering results emerging from each domain into a coherent clustering result with correct cluster labelings for all the data points.

We propose the following algorithm to cluster such a data set, that we emphasize, *actually targets completely different data domains or types that cannot be compared using traditional attribute-based distance measures*. The stages of the algorithm are listed below:

1. **Split-Domain Clustering:** First, cluster  $T_1$  with  $k_{T_1}$  number of clusters and cluster  $T_2$  with  $k_{T_2}$  number of clusters. Let  $M_{T_1}$  be the cluster membership matrix of domain  $T_1$  and  $M_{T_2}$  be



**Figure 3.3:** Different number of clusters per domain.



**Figure 3.4:** An illustration of the split-domain clustering stage.

the cluster membership matrix of domain  $T_2$ . Therefore,  $M_{T_1}$  is an  $N \times k_{T_1}$  matrix and  $M_{T_2}$  is an  $N \times k_{T_2}$  matrix, where  $N$  is the number of data records. The membership matrix  $M_T$  is such that entry  $M_T[i, j]$  is 1 or 0 depending on whether or not point  $i$  belongs to cluster  $j$  in the current domain  $T$  (see Figure 3.4).

2. **Inter-Domain Cluster Matching:** Next, we compute the Jaccard coefficient matrix  $J$  of size  $k_{T_1} \times k_{T_2}$  in which entry  $J[j_1, j_2]$  is defined as follows:

$$J[j_1, j_2] = \frac{|U_{T_1, j_1} \cap U_{T_2, j_2}|}{|U_{T_1, j_1} \cup U_{T_2, j_2}|},$$

where  $U_{T, j}$  is the set of points that belong to cluster  $j$  in domain  $T$ , i.e.,

$$U_{T, j} = \{x_i | M_T(i, j) > 0\}.$$

3. **All-Domain Cluster Merging:** Finally, we merge the clustering results of domains  $T_1$  and  $T_2$  using Algorithm 1, where  $T_{max}$  is the domain with the highest number of clusters, i.e., with

$k_{T_{max}} = \max\{k_{T_1}, k_{T_2}\}$  and  $M_{T_{max}}$  is the membership matrix of that domain.  $T_{other}$  is the other domain with a number of clusters  $k_{T_{other}}$  ( $k_{T_{other}} \leq k_{T_{max}}$ ) and its membership matrix is  $M_{T_{other}}$ .

---

**Algorithm 1:** Merging Algorithm for domains differing in the number of clusters

---

```

input :  $J, M_{T_{max}}, M_{T_{other}}, k_{T_{max}}, \alpha_1, \alpha_2$ 
output:  $M_{merge}$ 
 $U_{ap} = \emptyset$ ;
for  $j_1 = 1$  to  $k_{T_{max}}$  do // for all clusters in the domain with more
    clusters
         $U_{T_{max},j_1} = \{x_i | M_{T_{max}}(i, j_1) > 0\}$ ; // find points in cluster  $j_1$  in this
            domain
         $candidates = \{j_2 | J(j_1, j_2) > \alpha_1\}$ ; // find the possible candidate clusters
            with domain intersection higher than  $\alpha_1$ 
        for  $All\ j_2 = 1 \in candidates$  do
             $U_{T_{other},j_2} = \{x_i | M_{T_{other}}(i, j_2) > 0\}$ ; // find points in cluster  $j_2$  from
                other domain
             $U_{cp} = U_{T_{max},j_1} \cap U_{T_{other},j_2}$ ; // find the common points between these
                two clusters
             $U_{ap} = U_{ap} \cup U_{cp}$ ; // common points already assigned to a cluster
             $U_{merge} = (U_{T_{max},j_1} \cap U'_{T_{other},j_2}) \cap U'_{ap}$ ; // points which belong to  $T_{max}$  but
                not in  $T_{other}$ , and were not assigned to a cluster
            if  $\frac{|U_{merge}|}{|U_{cp}|} > \alpha_2$  then // if intersection ratio is higher than noise
                level
                 $U_{new} = \{x_i | x_i \in U_{merge}\}$ ; // then assign intersection points to a
                    new cluster
                 $U_{T_{max},j_1} = U_{T_{max},j_1} - U_{new}$ ; // remove intersection points from
                    first cluster in this domain
            end
        end
    end
end

```

---

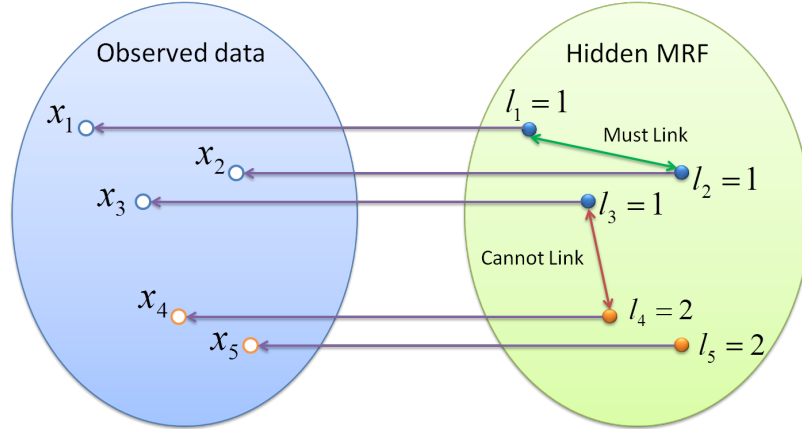
### 3.1.5 Computational Complexity

The complexity of the merging algorithm is mainly determined by stage 2, where we compute the Jaccard coefficient matrix  $J$  using the cluster memberships of the domains in time  $O(k_{T_1} k_{T_2} N)$ . The complexity of Algorithm 1 itself is  $O(k_{T_1} k_{T_2})$ . Thus, the total complexity of the merging algorithm is  $O(N)$ .

## 3.2 Constraint-based Inter-Domain Supervision

### 3.2.1 Mutual Inter-Domain Supervision using Hidden Markov Random Fields (HMRF): HMRF-KMeans

One of the leading methods for constrained-based semi-supervision is the HMRF-KMeans algorithm, that we use as a building block for our approach. The HMRF-KMeans algorithm [Basu et al., 2004] provides a principled probabilistic framework for incorporating supervision into prototype based clustering by using an objective function that is derived from the posterior energy of the Hidden Markov Random Fields framework for the constrained cluster label assignments. The HMRF consists of the hidden field of random variables with unobservable values corresponding to the cluster assignments/labels of the data, and an observable set of random variables which are the input data. The neighborhood structure over the hidden labels is defined based on the constraints between data point assignments (the neighbors of a data point are the points that are related to it via must-link or cannot-link constraints, see Figure 3.5). The HMRF-KMeans algorithm is an Expectation Max-



**Figure 3.5:** An illustration of the Hidden Markov Random Fields framework for the constrained cluster label assignments.

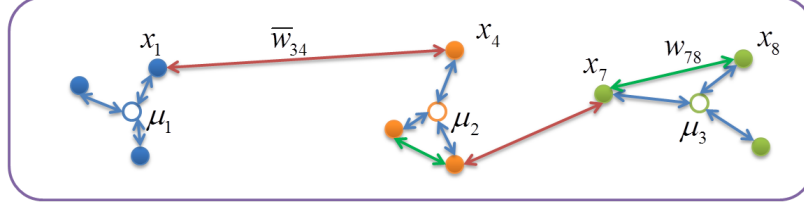
imization (EM) based partitional clustering algorithm for semi-supervised clustering that combines the constraint-based and distance-based approaches in a unified model. First, let us introduce the pertinent notation:  $X$  refer to a set of objects, whose representatives are enumerated as  $\{x_i\}_{i=1}^N$ ,  $x_{im}$  represents the  $m^{th}$  component of the  $d$ -dimensional vector  $x_i$ . This semi-supervised clustering model accepts as input a set of data points  $X$  with a specified distortion measure  $D$  between the points, and external supervision that is provided by a set of must-link constraints  $M = \{(x_i, x_j)\}$  (with its set of associated violation costs  $W$ ) and a set of cannot-link constraints  $C = \{(x_i, x_j)\}$  (with its

associated violation costs  $\bar{W}$ ). The goal of the algorithm is to partition the data into  $k$  clusters so that the total of the distortions  $D$  between the points and their corresponding cluster representatives  $\{\mu_h\}_{h=1}^k$  is minimized while violating a minimum number of constraints. The HMRF-KMeans objective function in (3.1) consists of four terms. The first term sums the distances between data objects and their corresponding cluster representatives. The second term adds a must-link violation penalty, which penalizes distant points that violate the must-link constraint higher compared to nearby points. This has the effect of penalizing the objective function to bring a pair of points that violate a must-link constraint closer to each other. Analogously, the next term represents the penalties for violating cannot-link constraints between pairs of data points thus encouraging the distance learning step to put cannot-linked points farther apart. Finally, the last term represents a normalization constant. The objective function [Basu et al., 2004] is given by

$$\begin{aligned} J_{obj} = & \sum_{x_i \in X} D(x_i, \mu_{l_i}) + \sum_{(x_i, x_j) \in M} w_{ij} \phi_D(x_i, x_j) I[l_i \neq l_j] \\ & + \sum_{(x_i, x_j) \in C} \bar{w}_{ij} (\phi_{D_{max}} - \phi_D(x_i, x_j)) I[l_i = l_j] + \log Z, \end{aligned} \quad (3.1)$$

where  $D(x_i, \mu_{l_i})$  is the distortion between  $x_i$  and  $\mu_{l_i}$ ,  $w_{ij}$  is the cost of violating the must-link constraint  $(i, j)$ ,  $\phi_D(x_i, x_j)$  is the penalty scaling function, chosen to be a monotonically increasing function of the distance between  $x_i$  and  $x_j$  according to the current distortion measure  $D$ .  $I$  is the indicator function ( $I(true) = 1$ ,  $I(false) = 0$ ), so that the must-link term is active only when cluster labels of  $x_i$  and  $x_j$  are different. In the next term,  $\bar{w}_{ij}$  is the cost of violating the cannot-link constraint  $(i, j)$ ,  $\phi_{D_{max}}$  is the maximum value of the scaling function  $\phi_D$  for the data set, and  $Z$  is a normalization constant (see Figure 3.6). Thus, the task is to minimize  $J_{obj}$  over cluster representatives  $\{\mu_h\}_{h=1}^k$ , cluster label configuration  $L = \{l_i\}_{i=1}^N$  (every  $l_i$  takes values from the set  $\{1, \dots, k\}$ ), and  $D$  (if the distortion measure is parameterized). Many distortion measures can be parameterized [Xing et al., 2002] and integrated into the HMRF-KMeans algorithm. In this work, we do not parametrize any distortion measure, and instead keep it as a function only of the data objects  $D = D(x_i, x_j)$ .

The main idea of HMRF-KMeans is as follows: in the E-step, given the current cluster representatives, every data point is re-assigned to the cluster that minimizes its contribution to  $J_{obj}$ . In the M-step, the cluster representatives  $\{\mu_h\}_{h=1}^k$  are re-estimated from the previous cluster assignments to minimize  $J_{obj}$  for the current assignment. The E-step and M-step are repeatedly alternated till a



**Figure 3.6:** An illustration of the HMRF-Kmeans [Basu et al., 2004] objective function. Blue arrows represent the distortion between data records and cluster centroids, green arrows represent the must-link constraints, and red arrows represent the cannot-link constraints.

specified convergence criterion is reached.

The HMRF-KMeans algorithm is flexible in the choice of the distortion measure  $D$ , however a single distortion measure must be used since the data is supposed to be of the same type or domain. In contrast, our data records consist of different domains, thus we will invoke several HMRF-KMeans processes one per domain, with each one receiving supervising constraints that were discovered in the other domains. For the sake of simplicity, we shall limit the data to consist of two parts in the rest of this paper: numerical and categorical. We start by dividing the set of attributes into two subsets: one subset, called domain  $T_1$ , with only attributes of one type, say numerical, such as  $T_1 = \{\text{age, income, ...}, \text{etc}\}$ , and a second subset, called  $T_2$ , with attributes of the other (say categorical) type such as  $T_2 = \{\text{eye color, gender, ...}, \text{etc}\}$ . The first subset consists of  $d_{T_1}$  attributes from domain  $T_1$  and the second subset consists of  $d_{T_2}$  attributes from domain  $T_2$ , such that that  $d_{T_1} + d_{T_2} = d$ , the total number of dimensions in the data. We use the Euclidean distance and simple matching distance  $\delta$  as a distortion measure  $D$  for the numerical and categorical domains, respectively. We also define the penalty scaling function  $\phi_D(x_i, x_j)$  to be equal to the corresponding distance function, and set the pairwise constraint violation costs  $W$  and  $\bar{W}$  to unit costs, so that  $w_{ij} = \bar{w}_{ij} = 1$  for any pair  $(i, j)$ .

Putting all this into (3.1) gives the following objective functions for the numerical domain  $T_1$ , with  $x_{im}$  denoting the  $m^{\text{th}}$  attribute of data record  $x_i$ ,

$$\begin{aligned}
 J_{T_1} = & \sum_{x_i \in X} \sqrt{\sum_{m \in T_1} (x_{im} - \mu_{im})^2} + \sum_{(x_i, x_j) \in M_{T_2}} \sqrt{\sum_{m \in T_1} (x_{im} - x_{jm})^2} I[l_i \neq l_j] \\
 & + \sum_{(x_i, x_j) \in C_{T_2}} (\phi_{D_{T_1, max}} - \sqrt{\sum_{m \in T_1} (x_{im} - x_{jm})^2}) I[l_i = l_j] + \log Z_{T_1}, \quad (3.2)
 \end{aligned}$$

and for the categorical domain  $T_2$ :

$$\begin{aligned}
J_{T_2} &= \sum_{x_i \in X} \sum_{m \in T_2} \delta(x_{im}, \mu_{l_i m}) + \sum_{(x_i, x_j) \in M_{T_1}} \sum_{m \in T_2} \delta(x_{im}, x_{jm}) I[l_i \neq l_j] \\
&+ \sum_{(x_i, x_j) \in C_{T_1}} (d_{T_2} - \sum_{m \in T_2} \delta(x_{im}, x_{jm})) I[l_i = l_j] + \log Z_{T_2}.
\end{aligned} \tag{3.3}$$

where  $M_{T_i}$  is a set of must-link constraints inferred based on the clustering of domain  $T_i$ , and  $C_{T_i}$  is a set of cannot-link constraints inferred based on the clustering of domain  $T_i$ . We further set the normalization constants  $Z_{T_1}$  and  $Z_{T_2}$  to be constant throughout the clustering iterations, and hence drop these terms from Equations 3.2 and 3.3.

In the seed-based mutual-supervision approach in Section 3.1, the number of clusters was assumed to be the same in each domain. This can be considered as the default approach, and has the advantage of being easier to design. However, in real life data, the different domains can have different numbers of clusters. One advantage of the constraint-based supervision used in the new methodology presented in this paper, is that it naturally solves the problem of clustering domains with different numbers of clusters.

### 3.2.2 Algorithm Flow

Our initial implementation, described below, can handle data records composed of two parts (such as numerical and categorical) within a semi-supervised inspired framework that consists of the following stages as shown in Figure 3.7:

1. **Domain Splitting:** The first stage consists of dividing the set of attributes into two subsets: one subset, called domain  $T_1$ , with only attributes of one type, e.g. numerical, (age, income, etc), and another subset, called domain  $T_2$ , with attributes of another type, e.g. categorical (eyes color, gender, etc).
2. **Baseline Clustering in the First Domain:** The next stage is to cluster one of the subsets  $T_1$  or  $T_2$  with the HMRF-KMeans algorithm without any constraints. Ideally, we try to start from the most promising domain in terms of data quality and guiding the clustering process, let us for simplicity assume that we start with domain  $T_1$ . The HMRF-KMeans algorithm runs for a small number of iterations  $t_{T_1}$  and yields a set of  $k_{T_1}$  cluster representatives  $\{\mu_h\}_{h=1}^{k_{T_1}}$  in that domain by minimizing Equation 3.2 with no constraints coming from the other domain, i.e.  $C_{T_2} = M_{T_2} = \emptyset$ .
3. **Inter-Domain Constraint Generation:** In the third stage, for each of the  $k_{T_1}$  cluster

representatives  $\mu_h$  we find the  $n_{T_1}$  closest points, according to the corresponding distance measure in domain  $T_1$ . Then using those  $k_{T_1} \times n_{T_1}$  points, we generate pairwise must-link constraints  $M_{T_1}$  using points that belong to the same cluster, and cannot-link constraints  $C_{T_1}$  using points that belong to different clusters. These constraints will later be sent to the clustering process in the other domain ( $T_2$ ) in the next stage.

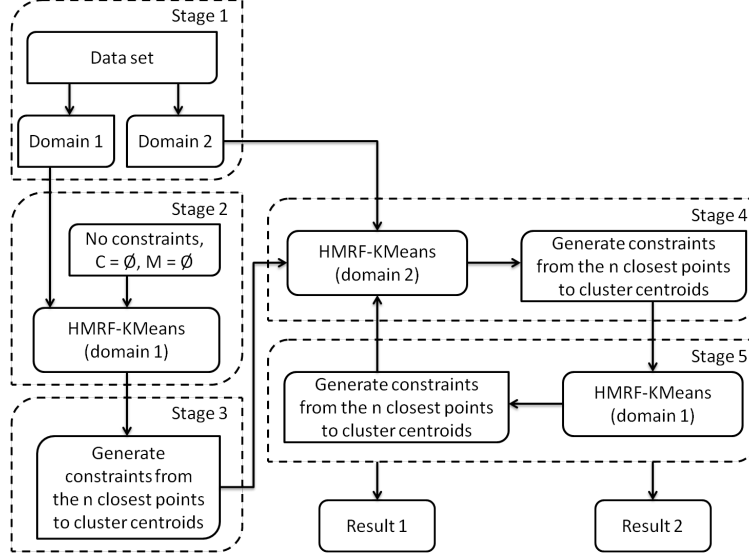
**4. Constraint-based Clustering on Domain 2 and New Constraint Generation:** In this stage, we cluster data in domain  $T_2$  with the HMRF-KMeans algorithm using the entire objective function penalized via the must-link constraints  $M_{T_1}$  and cannot-link constraints  $C_{T_1}$  obtained from the domain clustered in the previous stage. The HMRF-KMeans algorithm runs for a small number of iterations  $t_{T_2}$  and yields a set of cluster representatives  $\{\mu_h\}_{h=1}^{k_{T_2}}$  by minimizing Equation 3.3. Then again, for each cluster representative  $\mu_h$  we find the  $n_{T_2}$  closest points, according to the corresponding distance measure in domain  $T_2$ , and generate must-link constraints  $M_{T_2}$  and cannot-link constraints  $C_{T_2}$  using those points (as explained in detail in stage 3).

**5. Constraint-based Clustering on Domain 1 and New Constraint Generation:** Similarly, in the next stage, we use the previous domain's must-link constraints  $M_{T_2}$  and cannot-link constraints  $C_{T_2}$  obtained from stage 4 to penalize the objective function (3.2) in the HMRF-KMeans algorithm which runs for  $t_{T_1}$  iterations and yields a set of cluster representatives  $\{\mu_h\}_{h=1}^{k_{T_1}}$  by minimizing Equation 3.2. Then, for each cluster representative  $\mu_h$ , we recompute the  $n_{T_1}$  closest points, and generate must-link constraints  $M_{T_1}$  and cannot-link constraints  $C_{T_1}$  using those points.

We repeat stages 4 and 5 until both algorithms converge or the number of exchange iterations exceeds a maximum number.

### 3.2.3 Computational Complexity

The complexity of the proposed approach is mainly determined by the HMRF-KMeans algorithm, which incurs the heaviest cost during the initialization stage that uses both types of constraints and the unlabeled data to first compute the transitive closure on the must-link constraints to get connected components  $\lambda$ , consisting of points connected by must-link constraints [Basu et al., 2004], a procedure that costs  $O(N^3)$  time and  $O(N^2)$  space. Then for each pair of connected components with at least one cannot-link constraint between them, we add cannot-link constraints between every pair of points in that pair of connected components. This operation takes  $O(\lambda^2)$  time, thus  $O(k^2)$



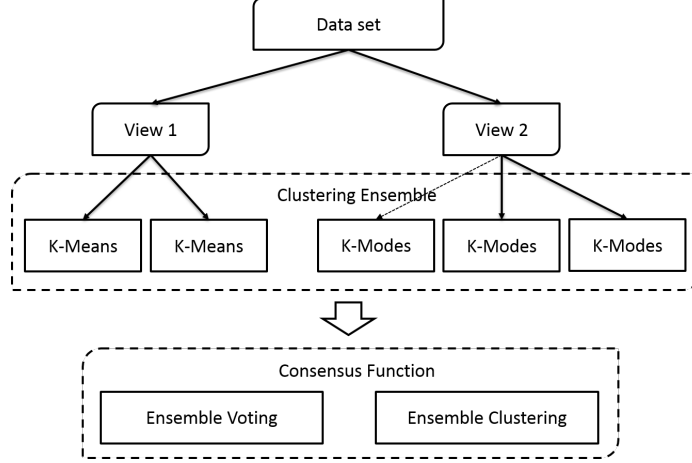
**Figure 3.7:** Outline of the mutual inter-domain supervision based heterogeneous data clustering using HMRF-KMeans.

of time, since  $\lambda$  is in the order of  $k$ . The second stage of the initialization is the cluster selection which is  $O(k^2)$ . The initialization step in the HMRF-KMeans is optional but essential for the success of the partitional clustering algorithm. The EM-based minimization of the HMRF-KMeans algorithm is  $O(N)$ . Finally, we need to account for the overhead complexity resulting from the process of coordination of and alternation of the constraint exchanges between the different domains during the mutual supervision process. This process finds the  $k \times n_T$  closest points to the cluster representatives in time  $O(N)$  for each domain, then generates the pairwise must-link and cannot-link constraints using those points in constant time. Thus the total computational complexity of the proposed approach is  $O(N^3)$  or  $O(N)$ , depending on whether we perform the initialization step with complete transitive closure or not, respectively.

### 3.3 How to use Other Existing Clustering Paradigms for the Purpose of Clustering Heterogeneous Data

#### 3.3.1 Ensemble Clustering

In Section 2.3.5, we described how ensemble clustering can be used to cluster heterogeneous data, by dedicating a different clustering process to each domain and aggregating the results within a consensus function. In our current implementation, we used 5 independent instances of two clustering algorithms. Figure 3.8 shows an example of the case, where we use 2 instances of k-means for numerical attributes and 3 instances of k-modes for the categorical attributes. Each instance of the



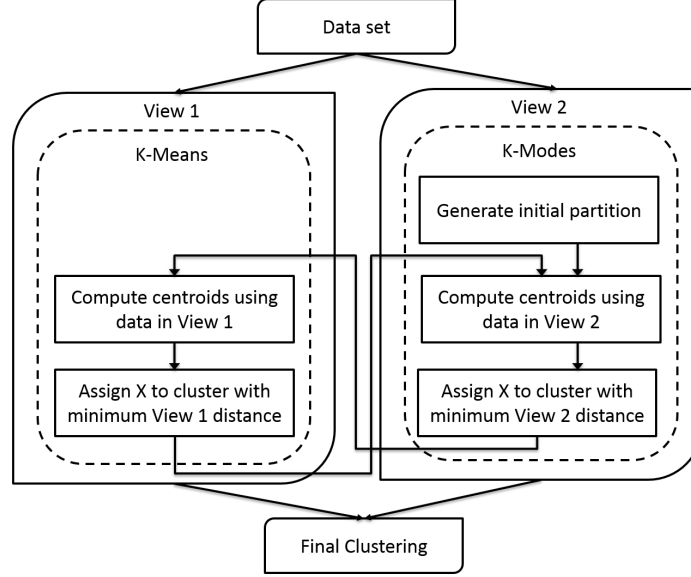
**Figure 3.8:** Ensemble clustering framework for clustering data with mixed attributes types.

algorithm had different initialization but the same number of clusters. As consensus function, we experimented with two approaches:

- **Voting:** The clustering results of the numerical instances of k-means and categorical instances of k-modes are combined as a categorical data set. Final clustering is produced through a majority vote of each record in the combined categorical data set. Unlike supervised classification, the patterns are unlabeled and therefore, there is no explicit correspondence between the labels delivered by different clusterings. We used Hungarian method to find optimal correspondence between clustering results of each instance.
- **Clustering:** The clustering results of the numerical instances of k-means and categorical instances of k-modes are combined as a categorical data set, on which the k-modes is used to produce a final clustering.

### 3.3.2 Multiview Clustering

Another competitive approach to the IDS framework is multiview clustering. We follow a similar idea to [Bickel and Scheffer, 2004], but instead of using only the spherical k-means for both views, we use a regular k-means for the numerical attributes, k-modes clustering algorithm for the categorical attributes (see Figure 3.9), and spherical k-means for the transactional like data (text and bag of words-visual domains in the MIRFlickr data set, see Chapter 4). After each iteration of the algorithm, we compute the objective function for each view, if the objective function did not change in the past three iterations, we terminate the optimization process. After termination, partitions  $\pi^{T_1}$  and  $\pi^{T_2}$  can be different. In order to compute the final clustering, first we compute the consensus



**Figure 3.9:** Multiview clustering framework for clustering data with mixed attributes types.

mean for each cluster and view. For the numerical and transactional like attributes it is defined as

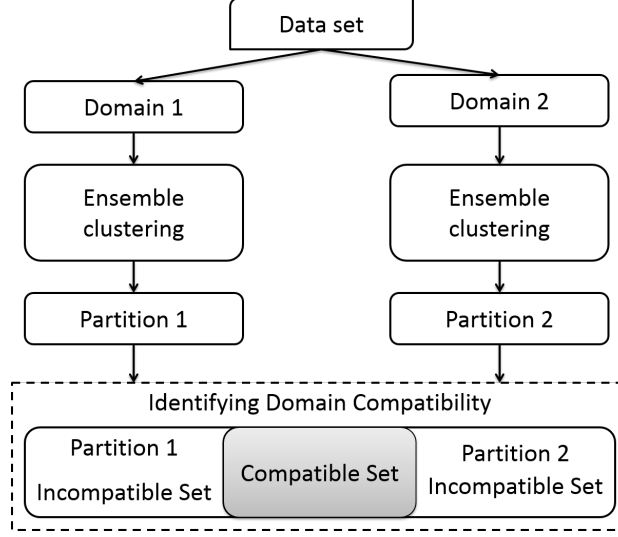
$$\mu_l^{T_1} = \frac{\sum_{x_i^{T_1} \in \pi_l^{T_1} \wedge x_i^{T_2} \in \pi_l^{T_2}} x_i^{T_1}}{\left\| \sum_{x_i^{T_1} \in \pi_l^{T_1} \wedge x_i^{T_2} \in \pi_l^{T_2}} x_i^{T_1} \right\|}, \quad (3.4)$$

and for the categorical attributes:

$$\mu_l^{T_2} = \text{Mode} \left( x_i^{T_1} \in \pi_l^{T_1} \wedge x_i^{T_2} \in \pi_l^{T_2} \right). \quad (3.5)$$

Then, based on the consensus centroids, we compute the final partition. We assign each observation to the cluster with the minimum sum of normalized Euclidean or matching distances in case of numerical or categorical types of attributes, respectively.

$$\begin{aligned} \pi_j &= \{x_i \in X : \frac{\sum_{m \in T_1} (x_{im} - \mu_{jm})^2}{\max_{x_a, x_b \in X, a \neq b} \left( \sum_{m \in T_1} (x_{am} - x_{bm})^2 \right)} + \frac{\sum_{m \in T_2} \delta(x_{im}, \mu_{jm})}{\max_{x_a, x_b \in X, a \neq b} \left( \sum_{m \in T_2} \delta(x_{am}, \mu_{bm}) \right)} \right. \\ &< \left. \frac{\sum_{m \in T_1} (x_{im} - \mu_{lm})^2}{\max_{x_a, x_b \in X, a \neq b} \left( \sum_{m \in T_1} (x_{am} - x_{bm})^2 \right)} + \frac{\sum_{m \in T_2} \delta(x_{im}, \mu_{lm})}{\max_{x_a, x_b \in X, a \neq b} \left( \sum_{m \in T_2} \delta(x_{am}, \mu_{bm}) \right)}, j \neq l\}. \end{aligned} \quad (3.6)$$



**Figure 3.10:** Domain Compatibility Analysis.

### 3.4 Discovering Domain Compatibility in Heterogeneous Data

Unlike semi-supervised learning where the external labels for some of the data comes with full certainty, the mutual supervision between different domains is naturally uncertain, and may even be misleading instead of supervising. This occurs when the domains are incompatible in how they represent the data. One important issue in clustering heterogeneous data is that the different domains may exhibit some compatibility (or agreement) for part of the data, while exhibiting incompatibility for the rest of the data. We call the set consisting of the first type of data, the *compatible* set, and call the set containing the rest of the data, the *incompatible* set.

Ideally, one would be motivated to build different descriptive or summarization models and different predictive models for the data depending on whether or not the data is deemed to be in the compatible set. That way, when data is available in different domains, these domains can be utilized to a full advantage in a judicious manner (separately or in combination) without forfeiting the abundance of data in the multiple domains. Therefore, to extend the methods in Section 3.1 and 3.2, we explore clustering the heterogeneous data separately depending on its compatibility status. In order to do this, we need to determine the domain compatibility.

For this purpose, we propose a method to identify the compatible and incompatible sets, based on performing the following three steps:

1. first, we cluster each domain with a reliable method that is unlikely to miss any clusters. We can use ensemble clustering [Dudoit and Fridlyand, 2003, Dimitriadou et al., 2001, Strehl and Ghosh, 2003]

for this purpose, although we have previously also investigated using Bisecting clustering, which did not perform as well as ensemble clustering;

2. then, we identify the corresponding clusters between the domains by solving a matching problem using the Hungarian method [Kuhn, 1955], which uses as input an inter-cluster matching weight inversely proportional to the Jaccard coefficient between the data membership assignment in each pair of clusters.
3. Finally, we compare the membership matrices and find the data records that were assigned to the same (corresponding) or different clusters. If a data record was assigned to corresponding clusters in both domains, it would indicate that the different domains agreed on this data, and if a data record was assigned to different clusters, this would indicate that the clusterings from different domains disagreed on this data, in which case it is considered part of the *incompatible* set.

The general flow of this procedure is presented in Figure 3.10 and Algorithm 2. Finally, we can apply the Inter-Domain Supervised clustering approach on each one of the two extracted data sets.

---

**Algorithm 2:** Finding compatible and incompatible sets

---

**input** : domain  $T_1$ , domain  $T_2$ , of data set  $X$   
**output:** Compatible set  $U_{comp}$ , Incompatible set  $U_{incomp}$   
 $U_{comp}, U_{incomp} = \emptyset$ ;  
Cluster  $T_1$  with Ensemble clustering -  $M_{T_1}$  ;  
Cluster  $T_2$  with Ensemble clustering -  $M_{T_2}$  ;  
Find cluster correspondence between  $M_{T_1}$  and  $M_{T_2}$  using the Hungarian method ;  
 $U_{comp} = \{x_i \in X | M_{T_1}(i, j) > 0, M_{T_2}(i, j) > 0\}$ ;  
 $U_{incomp} = X \setminus U_{comp}$ ;

---

### 3.5 Summary of the Chapter

In this chapter, we presented a seed-based inter-domain supervised approach to allow the transfer of information from the clustering in one domain to another. We also proposed a constraint-based inter-domain supervised approach to handle inconsistent partitions (different number of clusters) between different domains, which can now be combined into a consistent clustering result. We finally presented an approach for the domain compatibility analysis to help achieve a more effective clustering of heterogeneous data, that exploits the synergy between the different domains. In the next chapter, we will present our experiments to test the effectiveness of our approach compared to the most common existing techniques.

## CHAPTER 4

### EXPERIMENTAL RESULTS AND APPLICATION TO IMAGE ANNOTATION

In this chapter, we start by summarizing the evaluation metrics that will be used to validate our experimental results, outline our experimental plan, and then describe the benchmark data sets in Section 4.1. We report the experimental results for the seed-based IDS approach and the constraint-based IDS approach in Section 4.2. We then present an empirical study of domain compatibility analysis in Section 4.3. In Section 4.4, we apply our IDS methodology to automatic annotation of Flickr images based on clustering the images in two domains: visual features and text associated with the images. Finally, we summarize the chapter in Section 4.5.

The proposed inter-domain supervision (IDS) framework was evaluated using several internal and external clustering evaluation measures [Halkidi et al., 2002] (see Section 2.6). The characteristics of the evaluation measures are summarized in Table 4.1. Note that in calculating all internal indices, we used the same distance measures that were used in the clustering algorithms, namely, squared Euclidean distance for numerical data types, simple matching distance [Kaufman and Rousseeuw, 1990] for categorical data types, and cosine distance for asymmetric binary transactional data given in

Type	Validation Index	Minimum Value	Maximum Value	Value for “Perfect Clustering”
Internal	Davies-Bouldin (DB)	0	1	0
	Silhouette	−1	1	1
	Dunn	0	1	1
	Xie-Beni (XB)	0	$\infty$	0
External	Accuracy	0	1	1
	Precision	0	1	1
	Recall	0	1	1
	F-measure	0	1	1
	Purity	0	1	1
	Entropy	0	$\infty$	0
	NMI	0	1	1

**Table 4.1:** Overview of the clustering evaluation measures.

Equations 4.1.

$$\begin{aligned}
D_{sqE}(x_i, x_j) &= \sum_m (x_{im} - x_{jm})^2 \\
D_{cos}(x_i, x_j) &= 1 - \frac{\sum_m x_{im} x_{jm}}{\sqrt{\sum_m x_{im}^2 \sum_m x_{jm}^2}} \\
D_{binary}(x_i, x_j) &= \sum_m I(x_{im} \neq x_{jm}),
\end{aligned} \tag{4.1}$$

where  $x_{im}$  denotes the  $m^{th}$  attribute of data record  $x_i$  and  $I$  is the indicator function ( $I(true) = 1$ ,  $I(false) = 0$ ). In the following,  $N$  denotes the number of data points, and  $k$  is the number of clusters.

Note that for the MIRFlickr data set, we do not have an external class label, but rather a set of tags for each image. Thus, in addition to the regular validity indices, we also compute those same validity indices in the tag space (instead of the original data space) to capture how the clusters conform with the ground-truth tags for the data. These validity indices are referred to as *Tags DB*, *Tags Silhouette* and *Tags Dunn* in Table 4.13.

Tables 4.2 and 4.3 summarize the experiments that we performed in Sections 4.2 and 4.3, respectively. Tables 4.4 and 4.5 summarize the experiments performed in Section 4.4.

## 4.1 Real-Life Data Sets

We experimented with four real-life data sets with the characteristics shown in Table 4.6. The Adult, Credit approval, and Heart disease data sets were obtained from the UCI Machine Learning Repository [Frank and Asuncion, 2010] and the MIRFlickr25000 (MIRFlickr) data set was obtained from LIACS Medialab at Leiden University [Huiskes and Lew, 2008].

- *Adult Data.* The Adult data set was extracted by Barry Becker from the 1994 Census database. The data set has two classes: People who make over \$50K a year and people who make less than \$50K. The original data set consists of 48,842 instances. After deleting instances with missing and duplicate attributes, we obtained 45,179 instances. For detailed attribute description, see Table 4.7.
- *Heart Disease Data.* The Heart disease data, generated at the Cleveland Clinic, contains a mixture of categorical and numerical features. The data comes from two classes: people with no heart disease and people with different degrees of heart disease.

Section	Experiment	Data Set	Parameters	Figure or Table
4.2.1	Seed-based IDS: Value of the objective function with different seed exchange mechanisms	Adult	$k = 2$ , 1 run, maximum 2 seeds per domain.	Figure 4.2
		Heart disease		Figure 4.3
		Credit card approval		Figure 4.4
		MIRFlickr	$k = 16$ , 1 run, maximum 16 seeds per domain.	Figure 4.5
	Seed-based IDS: Clustering results for different seed exchange mechanisms	Adult	$k = 2$ , 10 runs, maximum 2 seeds per domain.	Table 4.10
		Heart disease		Table 4.11
		Credit card approval		Table 4.12
		MIRFlickr	$k = 16$ , 10 runs, maximum 16 seeds per domain.	Table 4.13
4.2.2	Constraint-based IDS: Effect of the number of constraints	Adult	$k = 2$ , 676 runs, $t_{T_1} = t_{T_2} = 1$ .	Figure 4.6
		Heart disease	$k = 2$ , 841 runs, $t_{T_1} = t_{T_2} = 1$ .	Figure 4.7
		Credit card approval		Figure 4.8
4.2.3	Value of the objective function for the seed-based IDS, constraint-based IDS, splitting, and multiview clustering algorithms	Adult	$k = 2$ , 1 run, maximum 2 seeds per domain, $n_{T_1} = n_{T_2} = 5$ , $t_{T_1} = t_{T_2} = 1$ .	Figure 4.9
		Heart disease		Figure 4.9
		Credit card approval		Figure 4.10
		MIRFlickr	$k = 16$ , 1 run, maximum 16 seeds per domain, $n_{T_1} = n_{T_2} = 5$ , $t_{T_1} = t_{T_2} = 1$ .	Figure 4.10
	Clustering results for the seed-based IDS, constraint-based IDS, splitting, conversion, ensemble, and multiview clustering	Adult	$k = 2$ , 10 runs, maximum 2 seeds per domain, $n_{T_1} = n_{T_2} = 5$ , $t_{T_1} = t_{T_2} = 1$ .	Table 4.14
		Heart disease	$k = 2$ , 50 runs, maximum 2 seeds per domain, $n_{T_1} = n_{T_2} = 5$ , $t_{T_1} = t_{T_2} = 1$ .	Table 4.15
		Credit card approval		Table 4.16
		MIRFlickr	$k = 16$ , 10 runs, maximum 16 seeds per domain, $n_{T_1} = n_{T_2} = 5$ , $t_{T_1} = t_{T_2} = 1$ .	Table 4.17

**Table 4.2:** Experimental plan overview for Section 4.2.

Experiment/Algorithm	Subsets of MIRFlickr data set	Parameters	Figure or table
Value of the objective function for the seed-based IDS	Compatible, incompatible	$k = 16$ , 1 run, maximum 16 seeds per domain.	Figure 4.13
Clustering results of the seed-based IDS	Mixed, incompatible, and compatible	$k = 16$ , 10 run, maximum 16 seeds per domain.	Table 4.18
Constrain-based IDS versus the baseline splitting algorithm: optimal number of constraints	Compatible	$k = 16$ , 196 runs, $t_{T_1} = t_{T_2} = 1$ .	Figure 4.14
Constrain-based IDS: optimal number of constraints	Compatible, mixed		Figure 4.15
Constrain-based IDS: optimal number of constraints	Compatible, incompatible		Figure 4.16
Value of the objective function for the constraint-based IDS	Compatible, incompatible	$k = 16$ , 1 run, $n_{T_1} = 11$ , $n_{T_2} = 5$ , $t_{T_1} = t_{T_2} = 1$ .	Figure 4.17
Clustering results of the constraint-based IDS	Mixed, incompatible, and compatible	$k = 16$ , 10 runs, $n_{T_1} = 11$ , $n_{T_2} = 5$ , $t_{T_1} = t_{T_2} = 1$ .	Table 4.19
Clustering results of the conversion algorithm		$k = 16$ , 10 runs.	Table 4.20
Clustering results of the splitting algorithm			Table 4.21
Clustering results of ensemble clustering		2 instance for the text domain, 3 instance for the visual domain, $k = 16$ , 10 runs.	Table 4.22
Clustering results of multiview clustering		1 view for the text domain, 1 view for the visual domain, $k = 16$ , 10 runs.	Table 4.23

**Table 4.3:** Experimental plan overview for Section 4.3.

MIRFlickr data set	Validation scheme	Algorithm	Parameters	Figure or table
The size of the training set is 22,430 data records, and the size of the test set is 1,000 data records.	Option 1: 10-NN to the cluster's centroid	Seed-based IDS	$k = 50, 100, 200$ , and 300 clusters, DB index-based seed exchange mechanism.	Table 4.24a shows $MAP_{f_{max}=3}$ and $MAP_{f_{max}=5}$ results. Figure 4.19a shows $MAP_{f_{max}=3}$ results.
			$k = 50, 100, 200$ , and 300 clusters, XB index-based seed exchange mechanism.	
			$k = 50, 100, 200$ , and 300 clusters, normal seed exchange mechanism.	
		Constraint-based IDS	$n_{T_1} = 11$ , $n_{T_2} = 5$ , $t_{T_1} = t_{T_2} = 1$ , $k = 50, 100, 200$ , and 300 clusters.	
		Conversion	$k = 50, 100, 200$ , and 300 clusters.	
		Multiview k-means	1 view for the text domain, 1 view for the visual domain, $k = 50, 100, 200$ , and 300 clusters.	
		Ensemble voting	2 instance for the text domain, 3 instance for the visual domain, $k = 16$ , 10 runs.	
		Splitting	$k = 50, 100, 200$ , and 300 clusters.	
	Option 2: 10-NN to the query image in the same cluster	Seed-based IDS	$k = 50, 100, 200$ , and 300 clusters, DB index-based seed exchange mechanism.	Table 4.24b shows $MAP_{f_{max}=3}$ and $MAP_{f_{max}=5}$ results. Figure 4.19b shows $MAP_{f_{max}=3}$ results.
			$k = 50, 100, 200$ , and 300 clusters, XB index-based seed exchange mechanism.	
			$k = 50, 100, 200$ , and 300 clusters, normal seed exchange mechanism.	
		Constraint-based IDS	$n_{T_1} = 11$ , $n_{T_2} = 5$ , $t_{T_1} = t_{T_2} = 1$ , $k = 50, 100, 200$ , and 300 clusters.	
		Conversion	$k = 50, 100, 200$ , and 300 clusters	
		Multiview k-means	1 view for the text domain, 1 view for the visual domain, $k = 50, 100, 200$ , and 300 clusters	
		Ensemble voting	2 instance for the text domain, 3 instance for the visual domain, $k = 50, 100, 200$ , and 300 clusters.	
		Splitting	$k = 50, 100, 200$ , and 300 clusters.	

**Table 4.4:** Experimental plan overview for Section 4.4: first set of the experiments.

Validation scheme	Algorithm	MIRFlickr data set	Parameters	Figure or table
Option 1: 10-NN to the cluster's centroid	Seed-based IDS	Training set: 2,629 data records from the compatible set, test set: 100 data records from the compatible and mixed sets.	$k = 16, 32, 50, 80$ , and 100 clusters, DB index-based seed exchange mechanism.	Table 4.25a shows $MAP_{f_{max}=3}$ and $MAP_{f_{max}=5}$ results. Figure 4.20a shows $MAP_{f_{max}=3}$ results.
			$k = 16, 32, 50, 80$ , and 100 clusters, XB index-based seed exchange mechanism.	
			$k = 16, 32, 50, 80$ , and 100 clusters, normal seed exchange mechanism.	
	Constraint-based IDS		$n_{T_1} = 11$ , $n_{T_2} = 5$ , $t_{T_1} = t_{T_2} = 1$ , $k = 16, 32, 50, 80$ , and 100 clusters.	
	Conversion	Training set: 2,629 data records from the mixed set, test set: 100 data records from the compatible and mixed sets.	$k = 16, 32, 50, 80$ , and 100 clusters.	
	Multiview k-means		1 view for the text domain, 1 view for the visual domain, $k = 16, 32, 50, 80$ , and 100 clusters.	
	Ensemble voting		2 instance for the text domain, 3 instance for the visual domain, $k = 16, 32, 50, 80$ , and 100 clusters.	
	Splitting		$k = 16, 32, 50, 80$ , and 100 clusters.	
Option 2: the 10-NN to a query image in a same cluster validation scheme.	Seed-based IDS	Training set: 2,629 data records from the compatible set, test set: 100 data records from the compatible and mixed set.	$k = 16, 32, 50, 80$ , and 100 clusters, DB index-based seed exchange mechanism.	Table 4.25b shows $MAP_{f_{max}=3}$ and $MAP_{f_{max}=5}$ results. Figure 4.20b shows $MAP_{f_{max}=3}$ results.
			$k = 16, 32, 50, 80$ , and 100 clusters, XB index-based seed exchange mechanism.	
			$k = 16, 32, 50, 80$ , and 100 clusters, normal seed exchange mechanism.	
	Constraint-based IDS		$n_{T_1} = 11$ , $n_{T_2} = 5$ , $t_{T_1} = t_{T_2} = 1$ , $k = 16, 32, 50, 80$ , and 100 clusters.	
	Conversion	Training set: 2,629 data records from the mixed set, test set: 100 data records from the compatible and mixed set.	$k = 16, 32, 50, 80$ , and 100 clusters.	
	Multiview k-means		1 view for the text domain, 1 view for the visual domain, $k = 16, 32, 50, 80$ , and 100 clusters.	
	Ensemble voting		2 instance for the text domain, 3 instance for the visual domain, $k = 16, 32, 50, 80$ , and 100 clusters.	
	Splitting		$k = 16, 32, 50, 80$ , and 100 clusters.	

**Table 4.5:** Experimental plan overview for Section 4.4: second set of the experiments.

Data set	# of Records	# of Attributes in Domain 1	# of Attributes in Domain 2	Missing Values	# of Classes	# of Clusters
Adult Data	45,179	6 (Numerical)	8 (Categorical)	Yes	2	2
Heart Disease Data	303	6 (Numerical)	7 (Categorical)	Yes	2	2
Credit Approval Data	690	6 (Numerical)	9 (Categorical)	Yes	2	2
MIRFlickr-25000 Data	23,430	2,105 (Text)	1,000 (Visual)	Yes	38	16

**Table 4.6:** Real-life data set properties.

#	Name	Data Type	Range or Values
1	Age	Numerical	[17, 90]
2	Work class	Categorical	Private, Self-emp-not-inc, Self-emp-inc, etc.
3	Final weight	Numerical	[13492, 1490400]
4	Education	Categorical	Bachelors, Some-college, 11th, HS graduate, etc
5	Education	Numerical	[1, 16]
6	Marital status	Categorical	Married, Divorced, Never married, etc.
7	Occupation	Categorical	Tech-support, Sales, Transport, etc
8	Relationship	Categorical	Wife, Husband, Not in family, Unmarried, etc.
9	Race	Categorical	White, Asian-Pacific-Islander, Black, etc
10	Sex	Categorical	Female, Male
11	Capital gain	Numerical	[0, 99999]
12	Capital loss	Numerical	[0, 4356]
13	Hours per week	Numerical	[1, 99]
14	Native country	Categorical	US, Cambodia, England, Canada, etc.

**Table 4.7:** Adult data set attribute description.

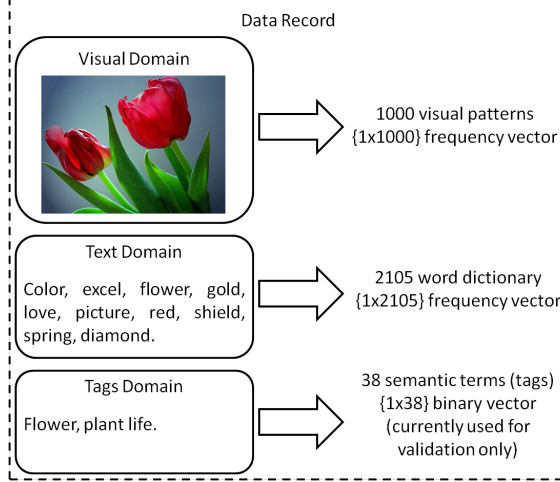
#	Name	Data Type	Range or Values
1	Age	Numerical	[29, 77]
2	Sex	Categorical	Female, Male
3	Chest pain type	Categorical	Typical angina, Atypical angina, etc.
4	Resting blood pressure in mm Hg	Numerical	[94, 200]
5	Serum cholestoral in mg/dl	Numerical	[126, 564]
6	Fasting blood sugar > 120 mg/dl	Categorical	True, False
7	Resting electrocardiographic results	Categorical	Normal, ST-T wave abnormality, etc.
8	Maximum heart rate achieved	Numerical	[71, 202]
9	Exercise induced angina	Categorical	Yes, No
10	ST depression	Numerical	[0, 6.2]
11	Slope of the peak exercise ST segment	Categorical	Up-sloping, Flat, Down-sloping
12	Number of major vessels	Numerical	[0, 3]
13	Thal	Categorical	Normal, Fixed defect, Reversible defect

**Table 4.8:** Heart disease data set attribute description.

#	Name	Data Type	Range or Values
1	A1	Categorical	b, a
2	A2	Numerical	[13.75, 80.25]
3	A3	Numerical	[0, 28]
4	A4	Categorical	u, y, l, t
5	A5	Categorical	g, p, gg
6	A6	Categorical	c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff
7	A7	Categorical	v, h, bb, j, n, z, dd, ff, o
8	A8	Numerical	[0, 28.5]
9	A9	Categorical	t, f
10	A10	Categorical	t, f
11	A11	Numerical	[0, 67]
12	A12	Categorical	t, f
13	A13	Categorical	g, p, s
14	A14	Numerical	[0, 2000]
15	A15	Numerical	[0, 100000]

**Table 4.9:** Credit card approval data set attribute description.

- *Credit Card Approval Data.* The data set has 690 instances, which were classified in two classes: approved and rejected. See Table 4.9 for details. Note that the attribute names and descriptions have been obfuscated on purpose to maintain the anonymity of the data subjects.
- *MIRFlickr Data.* The MIRFlickr-25000 image data set consists of 25,000 pictures and associated text, downloaded from the popular online photo-sharing service Flickr [Huiskes and Lew, 2008]. After removing missing values in both domains, we obtained 23,430 instances. The data set comes with the Flickr text description given by users, which can be considered as low level, noisy text. By processing this content, a 2,105-word dictionary is defined based on the most frequent terms [Caicedo et al., 2012]. The bag-of-features approach is used to represent visual content using a dictionary of 1,000 visual patterns which were extracted based on the image content. To do that, we used the same preprocessing steps as in [Caicedo et al., 2012]. Blocks of  $8 \times 8$  pixels were extracted from a set of training images with an overlap of 4 pixels along the  $x$ - and  $y$ -axes to build a set of training blocks. Each block is processed in the three RGB color channels using the discrete cosine transform (DCT) and the 21 largest coefficients per channel are used as features, leading to a block descriptor of 63 features with color and texture information [Monay and Gatica-Perez, 2007]. The k-means algorithm is applied to the block set to construct a vocabulary of 1,000 visual terms, which serve as reference vectors to quantize feature vectors extracted from blocks in any image. This image collection has also been manually annotated using a set of 38 semantic terms or tags provided as ground-truth for



**Figure 4.1:** Representation of a data record from the MIRFlickr data set.

validating information retrieval tasks. The annotation vector has binary elements indicating whether the photo can be described by the term or not. Figure 4.1 shows a sample from the data set.

The Adult, Heart disease, and Credit card approval data sets were clustered into 2 clusters, since each data set has 2 classes. The MIRFlickr data set was clustered in 16 clusters, since the value of the Silhouette index of the clustering results of Splitting algorithm with  $k = 16$  clusters had the highest value.

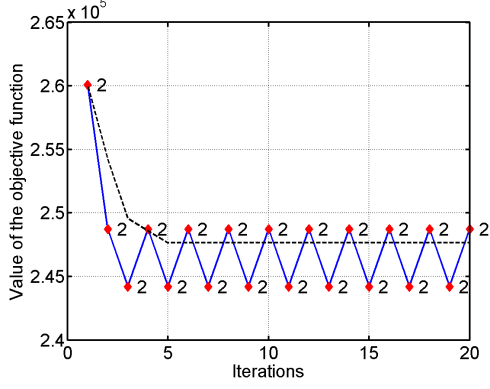
## 4.2 Results for the Inter-Domain Supervised Clustering

### 4.2.1 Results for the Seed-based IDS Clustering: Effect of the Seed Exchange Mechanism and Convergence

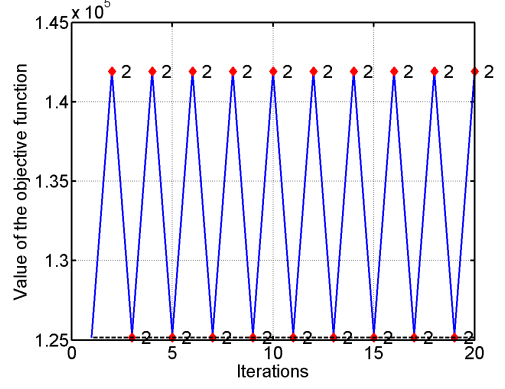
In Section 3.1.2, we proposed there types of seed exchange mechanisms. In this section, we present the convergence and exchange analysis and clustering performance of these mechanisms, evaluated on real life data sets. Figures 4.2, 4.3, 4.4, and 4.5 show how the value of the objective function changes with the number of iterations for the different exchange mechanisms for the seed-based IDS approach. For the numerical domain, we show the value of the k-means objective function; for the categorical domain, we plot the k-modes objective function; and for the text or visual domain, we show the value of the spherical k-means objective function. The seed-based IDS with no seed exchange is equivalent to the splitting approach (dashed line in Figures 4.2, 4.3, 4.4, and 4.5). Note that we obtained Figures 4.2-4.5 from on a typical run of the seed-based IDS approach. On the

contrary, Tables 4.10, 4.11, 4.12, and 4.13 show average results (in the format of  $\text{mean} \pm \text{std}[\text{min}, \text{median}, \text{max}]$  with the best results in a bold font) of the seed-based IDS clustering using these three exchange mechanisms based on 10 independent runs of the algorithm. Below, we analyze the results for each data set.

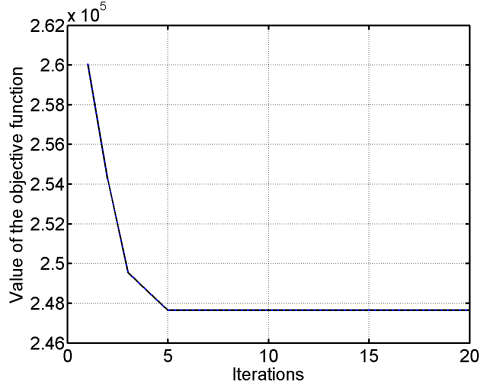
- Adult data set: Figures 4.2 (a,c,e) show the value of the k-means objective function with respect to the number of iterations for different seed exchange mechanisms, and Figures 4.2 (b,d,f) show the value of the k-modes objective function, also with respect to the number of iterations. Figures 4.2a and 4.2b present the results of the normal or “blind” seed exchange, while Figures 4.2c and 4.2d present the results of the XB index-based seed exchange, and Figures 4.2e and 4.2f show the results of the DB index-based seed exchange. As we can see from these figures, the proposed seed-based IDS approach with normal seed exchange exhibits an oscillating behavior. The value of the objective functions in the numerical and categorical domain keep moving from one local minimum to another (one of them happens to be better than the no-exchange baseline), and neither of the objective functions can reach convergence. In the numerical domain, the proposed approach reaches a lower value of the objective function compared to the objective function of the baseline (no-exchange) splitting algorithm, but sudden seed exchange between the domains forces the objective function to switch to another state. We observed the same kind of behavior in the categorical domain, as shown in Figure 4.2b. The XB and DB based seed exchange approaches give exactly the same results as the splitting algorithm, with an exception that with the XB index exchange in the categorical domain, there is a seed exchange between domains but that does not induce any change in the categorical objective function. Table 4.10 shows the results of the seed-based IDS using different exchange mechanisms for the Adult data set. As this table shows, the seed-based IDS with DB-based seed exchange performs better in both domains, showing significant improvements in all validation indices with the exception of entropy in the categorical domain, where normal exchange was the winner.
- Heart disease data set: Figures 4.3 (a-f) show very similar results to the results of the Adult data set. The normal seed exchange also shows oscillating behavior in both domains, but this time, the splitting algorithm performs better in both domains. The XB seed exchange outperforms the splitting algorithm with 2 seed exchanges in the first iteration in the numerical domain, and performs worse in the categorical domain with one seed replacement. The DB index based seed exchange outperforms all other seed exchange mechanisms and gives improved



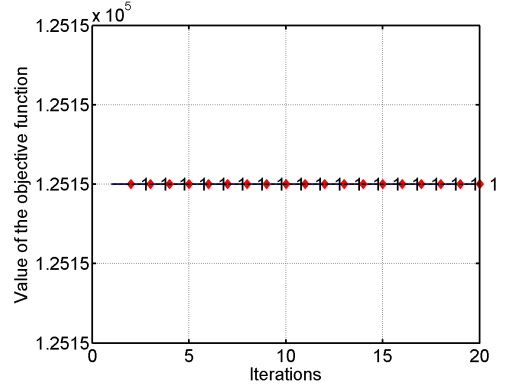
(a) Normal exchange: numerical domain



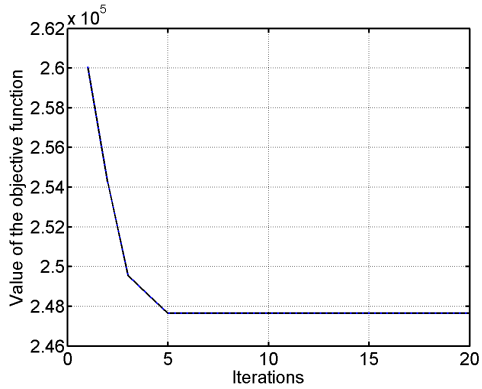
(b) Normal exchange: categorical domain



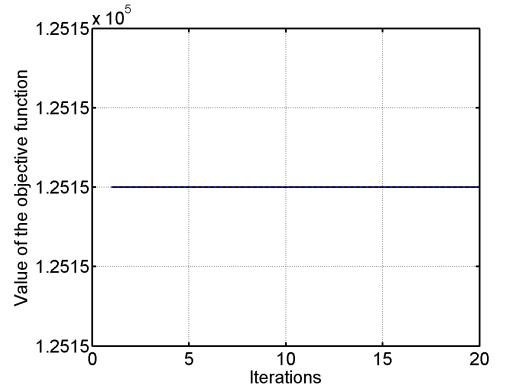
(c) XB exchange: numerical domain



(d) XB exchange: categorical domain

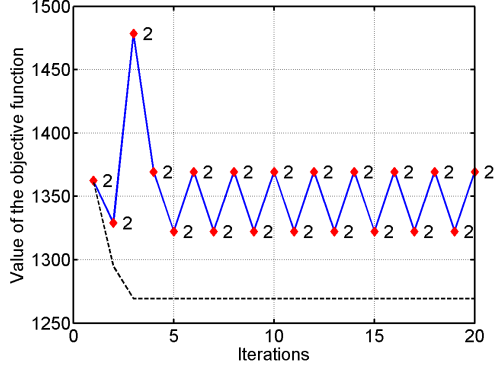


(e) DB exchange: numerical domain

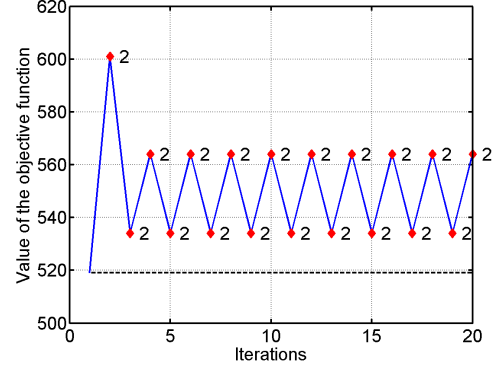


(f) DB exchange: categorical domain

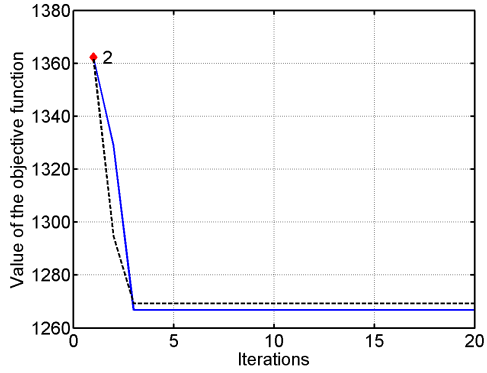
**Figure 4.2:** Value of the objective functions (with number of exchange seeds) for seed-based IDS with different exchange mechanisms for the Adult data set (dashed line: baseline splitting algorithm with no exchange).



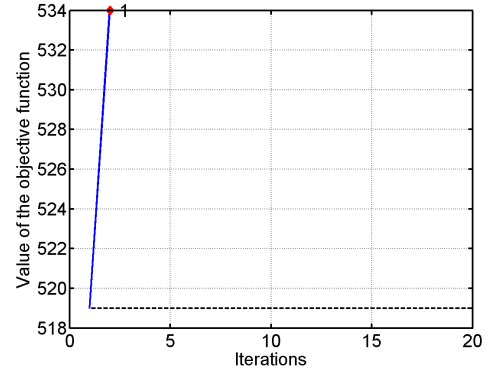
(a) Normal exchange: numerical domain



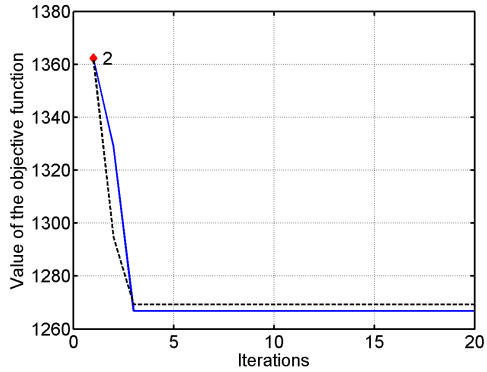
(b) Normal exchange: categorical domain



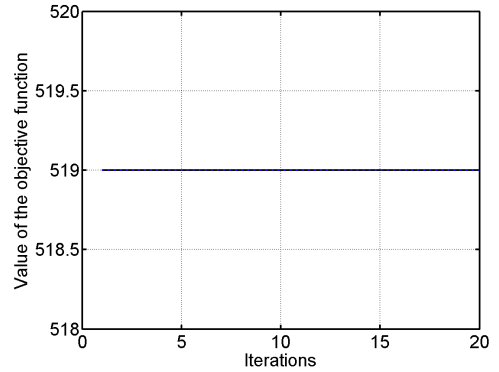
(c) XB exchange: numerical domain



(d) XB exchange: categorical domain

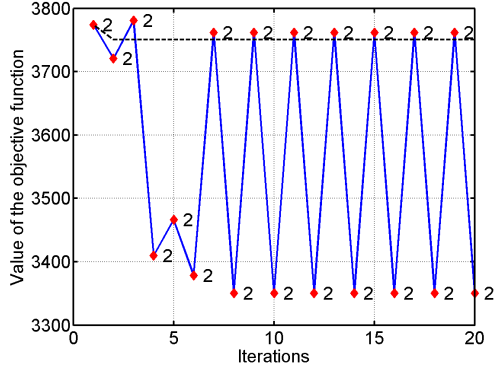


(e) DB exchange: numerical domain

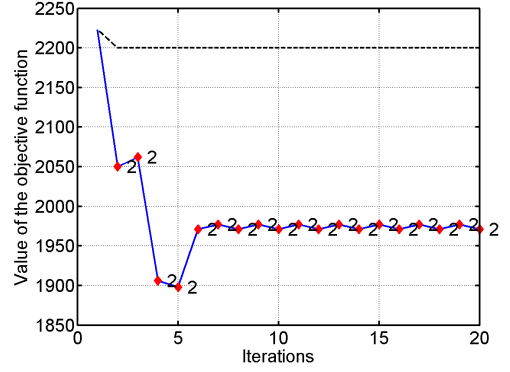


(f) DB exchange: categorical domain

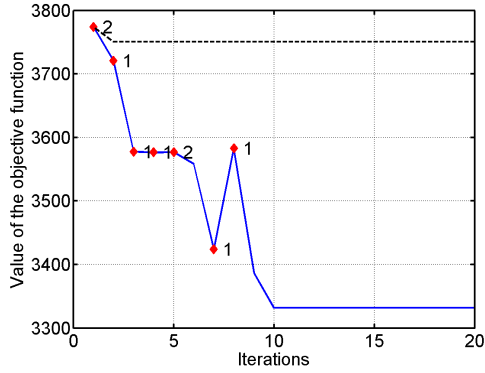
**Figure 4.3:** Value of the objective functions (with number of exchange seeds) for seed-based IDS with different exchange mechanisms for the Heart disease data set (dashed line: baseline splitting algorithm with no exchange).



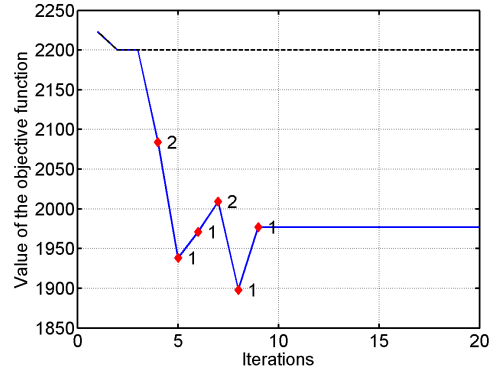
(a) Normal exchange: numerical domain



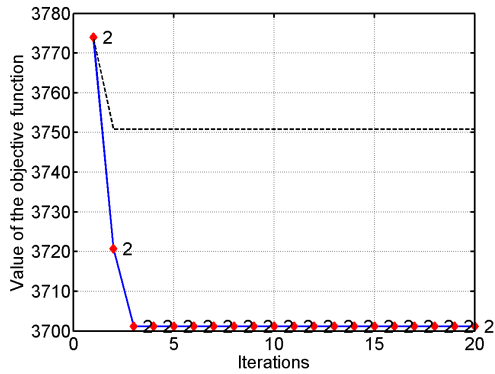
(b) Normal exchange: categorical domain



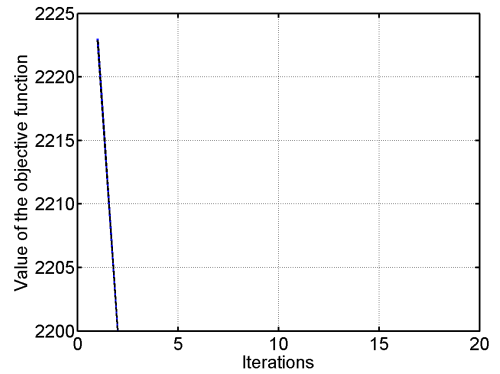
(c) XB exchange: numerical domain



(d) XB exchange: categorical domain

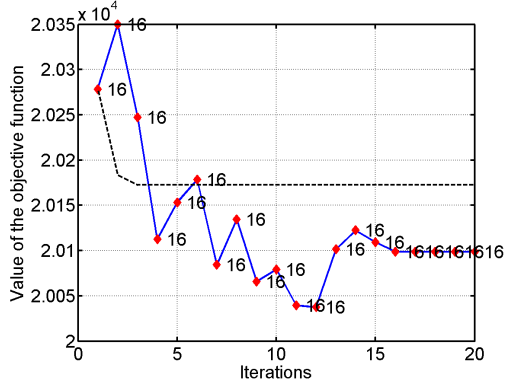


(e) DB exchange: numerical domain

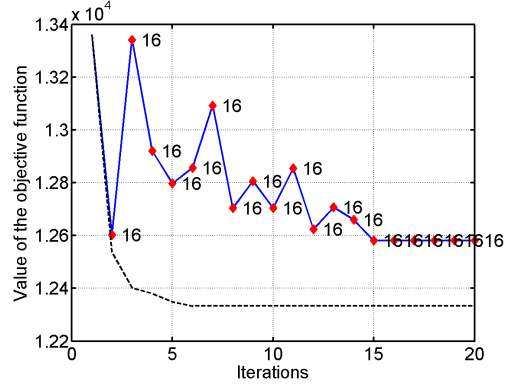


(f) DB exchange: categorical domain

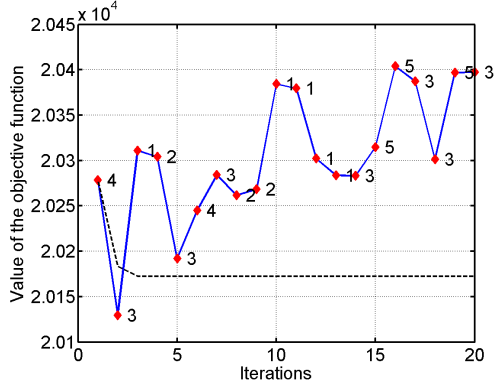
**Figure 4.4:** Value of the objective functions (with number of exchange seeds) for seed-based IDS with different exchange mechanisms for the Credit card approval data set (dashed line: baseline splitting algorithm with no exchange).



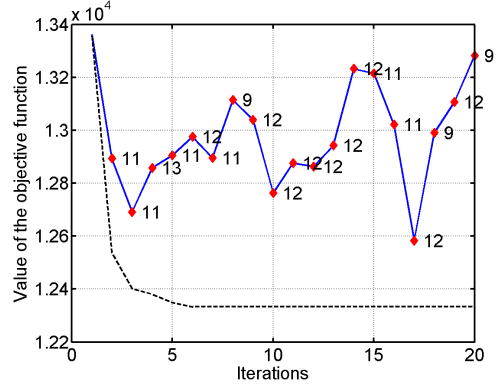
(a) Normal exchange: text domain



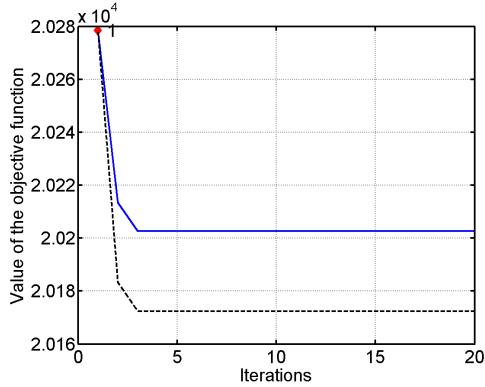
(b) Normal exchange: visual domain



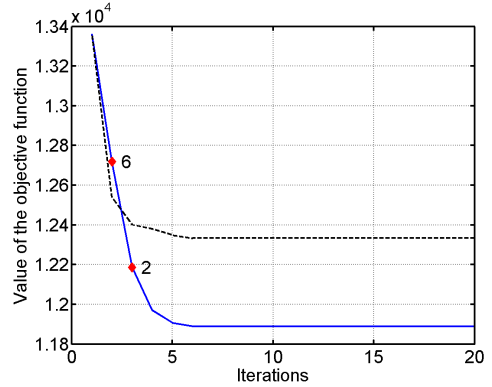
(c) XB exchange: text domain



(d) XB exchange: visual domain



(e) DB exchange: text domain



(f) DB exchange: visual domain

**Figure 4.5:** Value of the objective functions (with number of exchange seeds) for seed-based IDS with different exchange mechanisms for the MIRFlickr data set (dashed line: baseline splitting algorithm with no exchange).

Data type	Numerical		
Algorithm	Normal exchange	XB exchange	DB exchange
DB Index	$4.00 \pm 0.41$ [3.67, 4.00, 5.05]	$3.66 \pm 0.42$ [3.24, 3.63, 4.70]	<b><math>3.09 \pm 1.09</math></b> [0.42, 3.44, 3.77]
Silhouette Index	$0.22 \pm 0.03$ [0.17, 0.23, 0.27]	$0.24 \pm 0.07$ [0.19, 0.22, 0.43]	<b><math>0.29 \pm 0.18</math></b> [0.18, 0.21, 0.71]
Dunn Index	<b><math>0.0001 \pm 0</math></b> [0.0001, 0.0001, 0.0001]	<b><math>0.0001 \pm 0</math></b> [0.0001, 0.0001, 0.0001]	<b><math>0.0001 \pm 0</math></b> [0.0001, 0.0001, 0.0001]
Purity	$0.53 \pm 0.10$ [0.39, 0.55, 0.68]	$0.61 \pm 0.02$ [0.56, 0.60, 0.74]	<b><math>0.62 \pm 0.07</math></b> [0.52, 0.60, 0.75]
Entropy	$0.79 \pm 0.01$ [0.78, 0.79, 0.81]	<b><math>0.77 \pm 0.03</math></b> [0.70, 0.75, 0.78]	<b><math>0.77 \pm 0.03</math></b> [0.72, 0.78, 0.79]
NMI	$0.01 \pm 0.01$ [0.0003, 0.016, 0.03]	<b><math>0.06 \pm 0.03</math></b> [0.004, 0.05, 0.10]	<b><math>0.06 \pm 0.03</math></b> [0.02, 0.05, 0.10]
Data type	Categorical		
Algorithm	Normal exchange	XB exchange	DB exchange
DB Index	$1.34 \pm 0.08$ [1.12, 1.36, 1.40]	$1.33 \pm 0.25$ [1.10, 1.37, 1.92]	<b><math>1.22 \pm 0.14</math></b> [1.10, 1.12, 1.40]
Silhouette Index	$0.24 \pm 0.01$ [0.24, 0.25, 0.25]	<b><math>0.25 \pm 0.01</math></b> [0.23, 0.25, 0.27]	<b><math>0.25 \pm 0.01</math></b> [0.23, 0.24, 0.27]
Dunn Index	<b><math>0.125 \pm 0</math></b> [0.125, 0.125, 0.125]	<b><math>0.125 \pm 0</math></b> [0.125, 0.125, 0.125]	<b><math>0.125 \pm 0</math></b> [0.125, 0.125, 0.125]
Purity	$0.57 \pm 0.02$ [0.53, 0.56, 0.58]	$0.58 \pm 0.03$ [0.55, 0.59, 0.61]	<b><math>0.59 \pm 0.06</math></b> [0.50, 0.56, 0.67]
Entropy	<b><math>0.71 \pm 0.01</math></b> [0.71, 0.71, 0.73]	$0.72 \pm 0.01$ [0.69, 0.72, 0.73]	$0.73 \pm 0.02$ [0.71, 0.73, 0.78]
NMI	$0.08 \pm 0.01$ [0.06, 0.07, 0.09]	<b><math>0.09 \pm 0.01</math></b> [0.07, 0.10, 0.10]	<b><math>0.09 \pm 0.001</math></b> [0.08, 0.09, 0.11]

**Table 4.10:** Clustering results of seed-based IDS for the Adult data set with different seed exchange mechanisms (10 runs,  $k = 2$  clusters per domain).

results in the numerical domain, while giving similar results as the splitting algorithm in the categorical domain. As Table 4.3 shows, the DB index seed exchange outperforms all other methods in all evaluation metrics in both domains.

Data type	Numerical		
Algorithm	Normal exchange	XB exchange	DB exchange
DB Index	$1.92 \pm 0.17$ [1.62, 1.97, 2.26]	$1.78 \pm 0.15$ [1.63, 1.73, 2.02]	<b><math>1.73 \pm 0.15</math></b> [1.54, 1.71, 2.14]
Silhouette Index	$0.31 \pm 0.02$ [0.29, 0.30, 0.31]	$0.32 \pm 0.03$ [0.29, 0.32, 0.40]	<b><math>0.33 \pm 0.04</math></b> [0.26, 0.33, 0.41]
Dunn Index	$0.003 \pm 0.001$ [0.001, 0.003, 0.005]	$0.003 \pm 0.002$ [0.001, 0.004, 0.01]	<b><math>3.3e - 3 \pm 2.2e - 3</math></b> [ $1.2e - 5$ , $2.3e - 4$ , 0.35]
Purity	$0.68 \pm 0.02$ [0.66, 0.67, 0.73]	$0.71 \pm 0.01$ [0.69, 0.71, 0.73]	<b><math>0.72 \pm 0.03</math></b> [0.65, 0.72, 0.76]
Entropy	$0.88 \pm 0.04$ [0.81, 0.90, 0.92]	$0.85 \pm 0.01$ [0.81, 0.84, 0.88]	<b><math>0.84 \pm 0.03</math></b> [0.79, 0.84, 0.91]
NMI	$0.10 \pm 0.03$ [0.07, 0.08, 0.17]	$0.14 \pm 0.01$ [0.11, 0.14, 0.19]	<b><math>0.15 \pm 0.03</math></b> [0.08, 0.16, 0.20]
Data type	Categorical		
Algorithm	Normal exchange	XB exchange	DB exchange
DB Index	$0.81 \pm 0.02$ [0.79, 0.80, 0.82]	$0.78 \pm 0.11$ [0.65, 0.76, 0.99]	<b><math>0.76 \pm 0.01</math></b> [0.75, 0.76, 0.76]
Silhouette Index	$0.27 \pm 0.01$ [0.25, 0.26, 0.27]	$0.29 \pm 0.01$ [0.24, 0.29, 0.30]	<b><math>0.30 \pm 0.01</math></b> [0.29, 0.30, 0.31]
Dunn Index	$0.13 \pm 0.01$ [0.13, 0.13, 0.14]	<b><math>0.14 \pm 0</math></b> [0.14, 0.14, 0.14]	<b><math>0.14 \pm 0</math></b> [0.14, 0.14, 0.14]
Purity	$0.75 \pm 0.02$ [0.72, 0.74, 0.75]	$0.77 \pm 0.02$ [0.72, 0.77, 0.81]	<b><math>0.78 \pm 0.03</math></b> [0.71, 0.77, 0.81]
Entropy	$0.84 \pm 0.02$ [0.82, 0.84, 0.85]	$0.81 \pm 0.02$ [0.79, 0.81, 0.81]	<b><math>0.74 \pm 0.04</math></b> [0.70, 0.75, 0.87]
NMI	$0.21 \pm 0.01$ [0.12, 0.21, 0.27]	$0.24 \pm 0.02$ [0.16, 0.23, 0.28]	<b><math>0.25 \pm 0.04</math></b> [0.13, 0.24, 0.30]

**Table 4.11:** Clustering results of seed-based IDS for the Heart disease data set with different seed exchange mechanisms (50 runs,  $k = 2$  clusters per domain).

- Credit card approval data set: Again, the results of the proposed approach using the normal seed exchange show an oscillating behavior of the objective functions in both domains. Even with such an unstable optimization process, the seed-based IDS algorithm reaches significantly improved clustering results in both domains, see Figures 4.4a and 4.4b compared to the baseline splitting algorithm. With the XB index-based seed exchange, we reach the same value of the objective functions but this time, the seed-based IDS algorithm converges after 10 iterations, see Figures 4.4c and 4.4d. In contrast, using the DB index-based seed exchange, we reach convergence after only 3 iteration, with a higher value of the objective functions in both

domains, and yet still better than that of the splitting baseline algorithm. Table 4.12 shows the numerical results of all three seed exchange mechanisms: The DB index-based seed exchange shows superior results in all evaluation measures in the numerical domain. In the categorical domain, DB index-based seed exchange outperforms the other mechanisms in the internal validity measures, while yielding to the XB index-based seed exchange in the external validity measures.

Data type	Numerical		
Algorithm	Normal exchange	XB exchange	DB exchange
DB Index	$2.24 \pm 0.36$ [1.12, 2.28, 3.81]	$2.09 \pm 0.53$ [1.50, 1.74, 3.80]	<b><math>1.98 \pm 0.63</math></b> [0.01, 2.06, 3.81]
Silhouette Index	$0.52 \pm 0.07$ [0.20, 0.51, 0.66]	$0.54 \pm 0.09$ [0.29, 0.55, 0.68]	<b><math>0.56 \pm 0.14</math></b> [0.20, 0.55, 0.97]
Dunn Index	$0.0002 \pm 0.0002$ [0, 0.0001, 0.0009]	$0.0003 \pm 0.0003$ [0, 0.0002, 0.0009]	<b><math>0.0078 \pm 0.0497</math></b> [ $1.2e-5$ , $2.3e-4$ , 0.35]
Purity	$0.62 \pm 0.03$ [0.54, 0.60, 0.70]	<b><math>0.65 \pm 0.02</math></b> [0.57, 0.65, 0.70]	<b><math>0.65 \pm 0.05</math></b> [0.47, 0.66, 0.70]
Entropy	$0.95 \pm 0.02$ [0.87, 0.92, 0.99]	<b><math>0.91 \pm 0.03</math></b> [0.84, 0.92, 0.98]	<b><math>0.91 \pm 0.04</math></b> [0.84, 0.91, 0.99]
NMI	$0.07 \pm 0.01$ [0.005, 0.06, 0.12]	$0.09 \pm 0.03$ [0.008, 0.08, 0.18]	<b><math>0.10 \pm 0.04</math></b> [ $1.3e-4$ , 0.09, 0.18]
Data type	Categorical		
Algorithm	Normal exchange	XB exchange	DB exchange
DB Index	$1.69 \pm 0.26$ [1.16, 1.82, 1.95]	$1.58 \pm 0.39$ [0.97, 1.82, 2.86]	<b><math>1.41 \pm 0.31</math></b> [0.97, 1.38, 1.95]
Silhouette Index	$0.22 \pm 0.02$ [0.18, 0.23, 0.36]	$0.22 \pm 0.04$ [0.15, 0.23, 0.35]	<b><math>0.23 \pm 0.05</math></b> [0.16, 0.23, <b>0.36</b> ]
Dunn Index	<b><math>0.12 \pm 0.01</math></b> [0.11, 0.12, 0.22]	<b><math>0.12 \pm 0.02</math></b> [0.11, 0.11, 0.22]	<b><math>0.12 \pm 0.03</math></b> [0.11, 0.11, <b>0.22</b> ]
Purity	$0.71 \pm 0.02$ [0.54, 0.73, 0.78]	<b><math>0.75 \pm 0.06</math></b> [0.54, 0.78, 0.83]	$0.73 \pm 0.08$ [0.54, 0.77, 0.80]
Entropy	$0.85 \pm 0.07$ [0.64, 0.75, 0.98]	<b><math>0.77 \pm 0.07</math></b> [0.64, 0.72, 0.96]	$0.80 \pm 0.08$ [0.70, 0.78, 0.98]
NMI	$0.19 \pm 0.02$ [0.02, 0.22, 0.28]	<b><math>0.22 \pm 0.08</math></b> [0.02, 0.26, 0.36]	$0.19 \pm 0.08$ [0.01, 0.22, 0.30]

**Table 4.12:** Clustering results of seed-based IDS for the Credit card approval data set with different seed exchange mechanisms (50 runs,  $k = 2$  clusters per domain).

- MIRFlickr data set: As shown in Figures 4.5a and 4.5b, the normal seed exchange mechanism is able to achieve convergence after 16 iterations in both domains. Moreover, in the text domain, it leads to a lower value of the objective function than the objective function of the baseline splitting algorithm. The results of the XB seed exchange show lower performance compared to the baseline splitting algorithm and other seed exchange mechanisms, and also failed to converge (see Figures 4.5c and 4.5d). The results of the DB seed exchange yields to the results of the splitting algorithm in the text domain and outperforms it in the visual domain. As shown in Table 4.13, the DB seed exchange mechanism outperforms all other seed exchange methods in both domains. Note that for the MIRFlickr data set, we do not have an external class label, but rather a set of tags for each image. Thus, in addition to the regular validity indices, we also compute those same validity indices in the tag space (instead of the original data space) to capture how the clusters conform with the ground-truth tags for the data. These validity indices are referred to as *Tags DB*, *Tags Silhouette* and *Tags Dunn* in Table 4.13.

To conclude, the proposed seed-based IDS algorithm with the DB index-based seed exchange mechanism leads to better clustering results, algorithm stability and a faster convergence, than the

Data type	Text		
Algorithm	Normal exchange	XB exchange	DB exchange
DB Index	$2.46 \pm 0.06[2.39, 2.44, 2.56]$	$2.31 \pm 0.11[2.19, 2.28, 2.51]$	<b><math>2.29 \pm 0.06[2.22, 2.28, 2.40]</math></b>
Silhouette Index	<b><math>0.01 \pm 0.001[0.009, 0.01, 0.01]</math></b>	<b><math>0.01 \pm 0.003[0.008, 0.01, 0.016]</math></b>	<b><math>0.01 \pm 0.002[0.007, 0.011, 0.016]</math></b>
Dunn Index	<b><math>0.01 \pm 0.003[0.006, 0.01, 0.014]</math></b>	<b><math>0.01 \pm 0.004[0.007, 0.01, 0.02]</math></b>	<b><math>0.01 \pm 0.006[0.007, 0.01, 0.02]</math></b>
Tags DB Index	$61.40 \pm 18.78[39.94, 60.91, 98.35]$	$50.20 \pm 16.08[29.94, 50.01, 79.63]$	<b><math>48.58 \pm 14.29[26.27, 43.89, 69.95]</math></b>
Tags Silhouette Index	$0.14 \pm 0.03[0.11, 0.13, 0.20]$	$0.15 \pm 0.04[0.12, 0.17, 0.19]$	<b><math>0.16 \pm 0.03[0.11, 0.17, 0.21]</math></b>
Tags Dunn Index	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>
Data type	Visual		
Algorithm	Normal exchange	XB exchange	DB exchange
DB Index	$2.53 \pm 0.25[2.33, 2.41, 3.15]$	$2.71 \pm 0.13[2.53, 2.69, 3.00]$	<b><math>2.20 \pm 0.14[1.99, 2.20, 2.52]</math></b>
Silhouette Index	$0.08 \pm 0.01[0.05, 0.08, 0.09]$	$0.07 \pm 0.01[0.05, 0.07, 0.08]$	<b><math>0.10 \pm 0.01[0.08, 0.10, 0.11]</math></b>
Dunn Index	$0.003 \pm 0.003[0.0004, 0.003, 0.009]$	$0.002 \pm 0.002[0.0004, 0.0015, 0.005]$	<b><math>0.005 \pm 0.001[0.004, 0.005, 0.005]</math></b>
Tags DB Index	$65.42 \pm 9.47[53.42, 62.29, 80.57]$	$72.70 \pm 16.25[47.45, 68.84, 99.42]$	<b><math>69.55 \pm 13.91[53.02, 66.85, 93.60]</math></b>
Tags Silhouette Index	$0.08 \pm 0.009[0.07, 0.08, 0.10]$	$0.08 \pm 0.007[0.07, 0.08, 0.10]$	<b><math>0.09 \pm 0.01[0.07, 0.09, 0.10]</math></b>
Tags Dunn Index	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>

**Table 4.13:** Clustering results of seed-based IDS for the MIRFlickr data set with different seed exchange mechanisms (10 runs,  $k = 16$  clusters per domain).

seed-based IDS with the XB index-based and normal seed exchange mechanisms in terms of the internal and external validity measures.

#### 4.2.2 Results for the Constraint-based IDS Clustering: Studying the Impact of the Number of Constraints

The mechanism of the proposed constraint-based IDS clustering depends on the amount of information exchanged between the domains. This amount is determined by the number of constraints  $nc_T$  or number of exchange points per cluster  $n_T$ . The number of constraints generated in domain  $T_1$  and sent to domain  $T_2$  is defined as the number of possible pairs generated between a total of  $n_{T_1}k_{T_1}$  points, which is

$$nc_{T_1} = \frac{(n_{T_1}k_{T_1})(n_{T_1}k_{T_1} - 1)}{2}, \quad (4.2)$$

where  $k_{T_1}$  is the number of clusters in domain  $T_1$ . For example, if the numbers of exchange points in each domain are  $n_{T_1} = 5$ ,  $n_{T_2} = 10$  and the numbers of clusters are  $k_{T_1} = 2$ ,  $k_{T_2} = 3$  then  $nc_{T_1} = 45$  and  $nc_{T_2} = 435$  constraints, making in total 480 pairwise constraints.

Figures 4.6, 4.7, and 4.8 show the performance of the constraint-based IDS with respect to the different number of exchange points in the different domains. The heat maps (a) and (b) in each sub-figure show the percent-wise improvement (or decline) over the baseline value of NMI with no constraint exchange,

$$\Delta NMI = \frac{(NMI_{IDS} - NMI_{splitting})}{NMI_{splitting}} 100\% \quad (4.3)$$

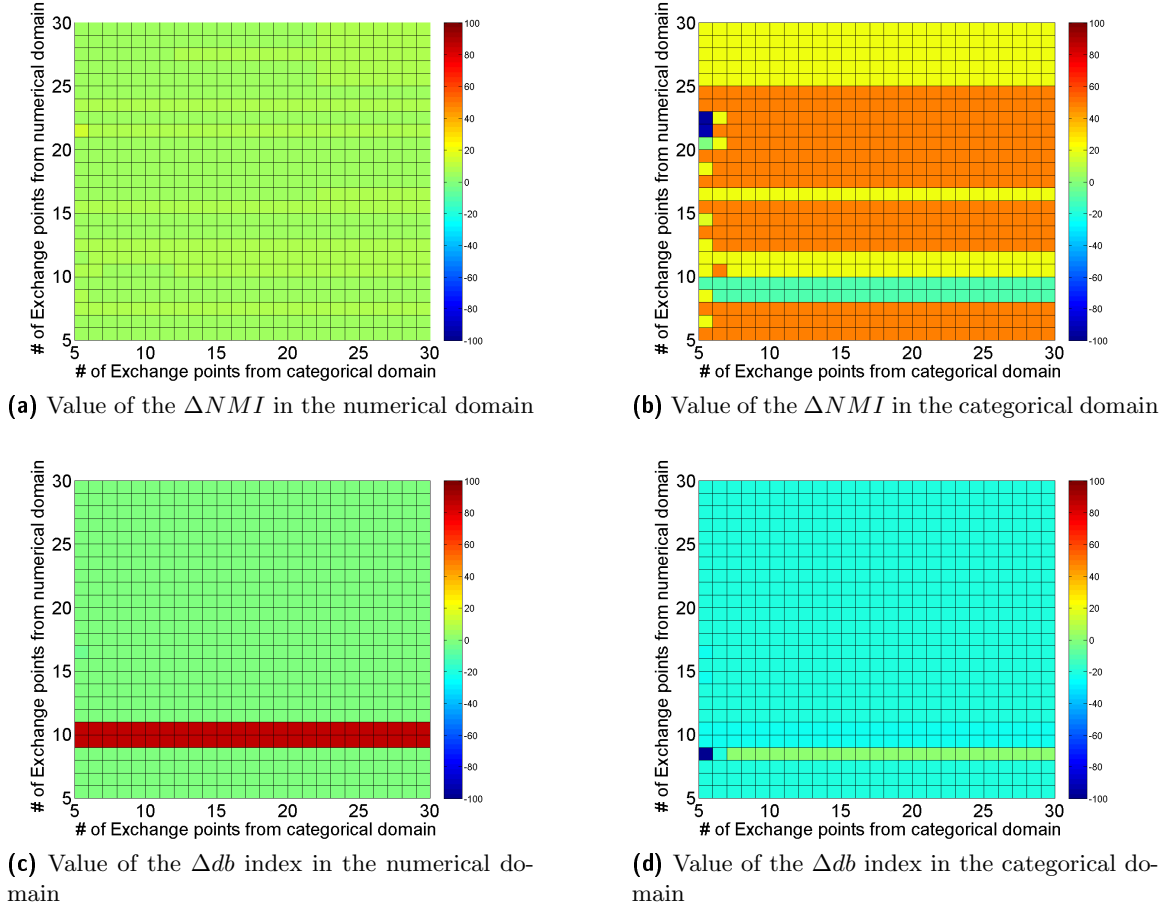
with respect to the number of exchange points from the different domains. The heat maps (c) and

(d) similarly show the improvement (or decline) but in terms of the DB index, i.e.

$$\Delta db = -\frac{(db_{IDS} - db_{splitting})}{db_{splitting}} 100\%. \quad (4.4)$$

We change the sign of the  $\Delta db$  value in (4.4) so that a lower value of the DB index reflects better clustering and all the heat maps follow the same color code. The color bar on the heat map ranges from  $-100\%$  (decay) to  $100\%$  (improvement), starting from the dark blue color (decay), continuing to the “cold” colors (neutral), then reaching the “warm” colors (improvement) and ending with dark red. Each point in the heat map is a result of an independent run of the constraint-based IDS approach, the size of each map is  $29 \times 29$  amounting in total to 841 experiments per data set (for the Adult data set the size of the heat map is  $26 \times 26$ , making in total 676 experiments), with a minimum number of exchange points per cluster,  $n_{T,min} = 2$  and maximum  $n_{T,max} = 30$ . The Adult, Heart disease, and Credit card approval data sets were clustered with  $k_{T_1} = k_{T_2} = 2$  clusters per domain, and we let the algorithm run for  $t_{T_1} = t_{T_2} = 1$  iterations in each turn of the exchange. Results for each data set are presented below:

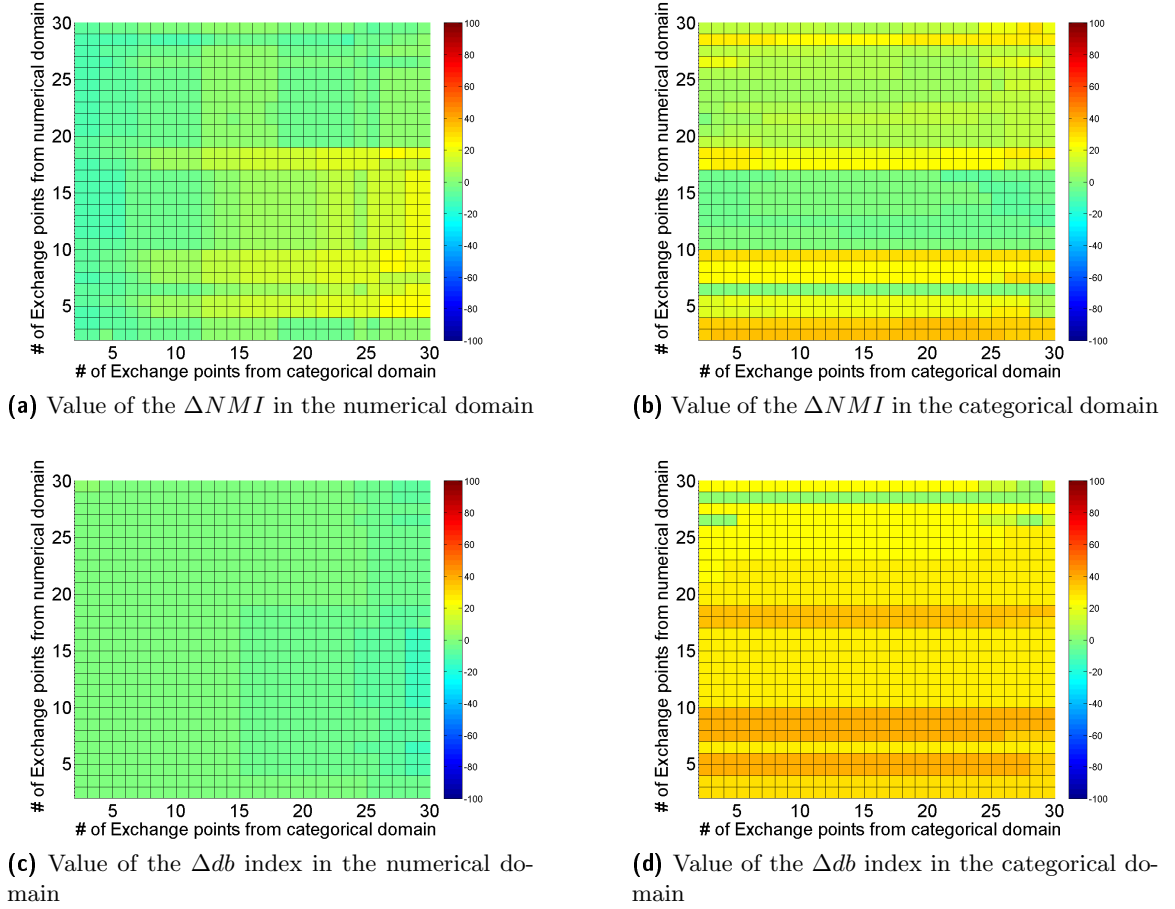
- Adult data set: Figure 4.6a shows that the value of  $\Delta NMI$  in the numerical domain is extremely stable and higher than that of  $NMI_{splitting}$  of the baseline splitting algorithm by  $10 - 20\%$ ; while in the categorical domain, it is higher by  $20 - 60\%$ , except for one strip corresponding to the number of exchange points in the numerical domain,  $n_{T_1} = 8$  or  $n_{T_1} = 9$ , and any number of exchange points,  $n_{T_2}$ , in the categorical domain, see Figure 4.6b. In Figure 4.6c we see a similar trend, but this time, almost the same strip indicates extremely high improvement over the baseline splitting algorithm in terms of the DB index. Such disagreement between an external and internal validity measure is not uncommon and indicates that the cluster structure does not match the “true” class labels. In the categorical domain, in Figure 4.6d, we see that, with the exception of the same number of exchange points  $n_{T_1} = 8 - 9$  the constraint-based IDS results are worse than the baseline splitting algorithm results. We can conclude for this data that there is an asymmetrical benefit from the inter-domain supervision, with the categorical domain offering more guidance toward a better internal cluster structure.
- Heart disease data set: In the numerical domain, Figures 4.7a and 4.7c show a smooth heat surface, indicating the algorithm’s stability, and overall (up to  $40\%$ ) improvement over the baseline splitting algorithm. Also, these figures show the overall agreement between external



**Figure 4.6:** Constraint-based IDS ( $k_{T_1} = k_{T_2} = 2$ ,  $t_{T_1} = t_{T_2} = 1$ ): effect of the number of constraints in the Adult data set. Warm colors indicate improvement and cold colors indicate decline over the baseline splitting algorithm.

and internal validity measures, indicating that the clustering structure and ground-truth class distribution are the same. In the categorical domain, Figures 4.7b and 4.7d show similar behavior with even more improvement resulting from the exchange, in some cases over 40% compared to the baseline algorithm. From the asymmetry of the maps, it seems that the categorical domain provides guidance over a wide range of exchange numbers.

- Credit card approval data set: Again, in the numerical domain, as seen in Figures 4.8a and 4.8c, the external and internal validity indices disagree and with an increase of the number of exchange points from the categorical domain, the improvement in terms of NMI increases, in contrast to an increase in the decay of the DB index. As for the categorical domain, we see an improvement in terms of NMI as long as  $n_{T_1} > 4$ ; while for the DB index, we see an improvement when  $10 \leq n_{T_1} \leq 25$  and  $n_{T_2} > 8$ .

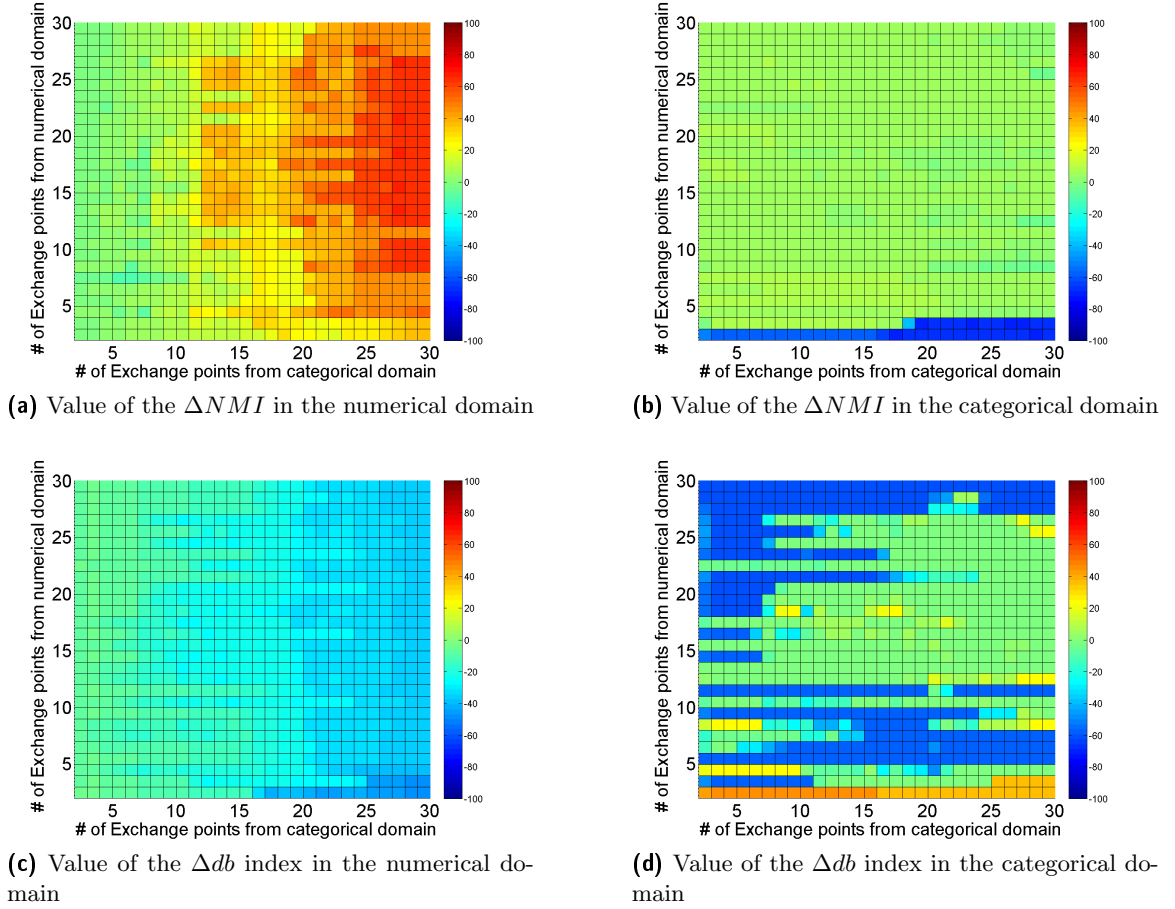


**Figure 4.7:** Constraint-based IDS ( $k_{T_1} = k_{T_2} = 2$ ,  $t_{T_1} = t_{T_2} = 1$ ): effect of the number of constraints in the Heart disease data set. Warm colors indicate improvement and cold colors indicate decline over the baseline splitting algorithm.

We do not present such exhaustive experiments for the MIRFlickr data set due to the high computation cost as a result of the high number of points and clusters in each domain ( $k_T = 16$ ). After several trials, we found that we achieve the best results with  $n_{T_1} = 5$  and  $n_{T_2} = 5$ , making in total 3,160 pairwise constraints from  $5 \times 16 = 80$  points per domain.

To conclude, in Figures 4.6-4.8, we observe that the “warm” colors are dominant, meaning that the proposed constraint-based IDS clustering generally results in an improvement over the splitting algorithm in the following aspects:

- over a wide range of the algorithm parameters,
- for different data sets with different sizes and number of features,
- in different validation measures,



**Figure 4.8:** Constraint-based IDS ( $k_{T_1} = k_{T_2} = 2$ ,  $t_{T_1} = t_{T_2} = 1$ ): effect of the number of constraints in the Credit card approval data set. Warm colors indicate improvement and cold colors indicate decline over the baseline splitting algorithm.

- for some data sets, the improvement is asymmetric, with one domain contributing more to guide the other,
- for some data sets, validations in terms of an external (NMI) and internal (DB) validity measures give opposite results. This reflects some disagreement between the internal structure and external labels. Of course, the external validity option is generally impossible without external “true” class labels, which is the case with most real-life data.

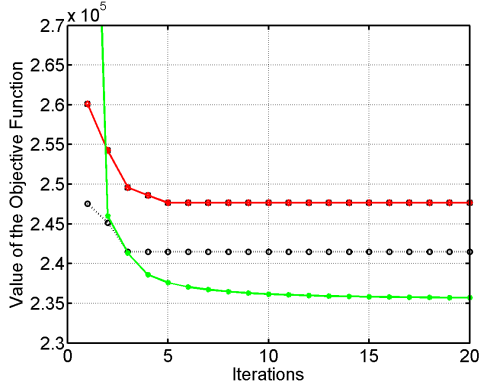
### 4.2.3 Comparison of the Proposed IDS Framework with Other Clustering Methods

We compare the proposed seed-based and constraint-based Inter-Domain Supervised clustering approaches with the following techniques:

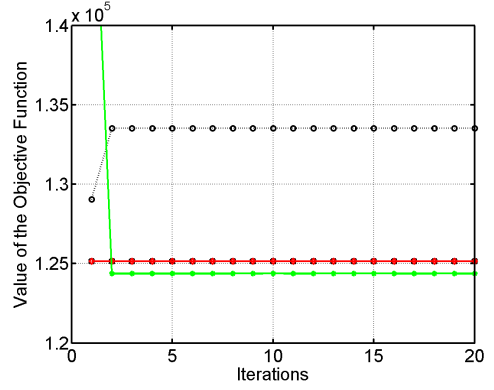
1. Splitting algorithm. A classical baseline approach, where we split the data according to its domain and run a specialized clustering algorithm on each domain separately. This is equivalent to traditional clustering with no exchange (see Section 2.3.1).
2. Conversion algorithm: Another traditional algorithm where we convert all data to the same attribute type and cluster it using a specialized clustering algorithm (see Section 2.3.1). Note that for the MIRFlickr data set, since both domains have the same bag of features or words (BOF or BOW) format, there is no need for converting one domain to another, instead we normalized each domain to an  $L_2$ -norm of 1, merged the data records together and normalized them again to an  $L_2$ -norm of 1. Despite a similar BOW format, the two domains arise from conceptually different sources (visual versus text, and can therefore have different structure).
3. Ensemble clustering with voting methods as a consensus function using as base algorithm k-means, k-modes, and spherical k-means for numerical, categorical, and BOW domains, respectively (see Section 3.3.1).
4. Ensemble clustering with post clustering of the cluster membership matrix (see Section 3.3.1).
5. Multiview clustering algorithm, where two independent hypotheses are trained on different domains with bootstrapping by providing each other with cluster labels for the unlabeled domain (see Section 3.3.2).

Figures 4.9 and 4.10 show how the value of the objective functions of the the seed-based IDS, constraint-based IDS, splitting, and multiview k-means algorithms behave during the clustering process. We do not show the values of the objective functions of the conversion and ensemble methods, since in these algorithm, there is no interaction between the domains. These figures were constructed based on a typical run of each algorithm. Tables 4.14, 4.15, 4.16, and 4.17 show average results (in the format of  $\text{mean} \pm \text{std} [\text{min}, \text{median}, \text{max}]$  with the best results in a bold font) of repeated experiments for each data set and algorithm. We repeated each experiment 50 times for the Heart disease and Credit card approval data sets and 10 times for the larger Adult and MIRFlickr data sets.

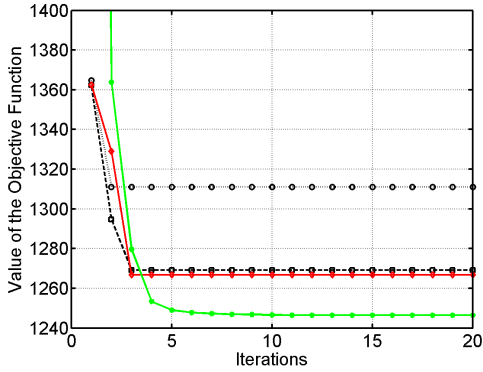
- Adult data set: Figures 4.9a and 4.9b show the value of the objective function for the compared algorithms in the numerical and categorical domains, respectively. In the numerical domain, the constrained-based IDS outperforms the seed-based IDS, splitting, and multiview clustering algorithms obtaining a lower value of the objective function. We run the seed-based IDS with



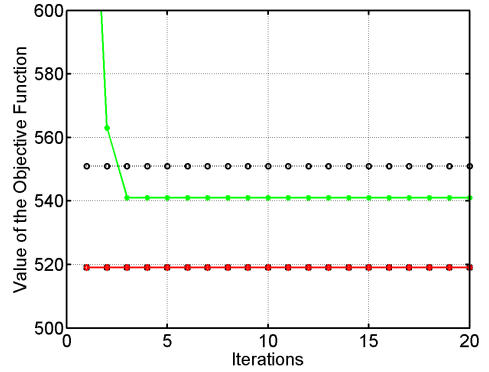
(a) Adult data set: numerical domain.



(b) Adult data set: categorical domain.

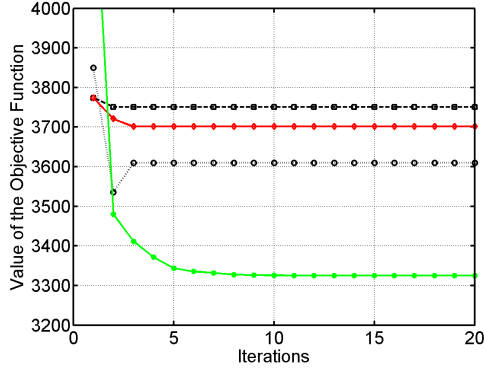


(c) Heart disease data set: numerical domain.

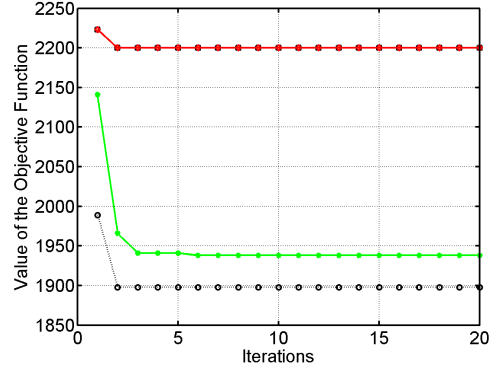


(d) Heart disease data set: categorical domain.

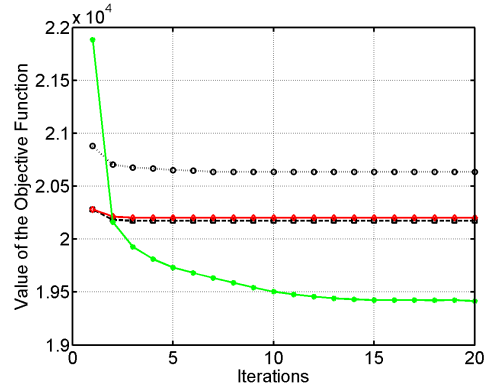
**Figure 4.9:** Value of the objective functions for seed-based IDS (red diamonds), constraint-based IDS (green stars), splitting clustering (dashed black squares), and multiview clustering (dotted black circles). See the value of the validity indices in Table 4.14 for the Adult data set and Table 4.15 for the Heart disease data set. The number of clusters is set to  $k = 2$  for both data sets.



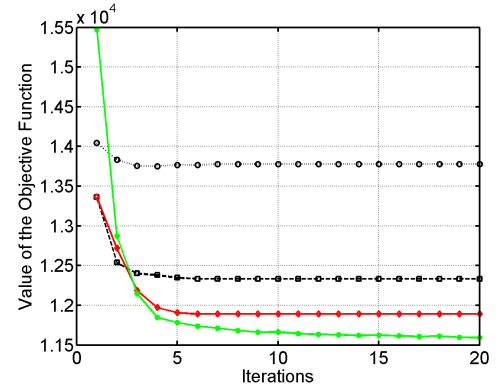
(a) Credit card approval data set: numerical domain.



(b) Credit card approval data set: categorical domain.



(c) MIRFlickr data set: text domain.



(d) MIRFlickr data set: visual domain.

**Figure 4.10:** Value of the objective functions for seed-based IDS (red diamonds), constraint based IDS (green stars), splitting clustering (dashed black squares), and multiview clustering (dotted black circles). See the value of the validity indices in Table 4.16 for the Credit card approval data set and Table 4.17 for the MIRFlickr data set. The number of cluster is  $k = 2$  for the Credit card approval data set, and  $k = 16$  for the MIRFlickr data set.

the DB index-based seed exchange mechanism and the following parameters:  $k = 2$  clusters with 2 seeds per cluster, while for the constraint-based IDS:  $n_{T_1} = n_{T_2} = 5$ ,  $t_{T_1} = t_{T_2} = 1$ . The seed-based IDS approach shows exactly the same results as the baseline splitting algorithm in both domains and yields to the multiview and constraint-based IDS clustering in the numerical domain. In the categorical domain, the constraint-based IDS shows better results than all other algorithms. Table 4.14a shows the results of the seed-based IDS, constraint-based IDS, splitting, and conversion methods with the best results shown in a bold font. The seed-based IDS framework outperforms all other techniques in terms of internal validity indices in the numerical domain, while the constraint-based IDS shows better results in term of external validity measures in the categorical domain. Note the extremely low minimum value of the DB and high value of the Silhouette indices of the seed-based IDS in both domains indicating the superior potential capabilities of the IDS approach. Table 4.14b shows the results of the proposed IDS approaches, ensemble techniques, and multiview k-mean clustering algorithm. Again, the Seed-based IDS outperforms the other methods in both domains in terms of internal validity indices, yielding to the ensemble clustering method only in terms of external indices.

- Heart disease data set: As Figure 4.9c illustrates, the constraint-based IDS outperforms the other methods in the numerical domain, while the seed-based IDS obtains similar results to the splitting algorithm. In the categorical domain, the seed-based IDS also shows similar results to the splitting algorithm but outperforms the constraint-based IDS and multiview clustering, see Figure 4.9d. Table 4.15a shows the results of the proposed IDS approaches and traditional clustering techniques. Here, the traditional methods outperform the proposed approaches in the numerical domain in all the validity measures. In the categorical domain, we see a completely opposite picture, where the constraint-based IDS obtains significantly better clustering results than all other techniques. Table 4.15b shows that the constraint-based IDS outperforms the ensemble and multiview clustering in the internal validity measures, but yields to ensemble clustering in the external indices. In the categorical domain, we observe a similar behavior again. The proposed IDS approaches outperform in the internal validity indices and concede to ensemble clustering in terms of the external indices. We ran the seed-based IDS with the following parameters:  $k = 2$  clusters and number of seeds equal 2, and for the constraint-based IDS:  $n_{T_1} = 5$ ,  $n_{T_2} = 11$ , and  $t_{T_1} = t_{T_2} = 1$ .
- Credit card approval data set: Figure 4.10a shows that the value of the objective function of the constraint-based IDS is much lower than the objective function of the other methods,

Numerical					
Data type	Seed-based IDS	Constraint-based IDS	Splitting	Conversion	
Algorithm					
DB Index	<b>3.09</b> $\pm$ 1.09[0.42, 3.44, 3.77]	3.30 $\pm$ 0.04[3.29, 3.29, 3.43]	3.29 $\pm$ 0.001[3.29, 3.29, 3.29]	11.53 $\pm$ 7.70[1.48, 12.31, 26.72]	
Silhouette Index	<b>0.29</b> $\pm$ 0.18[0.18, 0.21, <b>0.71</b> ]	0.21 $\pm$ 0.01[0.20, 0.20, 0.22]	0.21 $\pm$ 0.0[0.21, 0.21, 0.21]	0.07 $\pm$ 0.05[-0.02, 0.08, 0.17]	
Dunn Index	<b>0.001</b> $\pm$ 0[0.001, 0.001, 0.001]	<b>0.001</b> $\pm$ 0[0.001, 0.001, 0.001]	0 $\pm$ 0.00[0.00, 0.00, 0.00]	0 $\pm$ 0.00[0.00, 0.00, 0.00]	
Purity	0.62 $\pm$ 0.07[0.52, 0.60, 0.75]	<b>0.65</b> $\pm$ 0.01[0.64, 0.64, 0.67]	0.64 $\pm$ 0.001[0.64, 0.64, 0.64]	0.62 $\pm$ 0.11[0.25, 0.71, 0.75]	
Entropy	0.77 $\pm$ 0.03[0.72, 0.78, 0.79]	<b>0.71</b> $\pm$ 0.002[0.70, 0.71, 0.71]	<b>0.71</b> $\pm$ 0[0.71, 0.71, 0.71]	0.73 $\pm$ 0.06[0.69, 0.69, 0.81]	
NMI	0.06 $\pm$ 0.03[0.02, 0.05, 0.10]	<b>0.11</b> $\pm$ 0.001[0.11, 0.11, 0.11]	0.10 $\pm$ 0[0.10, 0.10, 0.10]	0.08 $\pm$ 0.07[2.1e - 4, 0.13, 0.13]	
Categorical					
Data type	Seed-based IDS	Constraint-based IDS	Splitting	Conversion	
Algorithm					
DB Index	1.22 $\pm$ 0.14[ <b>1.10</b> , 1.12, 1.40]	1.51 $\pm$ 0.55[1.12, 1.37, 3.07]	<b>1.15</b> $\pm$ 0.09[1.11, 1.12, 1.37]	1.50 $\pm$ 0.23[1.22, 1.43, 1.92]	
Silhouette Index	<b>0.25</b> $\pm$ 0.01[0.23, 0.24, 0.27]	<b>0.25</b> $\pm$ 0.06[0.07, 0.27, 0.27]	<b>0.25</b> $\pm$ 0.01[0.24, 0.24, 0.27]	0.22 $\pm$ 0.02[0.19, 0.21, 0.25]	
Dunn Index	<b>0.125</b> $\pm$ 0[0.125, 0.125, 0.125]	0.001 $\pm$ 0[0.001, 0.001, 0.001]	<b>0.125</b> $\pm$ 0[0.125, 0.125, 0.125]	0.00 $\pm$ 0.00[0.00, 0.00, 0.00]	
Purity	0.59 $\pm$ 0.06[0.50, 0.56, 0.67]	<b>0.65</b> $\pm$ 0.09[0.39, 0.69, 0.69]	0.59 $\pm$ 0.05[0.55, 0.55, 0.67]	0.56 $\pm$ 0.04[0.53, 0.55, 0.65]	
Entropy	0.73 $\pm$ 0.02[0.71, 0.73, 0.78]	<b>0.71</b> $\pm$ 0.04[0.69, 0.69, 0.80]	0.73 $\pm$ 0.01[0.71, 0.73, 0.73]	0.73 $\pm$ 0.02[0.70, 0.74, 0.75]	
NMI	0.09 $\pm$ 0.001[0.08, 0.09, 0.11]	<b>0.11</b> $\pm$ 0.04[0.004, 0.13, 0.13]	0.09 $\pm$ 0.01[0.08, 0.08, 0.11]	0.06 $\pm$ 0.03[0.07, 0.08, 0.12]	

(a) Comparison of the proposed IDS approaches with the traditional splitting and conversion algorithms.

Numerical					
Data type	Seed-based IDS	Constraint-based IDS	Ensemble: Voting	Ensemble: Clustering	Multiview K-means
Algorithm					
DB Index	<b>3.09</b> $\pm$ 1.09[0.42, 3.44, 3.77]	3.30 $\pm$ 0.04[3.29, 3.29, 3.43]	3.42 $\pm$ 0.41[3.29, 3.29, 4.51]	9.50 $\pm$ 4.06[4.58, 10.54, 14.46]	3.59 $\pm$ 0.26[3.44, 3.44, 4.08]
Silhouette Index	<b>0.29</b> $\pm$ 0.18[0.18, 0.21, <b>0.71</b> ]	0.21 $\pm$ 0.01[0.20, 0.20, 0.22]	0.19 $\pm$ 0.03[0.12, 0.20, 0.21]	0.17 $\pm$ 0.09[0.06, 0.25, 0.25]	0.15 $\pm$ 0.05[0.13, 0.14, 0.31]
Dunn Index	<b>0.001</b> $\pm$ 0[0.001, 0.001, 0.001]	0.001 $\pm$ 0[0.001, 0.001, 0.001]	0.001 $\pm$ 0[0.001, 0.001, 0.001]	0.001 $\pm$ 0[0.001, 0.001, 0.001]	0.001 $\pm$ 0[0.001, 0.001, 0.001]
Purity	0.62 $\pm$ 0.07[0.52, 0.60, 0.75]	0.65 $\pm$ 0.01[0.64, 0.64, 0.67]	0.64 $\pm$ 0.005[0.63, 0.64, 0.64]	<b>0.83</b> $\pm$ 0.19[0.54, 1, 1]	0.62 $\pm$ 0.04[0.60, 0.61, 0.73]
Entropy	0.77 $\pm$ 0.03[0.72, 0.78, 0.79]	0.71 $\pm$ 0.002[0.70, 0.71, 0.71]	0.71 $\pm$ 0.008[0.68, 0.71, 0.71]	<b>0.31</b> $\pm$ 0.36[0, 0, 0.70]	0.67 $\pm$ 0.01[0.66, 0.67, 0.73]
NMI	0.06 $\pm$ 0.03[0.02, 0.05, 0.10]	0.11 $\pm$ 0.001[0.11, 0.11, 0.11]	0.11 $\pm$ 0.01[0.10, 0.10, 0.13]	<b>0.61</b> $\pm$ 0.45[0.11, 1, 1]	[0.14 $\pm$ 0.01[0.09, 0.15, 0.16]
Categorical					
Data type	Seed-based IDS	Constraint-based IDS	Ensemble: Voting	Ensemble: Clustering	Multiview K-means
Algorithm					
DB Index	<b>1.22</b> $\pm$ 0.14[ <b>1.10</b> , 1.12, 1.40]	1.51 $\pm$ 0.55[1.12, 1.37, 3.07]	3.15 $\pm$ 0.58[1.58, 3.34, 3.34]	2.28 $\pm$ 0.58[1.54, 2.37, 3.12]	2.09 $\pm$ 0.43[1.54, 2.09, 3.13]
Silhouette Index	<b>0.25</b> $\pm$ 0.01[0.23, 0.24, 0.27]	0.25 $\pm$ 0.06[0.07, 0.27, 0.27]	0.06 $\pm$ 0.02[0.05, 0.05, 0.11]	0.12 $\pm$ 0.01[0.04, 0.04, 0.25]	0.13 $\pm$ 0.02[0.08, 0.13, 0.16]
Dunn Index	<b>0.125</b> $\pm$ 0[0.125, 0.125, 0.125]	0.001 $\pm$ 0[0.001, 0.001, 0.001]	0.001 $\pm$ 0[0.001, 0.001, 0.001]	0.001 $\pm$ 0[0.001, 0.001, 0.001]	0.001 $\pm$ 0[0.001, 0.001, 0.001]
Purity	0.59 $\pm$ 0.06[0.50, 0.56, 0.67]	0.65 $\pm$ 0.09[0.39, 0.69, 0.69]	0.64 $\pm$ 0.005[0.63, 0.64, 0.64]	<b>0.83</b> $\pm$ 0.19[0.54, 1, 1]	0.62 $\pm$ 0.04[0.60, 0.61, 0.73]
Entropy	0.73 $\pm$ 0.02[0.71, 0.73, 0.78]	0.71 $\pm$ 0.04[0.69, 0.69, 0.80]	0.71 $\pm$ 0.008[0.68, 0.71, 0.71]	<b>0.31</b> $\pm$ 0.36[0, 0, 0.70]	0.67 $\pm$ 0.01[0.66, 0.67, 0.73]
NMI	0.09 $\pm$ 0.001[0.08, 0.09, 0.11]	0.11 $\pm$ 0.04[0.004, 0.13, 0.13]	0.11 $\pm$ 0.01[0.10, 0.10, 0.13]	<b>0.61</b> $\pm$ 0.45[0.11, 1, 1]	[0.14 $\pm$ 0.01[0.09, 0.15, 0.16]

(b) Comparison of the proposed IDS approaches to the ensemble and multiview clustering algorithms.

**Table 4.14:** Clustering results for the Adult data set (10 runs,  $k = 2$  clusters per domain). We run the Seed-based IDS with the following parameters:  $k = 2$  number of seeds, and the constraint-based IDS:  $n_{T_1} = n_{T_2} = 5$ ,  $tr_1 = tr_2 = 1$ .

Numerical				
Data type	Seed-based IDS	Constraint-based IDS	Splitting	Conversion
Algorithm				
DB Index	$1.73 \pm 0.15$ [1.54, 1.71, 2.14]	$1.69 \pm 0$ [1.69, 1.69, 1.69]	<b><math>1.65 \pm 0.003</math></b> [1.65, 1.65, 1.65]	$2.97 \pm 0.56$ [0.21, 2.95, 5.16]
Silhouette Index	$0.33 \pm 0.04$ [0.26, 0.33, 0.41]	$0.35 \pm 0$ [0.35, 0.35, 0.35]	<b><math>0.36 \pm 0.005</math></b> [0.36, 0.36, 0.36]	$0.26 \pm 0.07$ [0.16, 0.25, 0.75]
Dunn Index	$3.3e - 3 \pm 2.2e - 3$ [1.2e - 5, 2.3e - 4, 0.35]	$0.01 \pm 0$ [0.01, 0.01, 0.01]	<b><math>4.6e - 3</math></b> [4.6e - 3, 4.6e - 3, 4.6e - 3]	$0.04 \pm 0.14$ [0.015, 0.015, 0.98]
Purity	$0.72 \pm 0.03$ [0.65, 0.72, 0.76]	$0.76 \pm 0$ [0.76, 0.76, 0.76]	$0.75 \pm 0.003$ [0.75, 0.75, 0.75]	<b><math>0.77 \pm 0.11</math></b> [0.47, 0.82, 0.82]
Entropy	$0.84 \pm 0.03$ [0.79, 0.84, 0.91]	$0.79 \pm 0$ [0.79, 0.79, 0.79]	$0.80 \pm 0.003$ [0.80, 0.80, 0.81]	<b><math>0.72 \pm 0.11</math></b> [0.67, 0.67, 0.99]
NMI	$0.15 \pm 0.03$ [0.08, 0.16, 0.20]	$0.20 \pm 0$ [0.20, 0.20, 0.20]	$0.19 \pm 0.004$ [0.18, 0.19, 0.19]	<b><math>0.28 \pm 0.11</math></b> [2.1e - 4, 0.32, 0.32]
Categorical				
Data type	Seed-based IDS	Constraint-based IDS	Splitting	Conversion
Algorithm				
DB Index	$0.76 \pm 0.01$ [0.75, 0.76, 0.76]	<b><math>0.65 \pm 0</math></b> [0.65, 0.65, 0.65]	$1.08 \pm 1.55$ [0.53, 0.56, 10.05]	$1.52 \pm 0.09$ [0.66, 1.24, 5.06]
Silhouette Index	$0.30 \pm 0.01$ [0.29, 0.30, 0.31]	$0.29 \pm 0$ [0.29, 0.29, 0.29]	<b><math>0.32 \pm 0.09</math></b> [0.04, 0.36, 0.44]	$0.15 \pm 0.05$ [0.02, 0.17, 0.24]
Dunn Index	$0.14 \pm 0$ [0.14, 0.14, 0.14]	<b><math>0.14 \pm 0</math></b> [0.14, 0.14, 0.14]	$0.02 \pm 0.01$ [0.02, 0.02, 0.02]	$0.01 \pm 0.001$ [0.01, 0.01, 0.01]
Purity	$0.78 \pm 0.03$ [0.71, 0.77, 0.81]	<b><math>0.78 \pm 0</math></b> [0.78, 0.78, 0.78]	$0.75 \pm 0.08$ [0.78, 0.81, 0.81]	$0.73 \pm 0.08$ [0.50, 0.77, 0.81]
Entropy	$0.74 \pm 0.04$ [0.70, 0.75, 0.87]	<b><math>0.72 \pm 0</math></b> [0.72, 0.72, 0.72]	$0.77 \pm 0.08$ [0.69, 0.75, 0.99]	$0.79 \pm 0.08$ [0.68, 0.77, 0.98]
NMI	$0.25 \pm 0.04$ [0.13, 0.24, 0.30]	<b><math>0.27 \pm 0</math></b> [0.27, 0.27, 0.27]	$0.22 \pm 0.09$ [0.23, 0.30, 0.30]	$0.19 \pm 0.08$ [0.01, 0.23, 0.31]

(a) Comparison of the proposed IDS approaches with the traditional splitting and conversion algorithms.

Numerical					
Data type	Seed-based IDS	Constraint-based IDS	Ensemble: Voting	Ensemble: Clustering	Multiview K-means
Algorithm					
DB Index	1.73 ± 0.15[1.54, 1.71, 2.14]	<b>1.69 ± 0</b> [1.69, 1.69, 1.69]	1.89 ± 0.54[1.64, 1.65, 4.07]	4.35 ± 1.27[2.00, 4.33, 7.31]	1.82 ± 0.05[1.78, 1.78, 1.91]
Silhouette Index	0.33 ± 0.04[0.26, 0.33, 0.41]	<b>0.35 ± 0</b> [0.35, 0.35, 0.35]	0.34 ± 0.05[0.15, 0.36, 0.37]	0.21 ± 0.05[0.14, 0.18, 0.35]	0.33 ± 0.001[0.33, 0.33, 0.34]
Dunn Index	3.3e − 3 ± 2.2e − 3[1.2e − 5, 2.3e − 4, 0.35]	<b>0.01 ± 0</b> [0.01, 0.01, 0.01]	0.004 ± 0.001[0.001, 0.001, 0.004]	0.002 ± 0.001[0.001, 0.002, 0.004]	0.004 ± 0.004[0.004, 0.004, 0.004]
Purity	0.72 ± 0.03[0.65, 0.72, 0.76]	0.76 ± 0[0.76, 0.76, 0.76]	0.75 ± 0.02[0.74, 0.74, 0.82]	<b>0.86 ± 0.10</b> [0.71, 0.81, 1]	0.78 ± 0.01[0.77, 0.77, 0.79]
Entropy	0.84 ± 0.03[0.79, 0.84, 0.91]	0.79 ± 0[0.79, 0.79, 0.79]	0.79 ± 0.04[0.67, 0.81, 0.83]	<b>0.47 ± 0.35</b> [0.68, 0.84]	0.74 ± 0.01[0.72, 0.76, 0.76]
NMI	0.15 ± 0.03[0.08, 0.16, 0.20]	0.20 ± 0[0.20, 0.20, 0.20]	0.19 ± 0.04[0.17, 0.18, 0.32]	<b>0.51 ± 0.35</b> [0.15, 0.30, 1]	0.24 ± 0.02[0.23, 0.23, 0.27]
Categorical					
Data type	Seed-based IDS	Constraint-based IDS	Ensemble: Voting	Ensemble: Clustering	Multiview K-means
Algorithm					
DB Index	0.76 ± 0.01[0.75, 0.76, 0.76]	<b>0.65 ± 0</b> [0.65, 0.65, 0.65]	1.38 ± 0.34[0.79, 1.66, 1.67]	1.46 ± 0.40[0.88, 1.66, 2.52]	0.89 ± 0.02[0.85, 0.90, 0.90]
Silhouette Index	<b>0.30 ± 0.01</b> [0.29, 0.30, 0.31]	0.29 ± 0[0.29, 0.29, 0.29]	0.12 ± 0.06[0.08, 0.09, 0.27]	0.22 ± 0.04[0.10, 0.20, 0.30]	0.18 ± 0.01[0.17, 0.17, 0.20]
Dunn Index	<b>0.14 ± 0</b> [0.14, 0.14, 0.14]	0.14 ± 0[0.14, 0.14, 0.14]	0.001 ± 0[0.001, 0.001, 0.001]	0.005 ± 0.03[0, 0, 0.14]	0.001 ± 0.001[0.001, 0.001, 0.001]
Purity	0.78 ± 0.03[0.71, 0.77, 0.81]	0.78 ± 0[0.78, 0.78, 0.78]	0.75 ± 0.02[0.74, 0.74, 0.82]	<b>0.86 ± 0.10</b> [0.71, 0.81, 1]	0.78 ± 0.01[0.77, 0.77, 0.79]
Entropy	0.74 ± 0.04[0.70, 0.75, 0.87]	0.72 ± 0[0.72, 0.72, 0.72]	0.79 ± 0.04[0.67, 0.81, 0.83]	<b>0.47 ± 0.35</b> [0.68, 0.84]	0.74 ± 0.01[0.72, 0.76, 0.76]
NMI	0.25 ± 0.04[0.13, 0.24, 0.30]	0.27 ± 0[0.27, 0.27, 0.27]	0.19 ± 0.04[0.17, 0.18, 0.32]	<b>0.51 ± 0.35</b> [0.15, 0.30, 1]	0.24 ± 0.02[0.23, 0.23, 0.27]

(b) Comparison of the proposed IDS approaches to the ensemble and multiview clustering algorithms.

**Table 4.15:** Clustering results for the Heart disease data set (50 runs,  $k = 2$  clusters per domain). We run the seed-based IDS with the following parameters:  $k = 2$  clusters and number of seeds, and the constraint-based IDS:  $n_{T_1} = 5$ ,  $n_{T_2} = 11$ , and  $t_{T_1} = t_{T_2} = 1$ .

therefore it achieves better clustering. The seed-based IDS yields to the multiview clustering but outperforms the traditional splitting algorithm. In the categorical domain, the constraint-based IDS yields to the multiview k-means but outperforms the seed-based IDS approach, which shows similar results to the splitting algorithm, see Figure 4.9b. Table 4.16a illustrates that the constraint-based IDS outperforms traditional approaches in terms of DB, Silhouette and purity in the numerical domain. Also note the low minimum value of the DB and high maximum value of the Silhouette index in the numerical domain for the seed-based IDS approach, showing that this approach can win by a large margin, trying to reach these best results in an unsupervised way. The proposed IDS approaches yield to the conversion algorithm in term of the Dunn index, entropy, and NMI. On the other hand, in the categorical domain, the IDS approach outperforms the traditional splitting and conversion algorithms. Table 4.16b shows that the constraint-based IDS approach outperforms all other techniques in terms of all internal validity indices in the numerical domain but concedes to the ensemble clustering algorithm in terms of all external indices. One possible reason is that the cluster structure does not match the “true” class labels or ground truth, which is common in unsupervised learning. We ran the seed-based IDS with the following parameters:  $k = 2$  clusters and number of seeds, and for the constraint-based IDS:  $n_{T_1} = 5$ ,  $n_{T_2} = 11$ , and  $t_{T_1} = t_{T_2} = 1$ .

- MIRFlickr data set: Figure 4.10c shows that the value of the objective function of the proposed constraint-based IDS approach outperforms other methods in the text and visual domains. The seed-based IDS approach yields to the constraint-based IDS in both domains but outperforms the traditional splitting algorithm and multiview clustering. Table 4.17a illustrates that the seed-based IDS approach yields to the splitting algorithm in all internal and tags DB indices but outperforms all other methods in terms of tags Silhouette and Dunn indices in the text domain. In the visual domain, we observe a similar behavior except that the constraint-based IDS performs better in terms of the Silhouette index. Table 4.17b shows that overall, the proposed IDS approaches outperform ensemble techniques and multiview clustering.

Note that the objective function of the constraint-based IDS (see Formula 3.1) is different from the standard k-means-like (sum of squared distances) objective function used in all other algorithms. The difference is in the two additional positive penalty terms, responsible for the must-link and cannot-link constraints. These terms are penalties for the unsatisfied must-link and cannot-link constraints. At the beginning of the optimization process, most of the constraints are naturally not met and the value of the objective function is still high. Then, closer to the convergence point, most

Data type	Numerical			
	Seed-based IDS	Constraint-based IDS	Splitting	Conversion
Algorithm				
DB Index	$1.98 \pm 0.63$ [0.01, 2.06, 3.81]	<b>1.84</b> $\pm$ 0.06[1.82, 1.83, 2.87]	$1.89 \pm 0.35$ [0.18, 1.97, 1.97]	$4.94 \pm 2.44$ [0.10, 4.87, 8.57]
Silhouette Index	$0.56 \pm 0.14$ [0.20, 0.55, <b>0.97</b> ]	<b>0.73</b> $\pm$ 0.01[0.72, 0.72, 0.81]	$0.63 \pm 0.06$ [0.62, 0.62, 0.95]	$0.35 \pm 0.27$ [0.12, 0.29, 0.92]
Dunn Index	$0.008 \pm 0.05$ [1.2e - 5, 2.3e - 4, 0.35]	$0.01 \pm 0.01$ [0.01, 0.01, 0.01]	$0.003 \pm 0.012$ [1.1e - 4, 1.1e - 4, 0.06]	<b>0.06</b> $\pm$ <b>0.15</b> [1.1e - 3, 0.011, 0.77]
Purity	<b>0.65</b> $\pm$ <b>0.05</b> [0.47, 0.66, 0.70]	<b>0.65</b> $\pm$ <b>0.01</b> [0.62, 0.65, 0.65]	$0.64 \pm 0.02$ [0.56, 0.64, 0.64]	<b>0.65</b> $\pm$ <b>0.12</b> [0.48, 0.56, 0.81]
Entropy	$0.91 \pm 0.04$ [0.84, 0.91, 0.99]	$0.92 \pm 0.01$ [0.92, 0.92, 0.94]	$0.93 \pm 0.01$ [0.93, 0.93, 0.98]	<b>0.86</b> $\pm$ <b>0.13</b> [0.68, 0.97, 0.99]
NMI	$0.10 \pm 0.04$ [1.3e - 4, 0.09, 0.18]	$0.08 \pm 0.01$ [0.05, 0.08, 0.08]	$0.08 \pm 0.01$ [0.03, 0.08, 0.08]	<b>0.13</b> $\pm$ <b>0.13</b> [1.2e - 4, 0.03, 0.31]
Categorical				
Algorithm				
DB Index	<b>1.41</b> $\pm$ <b>0.31</b> [0.97, 1.38, 1.95]	$1.78 \pm 0.21$ [1.49, 1.93, 1.93]	$1.81 \pm 0.25$ [1.37, 1.83, 2.87]	$7.49 \pm 8.11$ [0.83, 5.50, 39.10]
Silhouette Index	$0.23 \pm 0.05$ [0.16, 0.23, 0.36]	<b>0.24</b> $\pm$ <b>0.01</b> [0.24, 0.24, 0.31]	$0.23 \pm 0.01$ [0.19, 0.23, 0.24]	$0.06 \pm 0.03$ [-0.01, 0.06, 0.15]
Dunn Index	<b>0.12</b> $\pm$ <b>0.03</b> [0.11, 0.11, 0.22]	$0.01 \pm 0.001$ [0.01, 0.01, 0.01]	<b>0.12</b> $\pm$ <b>0.01</b> [0.11, 0.12, 0.13]	$0.01 \pm 0.001$ [0.01, 0.01, 0.01]
Purity	$0.73 \pm 0.08$ [0.54, 0.77, 0.80]	$0.76 \pm 0.03$ [0.58, 0.76, 0.76]	<b>0.79</b> $\pm$ <b>0.01</b> [0.76, 0.79, 0.82]	$0.77 \pm 0.06$ [0.56, 0.76, 0.80]
Entropy	$0.80 \pm 0.08$ [0.70, 0.78, 0.98]	<b>0.72</b> $\pm$ <b>0.03</b> [0.71, 0.71, 0.92]	$0.73 \pm 0.02$ [0.65, 0.73, 0.78]	$0.79 \pm 0.06$ [0.70, 0.76, 0.98]
NMI	$0.19 \pm 0.08$ [0.01, 0.22, 0.30]	<b>0.28</b> $\pm$ <b>0.03</b> [0.08, 0.28, 0.28]	$0.26 \pm 0.02$ [0.22, 0.27, 0.36]	$0.20 \pm 0.06$ [0.01, 0.22, 0.29]

(a) Comparison of the proposed IDS approaches with the traditional splitting and conversion algorithms.

Data type	Numerical			
	Seed-based IDS	Constraint-based IDS	Ensemble: Voting	Ensemble: Clustering
Algorithm				
DB Index	$1.98 \pm 0.63$ [0.01, 2.06, 3.81]	<b>1.84</b> $\pm$ <b>0.06</b> [1.82, 1.83, 2.87]	$2.08 \pm 0.82$ [1.62, 1.96, 9.16]	$16.41 \pm 21.15$ [1.76, 9.36, 125.52]
Silhouette Index	$0.56 \pm 0.14$ [0.20, 0.55, <b>0.97</b> ]	<b>0.73</b> $\pm$ <b>0.01</b> [0.72, 0.72, 0.81]	$0.59 \pm 0.05$ [0.34, 0.62, 0.62]	$0.16 \pm 0.15$ [-0.21, 0.15, 0.61]
Dunn Index	$0.008 \pm 0.05$ [1.2e - 5, 2.3e - 4, 0.35]	<b>0.01</b> $\pm$ <b>0.01</b> [0.01, 0.01, 0.01]	$0.001 \pm 0.001$ [0.001, 0.001, 0.001]	$0.001 \pm 0.001$ [0.001, 0.001, 0.001]
Purity	$0.65 \pm 0.05$ [0.47, 0.66, 0.70]	$0.65 \pm 0.01$ [0.62, 0.65, 0.65]	$0.64 \pm 0.02$ [0.55, 0.64, 0.76]	<b>0.84</b> $\pm$ <b>0.15</b> [0.46, 0.81, 1]
Entropy	$0.91 \pm 0.04$ [0.84, 0.91, 0.99]	$0.92 \pm 0.01$ [0.92, 0.92, 0.94]	$0.92 \pm 0.02$ [0.76, 0.92, 0.98]	<b>0.45</b> $\pm$ <b>0.40</b> [0.69, 0.99]
NMI	$0.10 \pm 0.04$ [1.3e - 4, 0.09, 0.18]	$0.08 \pm 0.01$ [0.05, 0.08, 0.08]	$0.08 \pm 0.03$ [0.003, 0.07, 0.24]	<b>0.54</b> $\pm$ <b>0.40</b> [0.30, 1]
Categorical				
Algorithm				
DB Index	<b>1.41</b> $\pm$ <b>0.31</b> [0.97, 1.38, 1.95]	$1.78 \pm 0.21$ [1.49, 1.93, 1.93]	$2.07 \pm 0.15$ [1.19, 2.10, 2.19]	$2.51 \pm 0.75$ [1.14, 2.27, 4.36]
Silhouette Index	$0.23 \pm 0.05$ [0.16, 0.23, 0.36]	<b>0.24</b> $\pm$ <b>0.01</b> [0.24, 0.24, 0.31]	$0.03 \pm 0.02$ [0.02, 0.03, 0.15]	$0.17 \pm 0.06$ [0.02, 0.15, 0.36]
Dunn Index	<b>0.12</b> $\pm$ <b>0.03</b> [0.11, 0.11, 0.22]	$0.01 \pm 0.001$ [0.01, 0.01, 0.01]	$0.001 \pm 0.001$ [0.001, 0.001, 0.001]	$0.05 \pm 0.05$ [0, 0, 0.22]
Purity	$0.73 \pm 0.08$ [0.54, 0.77, 0.80]	$0.76 \pm 0.03$ [0.58, 0.76, 0.76]	$0.64 \pm 0.02$ [0.55, 0.64, 0.76]	<b>0.84</b> $\pm$ <b>0.15</b> [0.46, 0.81, 1]
Entropy	$0.80 \pm 0.08$ [0.70, 0.78, 0.98]	$0.72 \pm 0.03$ [0.71, 0.71, 0.92]	$0.92 \pm 0.03$ [0.76, 0.92, 0.98]	<b>0.45</b> $\pm$ <b>0.40</b> [0.69, 0.99]
NMI	$0.19 \pm 0.08$ [0.01, 0.22, 0.30]	$0.28 \pm 0.03$ [0.08, 0.28, 0.28]	$0.08 \pm 0.02$ [0.003, 0.07, 0.24]	<b>0.54</b> $\pm$ <b>0.40</b> [0.30, 1]
Multiview K-means				
Algorithm				
DB Index	<b>1.41</b> $\pm$ <b>0.31</b> [0.97, 1.38, 1.95]	$1.78 \pm 0.21$ [1.49, 1.93, 1.93]	$2.07 \pm 0.15$ [1.19, 2.10, 2.19]	$2.51 \pm 0.75$ [1.14, 2.27, 4.36]
Silhouette Index	$0.23 \pm 0.05$ [0.16, 0.23, 0.36]	<b>0.24</b> $\pm$ <b>0.01</b> [0.24, 0.24, 0.31]	$0.03 \pm 0.02$ [0.02, 0.03, 0.15]	$0.17 \pm 0.06$ [0.02, 0.15, 0.36]
Dunn Index	<b>0.12</b> $\pm$ <b>0.03</b> [0.11, 0.11, 0.22]	$0.01 \pm 0.001$ [0.01, 0.01, 0.01]	$0.001 \pm 0.001$ [0.001, 0.001, 0.001]	$0.05 \pm 0.05$ [0, 0, 0.22]
Purity	$0.73 \pm 0.08$ [0.54, 0.77, 0.80]	$0.76 \pm 0.03$ [0.58, 0.76, 0.76]	$0.64 \pm 0.02$ [0.55, 0.64, 0.76]	<b>0.84</b> $\pm$ <b>0.15</b> [0.46, 0.81, 1]
Entropy	$0.80 \pm 0.08$ [0.70, 0.78, 0.98]	$0.72 \pm 0.03$ [0.71, 0.71, 0.92]	$0.92 \pm 0.03$ [0.76, 0.92, 0.98]	<b>0.45</b> $\pm$ <b>0.40</b> [0.69, 0.99]
NMI	$0.19 \pm 0.08$ [0.01, 0.22, 0.30]	$0.28 \pm 0.03$ [0.08, 0.28, 0.28]	$0.08 \pm 0.02$ [0.003, 0.07, 0.24]	<b>0.54</b> $\pm$ <b>0.40</b> [0.30, 1]
Multiview K-means				

(b) Comparison of the proposed IDS approaches to the ensemble and multiview clustering algorithms.

**Table 4.16:** Clustering results for the Credit card approval data set (50 runs,  $k = 2$  clusters per domain). We run the seed-based IDS with the following parameters:  $k = 2$  clusters and number of seeds, and the constraint-based IDS:  $n_{T_1} = 5$ ,  $n_{T_2} = 11$ , and  $t_{T_1} = t_{T_2} = 1$ .

Data type Algorithm	Text			
	Seed-based IDS	Constraint-based IDS	Splitting	Conversion
DB Index	$2.29 \pm 0.06[2.22, 2.28, 2.40]$	$2.13 \pm 0.02[2.11, 2.13, 2.17]$	$2.12 \pm 0.03[2.09, 2.12, 2.19]$	$2.51 \pm 0.05[2.39, 2.52, 2.57]$
Silhouette Index	$0.01 \pm 0.002[0.007, 0.011, 0.016]$	$0.02 \pm 0.001[0.02, 0.02, 0.02]$	$0.02 \pm 0.01[0.01, 0.019, 0.02]$	$0.05 \pm 0.001[0.05, 0.05, 0.06]$
Dunn Index	$0.01 \pm 0.006[0.007, 0.01, 0.02]$	$0.001 \pm 0.001[0.001, 0.001, 0.001]$	$0.017 \pm 0.007[0.008, 0.02, 0.02]$	$0.09 \pm 0.03[0.07, 0.08, 0.13]$
Tags DB Index	$48.58 \pm 14.29[26.27, 43.89, 69.95]$	$43.51 \pm 10.67[25.54, 44.58, 55.37]$	$43.09 \pm 8.67[25.76, 42.28, 54.83]$	$74.08 \pm 8.13[63.88, 73.37, 93.12]$
Tags Silhouette Index	$0.16 \pm 0.03[0.11, 0.17, 0.21]$	$0.11 \pm 0.02[0.08, 0.10, 0.14]$	$0.11 \pm 0.02[0.09, 0.12, 0.14]$	$0.09 \pm 0.006[0.07, 0.09, 0.10]$
Tags Dunn Index	$0.001 \pm 0[0.001, 0.001, 0.001]$	$0.001 \pm 0.001[0.001, 0.001, 0.001]$	$0.001 \pm 0.001[0.001, 0.001, 0.001]$	$0.001 \pm 0.001[0.001, 0.001, 0.001]$
Data type Algorithm	Visual			
	Seed-based IDS	Constraint-based IDS	Splitting	Conversion
DB Index	$2.20 \pm 0.14[1.99, 2.20, 2.52]$	$1.93 \pm 0.05[1.87, 1.9263, 2.0223]$	$1.90 \pm 0.03[1.87, 1.89, 1.96]$	$2.51 \pm 0.05[2.39, 2.52, 2.57]$
Silhouette Index	$0.10 \pm 0.01[0.08, 0.10, 0.11]$	$0.12 \pm 0.002[0.11, 0.12, 0.12]$	$0.11 \pm 0.001[0.11, 0.11, 0.12]$	$0.05 \pm 0.001[0.05, 0.05, 0.06]$
Dunn Index	$0.005 \pm 0.001[0.004, 0.005, 0.005]$	$0.008 \pm 0.006[0.0004, 0.012, 0.015]$	$0.01 \pm 0.002[0.01, 0.01, 0.011]$	$0.09 \pm 0.03[0.07, 0.08, 0.13]$
Tags DB Index	$69.55 \pm 13.91[53.02, 66.85, 93.60]$	$77.38 \pm 8.77[66.34, 78.32, 91.73]$	$74.36 \pm 5.29[69.59, 72.45, 83.82]$	$74.08 \pm 8.13[63.88, 73.37, 93.12]$
Tags Silhouette Index	$0.09 \pm 0.01[0.09, 0.09, 0.10]$	$0.09 \pm 0.008[0.08, 0.092, 0.10]$	$0.09 \pm 0.01[0.08, 0.09, 0.10]$	$0.09 \pm 0.006[0.07, 0.09, 0.10]$
Tags Dunn Index	$0.001 \pm 0.001[0.001, 0.001, 0.001]$	$0.001 \pm 0.001[0.001, 0.001, 0.001]$	$0.001 \pm 0.001[0.001, 0.001, 0.001]$	$0.001 \pm 0.001[0.001, 0.001, 0.001]$

(a) Comparison of the proposed IDS approaches with the traditional splitting and conversion algorithms.

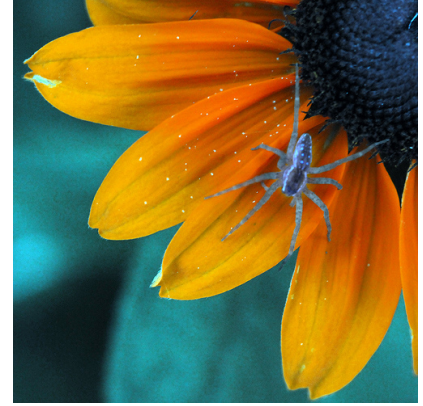
Data type Algorithm	Text			
	Seed-based IDS	Constraint-based IDS	Ensemble: Voting	Ensemble: Clustering
DB Index	$2.29 \pm 0.06[2.22, 2.28, 2.40]$	$2.13 \pm 0.02[2.11, 2.13, 2.17]$	$5.26 \pm 1.07[3.65, 5.30, 7.72]$	$7.99 \pm 1.20[5.98, 8.00, 9.67]$
Silhouette Index	$0.01 \pm 0.002[0.007, 0.011, 0.016]$	$0.02 \pm 0.001[0.02, 0.02, 0.02]$	$0.003 \pm 0.001[0.001, 0.003, 0.004]$	$0.10 \pm 0.01[0.08, 0.10, 0.14]$
Dunn Index	$0.01 \pm 0.006[0.007, 0.01, 0.02]$	$0.001 \pm 0.001[0.001, 0.001, 0.001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$
Tags DB Index	$48.58 \pm 14.29[26.27, 43.89, 69.95]$	$43.51 \pm 10.67[25.54, 44.58, 55.37]$	$53.26 \pm 15.33[34.98, 50.52, 89.84]$	$48.64 \pm 8.30[38.82, 46.15, 67.70]$
Tags Silhouette Index	$0.16 \pm 0.03[0.11, 0.17, 0.21]$	$0.11 \pm 0.02[0.08, 0.10, 0.14]$	$0.10 \pm 0.01[0.08, 0.10, 0.12]$	$0.11 \pm 0.01[0.08, 0.11, 0.13]$
Tags Dunn Index	$0.001 \pm 0[0.001, 0.001, 0.001]$	$0.001 \pm 0.001[0.001, 0.001, 0.001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$
Data type Algorithm	Visual			
	Seed-based IDS	Constraint-based IDS	Ensemble: Voting	Ensemble: Clustering
DB Index	$2.20 \pm 0.14[1.99, 2.20, 2.52]$	$1.93 \pm 0.05[1.87, 1.9263, 2.0223]$	$5.39 \pm 1.83[3.00, 5.60, 8.09]$	$21.01 \pm 10.22[11.72, 18.51, 44.37]$
Silhouette Index	$0.10 \pm 0.01[0.08, 0.10, 0.11]$	$0.12 \pm 0.002[0.11, 0.12, 0.12]$	$0.031 \pm 0.018[0.002, 0.0323, 0.0591]$	$0.14 \pm 0.02[0.12, 0.14, 0.19]$
Dunn Index	$0.005 \pm 0.001[0.004, 0.005, 0.005]$	$0.008 \pm 0.006[0.0004, 0.012, 0.015]$	$0.0005 \pm 0.0007[0.0004, 0.0025]$	$0.001 \pm 0.001[0.0001, 0.0004, 0.005]$
Tags DB Index	$69.55 \pm 13.91[53.02, 66.85, 93.60]$	$77.38 \pm 8.77[66.34, 78.32, 91.73]$	$53.26 \pm 15.33[34.98, 50.52, 89.84]$	$48.64 \pm 8.30[38.82, 46.15, 67.70]$
Tags Silhouette Index	$0.09 \pm 0.01[0.09, 0.09, 0.10]$	$0.09 \pm 0.008[0.08, 0.092, 0.10]$	$0.10 \pm 0.01[0.08, 0.10, 0.12]$	$0.11 \pm 0.01[0.08, 0.11, 0.13]$
Tags Dunn Index	$0.001 \pm 0.001[0.001, 0.001, 0.001]$	$0.001 \pm 0.001[0.001, 0.001, 0.001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$

(b) Comparison of the proposed IDS approaches to the ensemble and multiview clustering algorithms.

**Table 4.17:** Clustering results for the MIRFlickr data set (10 runs,  $k = 16$  clusters per domain). We run the seed-based IDS with the following parameters:  $k = 2$  clusters and number of seeds, and the constraint-based IDS:  $n_{T_1} = n_{T_2} = 5$ ,  $t_{T_1} = t_{T_2} = 1$ .



(a) Text: graffiti.



(b) Text: flower, green, orange, petal, spider, yellow.



(c) Text: grey, horse, friend.



(d) Text: animal, close up, detail, flower, insect, red, water.

**Figure 4.11:** Sample data from the compatible clusters.

of the constraints are satisfied and the value of each of the penalty terms approaches zero. This explains why the objective function of the constraint-based IDS always starts from a higher value than others but quickly reaches a lower value, reflecting a better clustering.

### 4.3 Results of the Compatibility Analysis Experiments

Using the methodology described in Section 3.4, we extracted two subsets of the MIRFlickr data set depending on whether the domains were compatible or incompatible. The compatible subset consists of 2,679 data records, while the size of the incompatible subset was 20,751 data records. Since the incompatible subset had almost eight times as many data records as the compatible set, we used only 2,679 randomly selected data records to perform our experiments, and therefore get comparable metrics that are not biased by the size of the data sets. We also randomly selected 2,679 data records from the entire data set, and called this set the *mixed* set. Figures 4.11 and 4.12 illustrate four randomly selected images from the data along with their text data from the compatible and incompatible subsets, respectively. To show the importance of the domain compatibility in



(a) Text: hawaii, light house, vacation.



(b) Text: gold, record, sing, vintage, vinyl.

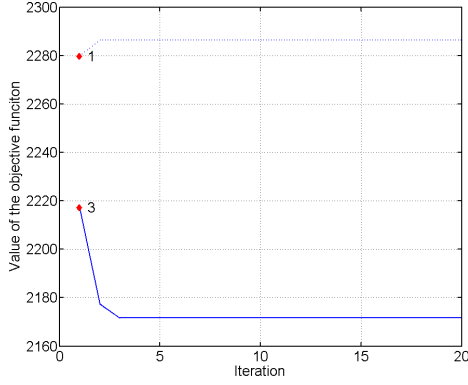


(c) Text: canon, japan, flower, rainy.

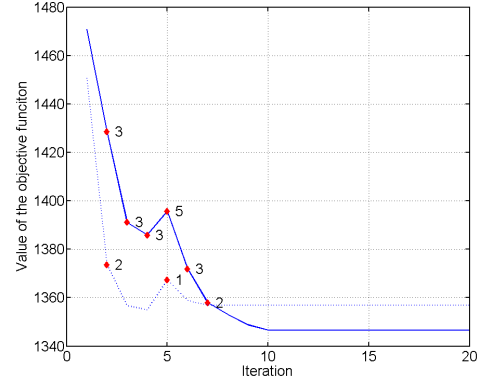


(d) Text: Indonesia, sun, temple.

**Figure 4.12:** Sample data from the incompatible clusters.



(a) MIRFlickr data set: text domain



(b) MIRFlickr data set: visual domain.

**Figure 4.13:** Value of the objective function (and number of exchange seeds) within seed-based IDS for the compatible (solid blue line) and incompatible (dashed blue line) sets. Red diamonds represent a seed exchange between domains. The maximum number of seed exchanged between domains is 16, same as the number of clusters.

clustering heterogeneous data, we performed three similar experiments for each clustering technique. In the first experiment, we used only data from the mixed subset; in the second experiment, we used data from the incompatible subset; and in the third, we used data from the compatible subset. We repeated each experiment 10 times and as before, we report the validity indices in the format:  $\text{mean} \pm \text{std}[\text{min}, \text{median}, \text{max}]$ . The results of these experiments are described below:

**Seed-based IDS:** Table 4.18 shows the results for the compatible set which significantly outperform the results for the other subsets with better results in terms of all internal and external validity measures. Figure 4.13 shows a typical run of the seed-based IDS for the compatible and incompatible sets. The seed-based IDS results are better in the compatible set than in the incompatible set, showing that the proposed IDS approach is more active and involves more seed exchanges between domains in the compatible set. This happens because the cluster structure in the text domain and in the visual domain agree in the compatible set: an image-text pair is always assigned to the same cluster in both domains and every seed exchange guides the clustering process a better clustering. We also observe that the visual domain benefits from more exchanged seeds received from the text domain (Figure 4.13b).

**Constraint-based IDS:** Again, Table 4.19 shows that the proposed constraint-based IDS approach for the compatible set outperforms the results for the mixed and incompatible sets in both the text and visual domains. Figures 4.14, 4.15, and 4.16 show the improvement or decay of the constrained-based IDS with respect to the different number of exchange points in the different

Data type	Text domain		
Algorithm	Mixed set	Incompatible set	Compatible Set
DB Index	$2.11 \pm 0.01[1.96, 2.10, 2.25]$	$2.14 \pm 0.08[2.02, 2.12, 2.27]$	<b><math>2.05 \pm 0.04[2.00, 2.04, 2.13]</math></b>
Silhouette Index	$0.01 \pm 0.006[0.002, 0.009, 0.018]$	$0.009 \pm 0.002[0.007, 0.009, 0.01]$	<b><math>0.02 \pm 0.003[0.013, 0.018, 0.025]</math></b>
Dunn Index	$0.055 \pm 0.04[0.03, 0.03, 0.11]$	$0.044 \pm 0.002[0.042, 0.043, 0.049]$	<b><math>0.076 \pm 0.008[0.02, 0.06, 0.09]</math></b>
Tags DB Index	$25.64 \pm 6.60[16.37, 24.77, 34.88]$	$28.34 \pm 5.99[18.90, 28.84, 39.24]$	<b><math>20.87 \pm 3.14[13.95, 20.61, 25.09]</math></b>
Tags Silhouette Index	$0.13 \pm 0.06[0.08, 0.13, 0.16]$	$0.13 \pm 0.034[0.08, 0.13, 0.16]$	<b><math>0.16 \pm 0.03[0.10, 0.16, 0.19]</math></b>
Tags Dunn Index	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>
Data type	Visual domain		
Algorithm	Mixed set	Incompatible set	Compatible Set
DB Index	$2.25 \pm 0.22[2.02, 2.20, 2.73]$	$2.28 \pm 0.16[2.06, 2.30, 2.46]$	<b><math>2.14 \pm 0.15[1.80, 2.14, 2.32]</math></b>
Silhouette Index	$0.09 \pm 0.008[0.07, 0.09, 0.10]$	$0.09 \pm 0.009[0.08, 0.09, 0.11]$	<b><math>0.11 \pm 0.01[0.10, 0.11, 0.13]</math></b>
Dunn Index	$0.027 \pm 0.019[0.0024, 0.021, 0.06]$	$0.027 \pm 0.02[0.02, 0.03, 0.06]$	<b><math>0.04 \pm 0.017[0.004, 0.042, 0.054]</math></b>
Tags DB Index	$40.76 \pm 7.60[32.00, 38.3, 53.89]$	$37.10 \pm 4.04[31.40, 36.06, 44.94]$	<b><math>26.28 \pm 4.52[20.5769, 26.5659, 34.1164]</math></b>
Tags Silhouette Index	$0.09 \pm 0.01[0.07, 0.09, 0.09]$	$0.08 \pm 0.01[0.06, 0.08, 0.08]$	<b><math>0.10 \pm 0.01[0.07, 0.10, 0.10]</math></b>
Tags Dunn Index	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>

**Table 4.18:** Clustering results of the seed-based IDS (with DB-exchange) for the mixed, incompatible, and compatible sets (10 runs,  $k = 16$  clusters per domain).

Data type	Text domain		
Algorithm	Mixed set	Incompatible set	Compatible Set
DB Index	$2.01 \pm 0.01[2.00, 2.01, 2.03]$	$2.04 \pm 0.01[2.03, 2.04, 2.06]$	<b><math>1.92 \pm 0.01[1.91, 1.92, 1.96]</math></b>
Silhouette Index	$0.016 \pm 0.001[0.015, 0.016, 0.016]$	$0.015 \pm 0.001[0.014, 0.015, 0.017]$	<b><math>0.03 \pm 0.001[0.026, 0.028, 0.03]</math></b>
Dunn Index	$0.01 \pm 0.014[0, 0.01, 0.03]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	<b><math>0.03 \pm 0.026[0, 0.05, 0.05]</math></b>
Tags DB Index	$36.14 \pm 1.11[34.11, 36.37, 37.37]$	$31.1629 \pm 1.5180[28.76, 30.91, 34.1797]$	<b><math>18.05 \pm 1.54[16.42, 17.30, 20.48]</math></b>
Tags Silhouette Index	$0.1102 \pm 0.0021[0.10, 0.11, 0.11]$	$0.0745 \pm 0.0007[0.074, 0.074, 0.075]$	<b><math>0.15 \pm 0.003[0.14, 0.15, 0.15]</math></b>
Tags Dunn Index	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>
Data type	Visual domain		
Algorithm	Mixed set	Incompatible set	Compatible Set
DB Index	$1.93 \pm 0.04[1.89, 1.92, 1.2.07]$	$2.14 \pm 0.05[1.92, 2.01, 2.17]$	<b><math>1.91 \pm 0.02[1.89, 1.91, 2.00]</math></b>
Silhouette Index	$0.10 \pm 0.002[0.10, 0.10, 0.11]$	$0.1076 \pm 0.0021[0.1054, 0.1072, 0.1111]$	<b><math>0.12 \pm 0.002[0.12, 0.12, 0.13]</math></b>
Dunn Index	$0.0251 \pm 0.0142[0.0205, 0.0206, 0.0654]$	$0.0140 \pm 0.021[0.0008, 0.0008, 0.0582]$	<b><math>0.05 \pm 0.014[0.04, 0.04, 0.07]</math></b>
Tags DB Index	$38.73 \pm 3.0666[34.86, 37.80, 45.08]$	$37.42 \pm 4.1019[31.02, 37.31, 46.11]$	<b><math>30.12 \pm 3.79[22.60, 30.76, 37.14]</math></b>
Tags Silhouette Index	$0.0942 \pm 0.0057[0.08, 0.09, -0.10]$	$0.094 \pm 0.004[0.09, 0.09, 0.091]$	<b><math>0.099 \pm 0.005[0.0939, 0.0986, 0.10]</math></b>
Tags Dunn Index	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>

**Table 4.19:** Clustering results of the constraint-based IDS for the mixed, incompatible, and compatible sets (10 runs,  $k = 16$  clusters per domain,  $n_{T_1} = 11$ ,  $n_{T_2} = 5$ , and  $t_{T_1} = t_{T_2} = 1$ ).

domains for the compatible set of the MIRFlickr data set over the baseline splitting algorithm, incompatible, and mixed sets, respectively.

- In Figure 4.14, the heat maps (a) and (b) in each sub-figure show the percent-wise improvement (or decline) over the value of the tag DB index for the splitting algorithm,

$$\Delta db^{Tag} = - \frac{(db_{compatible}^{Tag} - db_{splitting}^{Tag})}{db_{splitting}^{Tag}} 100\%. \quad (4.5)$$

with respect to the number of exchange points from the different domains, while the heat maps (c) and (d) show the same improvement (or decline) in terms of the DB index, i.e.

$$\Delta db = - \frac{(db_{compatible} - db_{splitting})}{db_{splitting}} 100\%. \quad (4.6)$$

- In Figure 4.15, the heat maps (a) and (b) in each sub-figure show the percent-wise improvement

(or decline) over the value of the tag DB index for the mixed set,

$$\Delta db_{mixed}^{Tag} = -\frac{(db_{compatible}^{Tag} - db_{mixed}^{Tag})}{db_{mixed}^{Tag}} 100\%. \quad (4.7)$$

with respect to the number of exchange points from the different domains, while the heat maps (c) and (d) show the same improvement (or decline) in terms of the DB index, i.e.

$$\Delta db_{mixed} = -\frac{(db_{compatible} - db_{mixed})}{db_{mixed}} 100\%. \quad (4.8)$$

- In Figure 4.16, the heat maps (a) and (b) in each sub-figure show the percent-wise improvement (or decline) over the value of the tag DB index for the incompatible set,

$$\Delta db_{incompatible}^{Tag} = -\frac{(db_{compatible}^{Tag} - db_{incompatible}^{Tag})}{db_{incompatible}^{Tag}} 100\%. \quad (4.9)$$

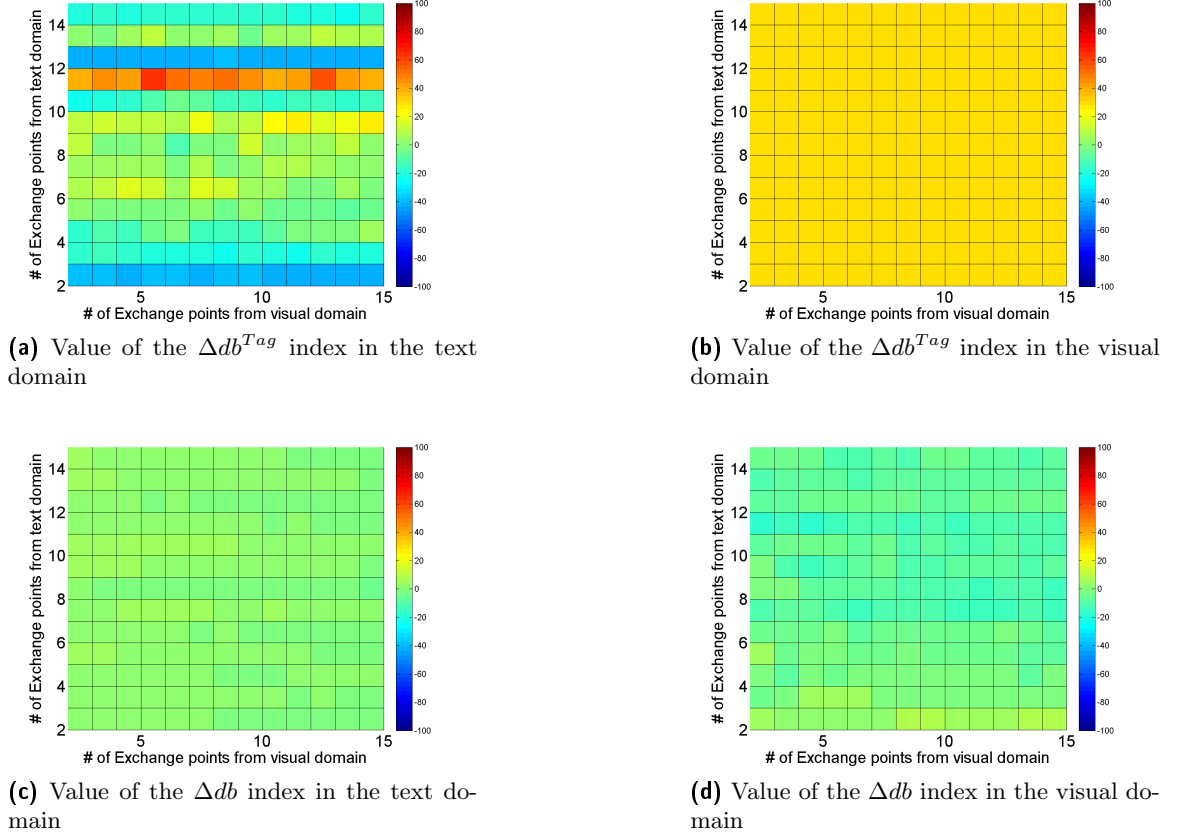
with respect to the number of exchange points from the different domains, while the heat maps (c) and (d) show the same improvement (or decline) in terms of the DB index, i.e.

$$\Delta db_{incompatible} = -\frac{(db_{compatible} - db_{incompatible})}{db_{incompatible}} 100\%. \quad (4.10)$$

The color bar on the heat map ranges from  $-100\%$  (decay) to  $100\%$  (improvement), starting from the dark blue color (decay), continuing to the “cold” colors (neutral), then reaching to the “warm” colors (improvement) and ending with the dark red. Each point in the heat map is a result of an independent run of the constraint-based IDS approach, the size of each map is  $14 \times 14$ , making in total 196 experiments for each set, with a minimum number of exchange points per cluster,  $n_{T,min} = 2$  and maximum  $n_{T,max} = 15$ .

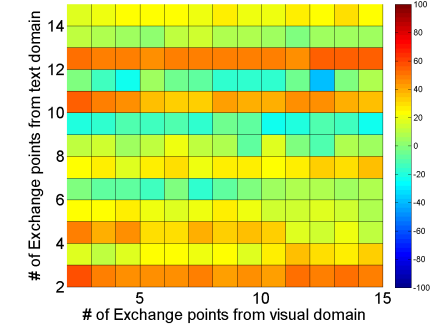
- Figure 4.14a shows that the value of the  $\Delta db^{Tag}$  in the text domain gradually increases with an increase in the number of exchange points from the text domain, except for one strip corresponding to the number of exchange points in the text domain,  $n_{T_1} = 12$ , and any number of exchange points,  $n_{T_2}$ , in the visual domain. The performance of the constraint-based IDS in the text domain is not effected by the exchange points coming from the visual domain. We obtained a maximum improvement of 64% over the baseline splitting algorithm with  $n_{T_1} = 11$  exchange points in the text domain and  $n_{T_2} = 5$  exchange points in the visual domain. Figure 4.14b shows that any value of the exchange points coming from both domains helps to improve

the results compared to the baseline splitting algorithm by 20 – 30%. Figures 4.14c and 4.14d show a smooth heat surface, indicating the algorithm’s stability, and an overall (up to 20%) improvement over the results of the baseline splitting algorithm.

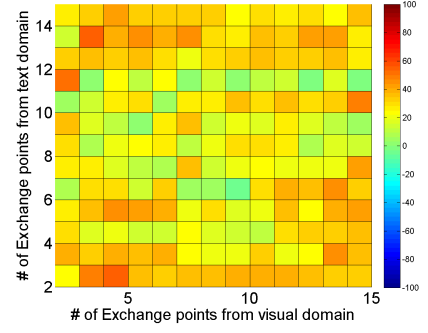


**Figure 4.14:** Constraint-based IDS ( $k_{T_1} = k_{T_2} = 16$ ,  $t_{T_1} = t_{T_2} = 1$ ): effect of the number of constraints in the compatible set of the MIRFlickr data set. Warm colors indicate improvement and cold colors indicate decline over the baseline splitting algorithm.

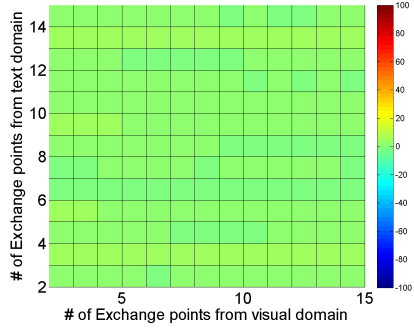
- Figures 4.15a and 4.15b show the overall improvement of the results of the compatible set over the results of the mixed set in the text and visual domains, respectively. Figures 4.15c and 4.15d show a smooth heat surface, indicating the algorithm’s stability, and an overall (up to 20%) improvement over the results of the results of the mixed set.
- The heat maps in Figure 4.16 are similar to the previous heat maps in Figure 4.15, but with a higher improvement over the results of the incompatible set. Figure 4.16.a shows up to 80% improvement with  $n_{T_1} = 12$  exchange points in the text domain and  $n_{T_2} = 13$  exchange points in the visual domain. Again, the values of the improvement of the internal DB index  $\Delta db_{incompatible}$  in the text and visual domains show a smooth heat surface, indicating



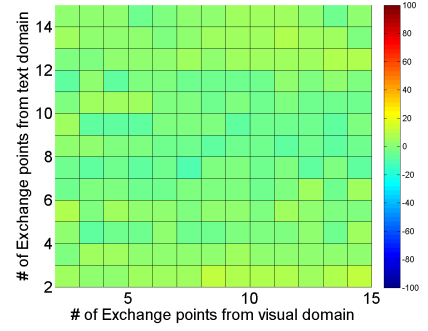
(a) Value of the  $\Delta db_{mixed}^{Tag}$  index in the text domain



(b) Value of the  $\Delta db_{mixed}^{Tag}$  index in the visual domain



(c) Value of the  $\Delta db_{mixed}$  index in the text domain



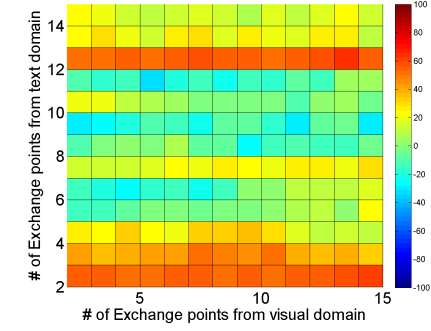
(d) Value of the  $\Delta db_{mixed}$  index in the visual domain

**Figure 4.15:** Constraint-based IDS ( $k_{T_1} = k_{T_2} = 16$ ,  $t_{T_1} = t_{T_2} = 1$ ): effect of the number of constraints in the compatible set of the MIRFlickr data set. Warm colors indicate improvement and cold colors indicate decline over the mixed set.

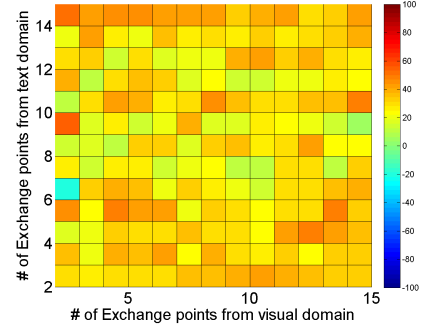
the algorithm's stability, and an overall (up to 30%) improvement over the results of the incompatible set.

Figure 4.17 shows the value of the objective function for the constraint-based IDS for the compatible and incompatible sets. The value of the objective function of the compatible set is 5% lower than the value of the objective function for the incompatible set, which indicates a better clustering.

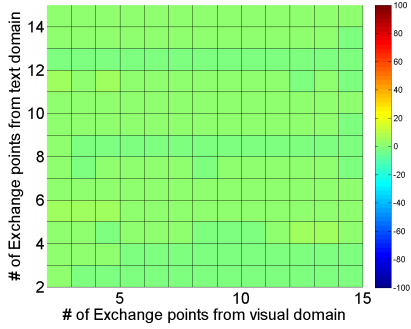
**Conversion, Splitting, Ensemble, and Multiview clustering:** Tables 4.20, 4.21, 4.22, and 4.23 show the clustering results of the mixed, incompatible, and compatible sets for the conversion, splitting, ensemble, and multiview clustering algorithms, respectively. As we can see from these tables, the results for the compatible set dominate over the results for the incompatible and mixed sets.



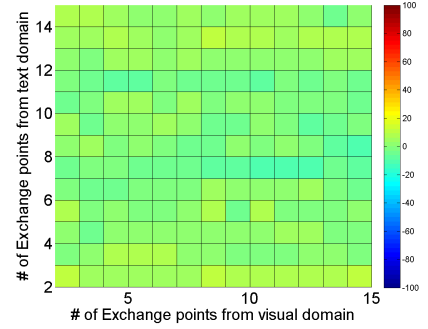
(a) Value of the  $\Delta db_{incompatible}^{Tag}$  index in the text domain



(b) Value of the  $\Delta db_{incompatible}^{Tag}$  index in the visual domain



(c) Value of the  $\Delta db_{incompatible}$  index in the text domain



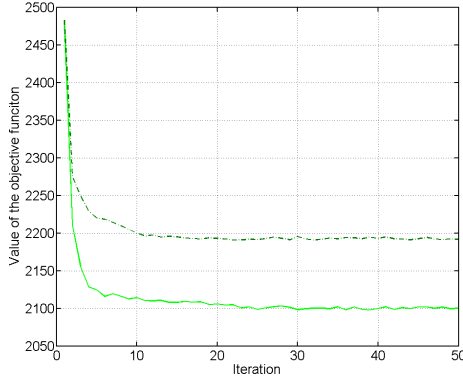
(d) Value of the  $\Delta db_{incompatible}$  index in the visual domain

**Figure 4.16:** Constraint-based IDS ( $k_{T_1} = k_{T_2} = 16$ ,  $t_{T_1} = t_{T_2} = 1$ ): effect of the number of constraints in the compatible set of the MIRFlickr data set. Warm colors indicate improvement and cold colors indicate decline over the incompatible set.

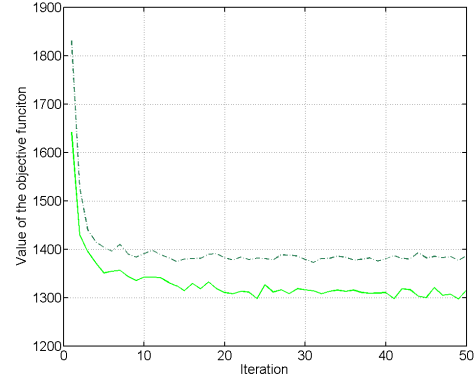
## 4.4 Application: Image Auto-Annotation

For the purpose of application, we performed two sets of experiments:

- In the first set of experiments, we split the MIRFlickr data set in two sets: training and testing. The size of the training set is 22,430 data records and the size of the testing set is 1,000 data records. We cluster the training set into  $k = 50, 100, 200$ , and 300 clusters. In the seed-based IDS, constraint-based IDS, conversion, multiview, and ensemble clustering algorithms, we cluster the text and visual domains together, and in the splitting algorithm, we cluster only the visual domain.
- In the second set of experiments, we first performed a compatibility analysis and then split the MIRFlickr data set in two training sets and one test set. The size of the first training set is 2,629 data records consisting of data records from the compatible set, while the second



(a) MIRFlickr data set: text domain



(b) MIRFlickr data set: visual domain.

**Figure 4.17:** Value of the objective function for constraint-based IDS for the compatible (solid green line) and incompatible (dashed dark green line) sets ( $k = 16$  clusters per domain,  $n_{T_1} = 5$ ,  $n_{T_2} = 5$ , and  $t_{T_1} = t_{T_2} = 1$ ).

Data type	Text domain		
Algorithm	Mixed set	Incompatible set	Compatible Set
DB Index	$2.51 \pm 0.13[2.29, 2.52, 2.72]$	$2.55 \pm 0.12[2.42, 2.50, 2.75]$	<b><math>2.43 \pm 0.07[2.34, 2.44, 2.55]</math></b>
Silhouette Index	$0.053 \pm 0.001[0.051, 0.053, 0.054]$	$0.053 \pm 0.002[0.048, 0.053, 0.056]$	<b><math>0.068 \pm 0.004[0.06, 0.068, 0.074]</math></b>
Dunn Index	$0.2449 \pm 0.0294[0.1891, 0.2588, 0.2588]$	$0.1760 \pm 0.0515[0.1295, 0.1597, 0.2433]$	<b><math>0.28 \pm 0.0161[0.2462, 0.2825, 0.3038]</math></b>
Tags DB Index	$41.75 \pm 4.12[34.48, 42.42, 47.68]$	$43.85 \pm 5.18[35.42, 43.15, 55.84]$	<b><math>27.78 \pm 6.54[17.01, 29.61, 35.33]</math></b>
Tags Silhouette Index	<b><math>0.10 \pm 0.01[0.08, 0.10, 0.11]</math></b>	$0.095 \pm 0.008[0.08, 0.09, 0.10]$	$0.10 \pm 0.007[0.08, 0.089, 0.10]$
Tags Dunn Index	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>
Data type	Visual domain		
Algorithm	Mixed set	Incompatible set	Compatible Set
DB Index	<b><math>2.44 \pm 0.07[2.34, 2.44, 2.55]</math></b>	$2.55 \pm 0.12[2.42, 2.49, 2.75]$	$2.51 \pm 0.13[2.29, 2.51, 2.72]$
Silhouette Index	$0.05 \pm 0.001[0.05, 0.05, 0.05]$	$0.053 \pm 0.0023[0.05, 0.054, 0.06]$	<b><math>0.068 \pm 0.005[0.059, 0.069, 0.074]</math></b>
Dunn Index	<b><math>0.28 \pm 0.016[0.24, 0.28, 0.30]</math></b>	$0.25 \pm 0.03[0.19, 0.26, 0.26]$	$0.18 \pm 0.05[0.13, 0.16, 0.24]$
Tags DB Index	$41.75 \pm 4.12[34.48, 42.42, 47.68]$	$43.85 \pm 5.18[35.42, 43.15, 55.84]$	<b><math>27.78 \pm 6.54[17.01, 29.61, 35.33]</math></b>
Tags Silhouette Index	<b><math>0.10 \pm 0.01[0.08, 0.10, 0.11]</math></b>	$0.095 \pm 0.008[0.08, 0.09, 0.10]$	$0.10 \pm 0.007[0.08, 0.089, 0.10]$
Tags Dunn Index	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>

**Table 4.20:** Clustering results of the conversion algorithm for the mixed, incompatible, and compatible sets. (10 runs,  $k = 16$  clusters per domain).

training set consisted of 2,629 randomly selected data records from the mixed set. The test set consisted of 50 data records from the compatible set and 50 data records from the mixed sets, making in total 100 data records. We cluster both training sets into  $k = 16, 32, 50, 80$ , and 100 clusters. We cluster the compatible training set using the seed-based and constraints-based IDS, and the mixed training set using the conversion, multiview, ensemble, and splitting clustering algorithms because these algorithms do not have any compatibility analysis in their original definitions.

The image auto-annotation process then proceeds as follows:

1. First, we cluster a training set with a corresponding number of clusters.
2. Then for the image auto-annotation, we used two different nearest-neighbor (NN) schemes [Cover and Hart, 1967] (see Figure 4.18):

Data type	Text domain		
Algorithm	Mixed set	Incompatible set	Compatible Set
DB Index	$2.04 \pm 0.39[1.98, 2.02, 2.11]$	$2.04 \pm 0.0251[1.99, 2.05, 2.07]$	<b><math>2.03 \pm 0.04[1.92, 1.97, 2.06]</math></b>
Silhouette Index	$0.017 \pm 0.0012[0.015, 0.017, 0.019]$	$0.016 \pm 0.001[0.014, 0.017, 0.018]$	<b><math>0.018 \pm 0.002[0.015, 0.017, 0.019]</math></b>
Dunn Index	$0.055 \pm 0.031[0.031, 0.031, 0.103]$	<b><math>0.071 \pm 0.014[0.063, 0.066, 0.093]</math></b>	$0.04 \pm 0.007[0.04, 0.04, 0.05]$
Tags DB Index	$31.46 \pm 5.59[24.52, 30.92, 44.58]$	$27.33 \pm 4.51[20.32, 27.31, 36.03]$	<b><math>24.13 \pm 5.63[19.31, 22.43, 36.97]</math></b>
Tags Silhouette Index	<b><math>0.12 \pm 0.03[0.08, 0.12, 0.17]</math></b>	$0.12 \pm 0.016[0.09, 0.11, 0.14]$	$0.08 \pm 0.01[0.05, 0.08, 0.12]$
Tags Dunn Index	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$
Data type	Visual domain		
Algorithm	Mixed set	Incompatible set	Compatible Set
DB Index	$1.90 \pm 0.05[1.83, 1.90, 1.97]$	$1.91 \pm 0.06[1.83, 1.89, 2.00]$	<b><math>1.90 \pm 0.08[1.77, 1.91, 2.04]</math></b>
Silhouette Index	$0.117 \pm 0.002[0.11, 0.12, 0.12]$	$0.12 \pm 0.003[0.11, 0.12, 0.12]$	<b><math>0.13 \pm 0.01[0.12, 0.13, 0.14]</math></b>
Dunn Index	$0.046 \pm 0.034[0.02, 0.02, 0.10]$	<b><math>0.09 \pm 0.01[0.08, 0.09, 0.11]</math></b>	$0.06 \pm 0.01[0.05, 0.05, 0.08]$
Tags DB Index	$41.55 \pm 4.57[34.17, 40.85, 49.12]$	$42.58 \pm 5.29[36.14, 42.18, 52.31]$	<b><math>28.84 \pm 6.88[19.60, 22.39, 40.57]</math></b>
Tags Silhouette Index	$0.09 \pm 0.007[0.08, 0.09, 0.10]$	$0.09 \pm 0.008[0.07, 0.09, 0.10]$	<b><math>0.10 \pm 0.01[0.08, 0.1179, 0.10, 0.12]</math></b>
Tags Dunn Index	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>

**Table 4.21:** Clustering results of the splitting algorithm for the mixed, incompatible, and compatible sets (10 runs,  $k = 16$  clusters per domain).

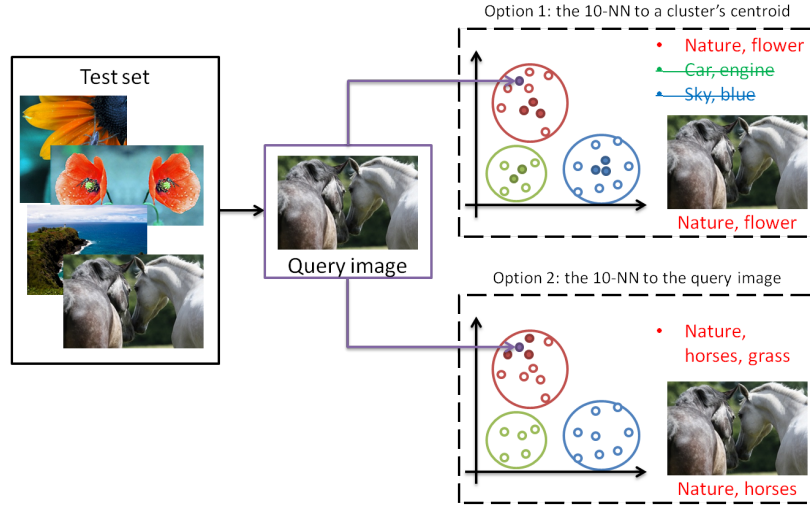
Data type	Text domain		
Algorithm	Mixed set	Incompatible set	Compatible Set
DB Index	<b><math>2.70 \pm 0.18[2.45, 2.74, 3.00]</math></b>	$2.82 \pm 0.15[2.58, 2.80, 3.04]$	$2.93 \pm 0.39[2.45, 2.88, 3.53]$
Silhouette Index	$0.0014 \pm 0.0017[0.001, 0.004, 0.008]$	<b><math>0.0031 \pm 0.001[0.0018, 0.002, 0.008]</math></b>	$0.012 \pm 0.003[0.008, 0.012, 0.017]$
Dunn Index	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>
Tags DB Index	$37.07 \pm 2.93[32.80, 37.74, 41.10]$	$33.8585 \pm 4.92[27.14, 33.33, 44.36]$	<b><math>21.65 \pm 5.52[14.37, 19.87, 31.66]</math></b>
Tags Silhouette Index	$0.11 \pm 0.02[0.08, 0.1051, 0.14]$	<b><math>0.12 \pm 0.02[0.09, .12, 0.14]</math></b>	$0.08 \pm 0.015[0.05, 0.07, 0.12]$
Tags Dunn Index	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>
Data type	Visual domain		
Algorithm	Mixed set	Incompatible set	Compatible Set
DB Index	$6.30 \pm 1.338[3.60, 6.40, 9.05]$	$5.42 \pm 0.97[4.32, 5.1794, 7.0726]$	<b><math>3.56 \pm 0.75[2.75, 3.42, 5.01]</math></b>
Silhouette Index	$0.002 \pm 0.02[0.024, 0.0013, 0.03]$	$0.006 \pm 0.0205[-0.0129, -0.0013, 0.0549]$	<b><math>0.062 \pm 0.022[0.025, 0.057, 0.095]</math></b>
Dunn Index	$0.004 \pm 0.003[0.002, 0.002, 0.012]$	$0.006 \pm 0.013[0.0008, 0.0009, 0.044]$	<b><math>0.012 \pm 0.018[0.0009, 0.004, 0.054]</math></b>
Tags DB Index	$37.07 \pm 2.93[32.80, 37.74, 41.10]$	$33.85 \pm 4.92[27.14, 33.33, 44.36]$	<b><math>21.65 \pm 5.52[14.37, 19.87, 31.66]</math></b>
Tags Silhouette Index	$0.11 \pm 0.02[0.08, 0.11, 0.14]$	<b><math>0.12 \pm 0.02[0.10, 0.12, 0.15]</math></b>	$0.08 \pm 0.02[0.05, 0.07, 0.11]$
Tags Dunn Index	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	$0.0001 \pm 0[0.0001, 0.0001, 0.0001]$	<b><math>0.0001 \pm 0[0.0001, 0.0001, 0.0001]</math></b>

**Table 4.22:** Clustering results of the ensemble voting algorithm for the mixed, incompatible, and compatible sets (10 runs,  $k = 16$  clusters per domain, 2 instances for the text domain, and 3 instances for the text domain).

- Option 1: the 10-NN to a cluster’s centroid. For each cluster, we find the 10 closest images to the cluster’s centroid (in the visual domain) and extract a set of tags associated with each image. We then find and store a set,  $T_r$ , of the top  $f_{max}$  most frequent tags associated with each cluster. Finally, when a query image arrives, we assign the image to the closest cluster and use the associated cluster’s tags to auto-annotate the query image.
- Option 2: the 10-NN to the query image in the same cluster. When a query image arrives, we assign the image to the closest cluster (in the visual domain). Then in that cluster, we find the 10 nearest images to the query image. For each such image, we find the set of its associated tags. We then combine these 10 sets of tags, and store a set,  $T_r$ , of the top  $f_{max}$  most frequent tags to use them to auto-annotate the query image.

Data type	Text domain		
Algorithm	Mixed set	Incompatible set	Compatible Set
DB Index	<b>3.31 ± 0.07</b> [3.24, 3.30, 3.49]	3.43 ± 0.05[3.36, 3.43, 3.50]	3.39 ± 0.36[2.73, 3.47, 3.74]
Silhouette Index	0.004 ± 0.0002[0.0045, 0.0045, 0.0045]	0.004 ± 0.0002[0.004, 0.0045, 0.005]	<b>0.015 ± 0.002</b> [0.012, 0.016, 0.018]
Dunn Index	0.0001 ± 0[0.0001, 0.0001, 0.0001]	0.0001 ± 0[0.0001, 0.0001, 0.0001]	0.0001 ± 0[0.0001, 0.0001, 0.0001]
Tags DB Index	35.48 ± 4.91[27.46, 35.59, 42.69]	41.67 ± 3.54[33.11, 42.64, 45.75]	<b>24.40 ± 7.24</b> [14.49, 24.56, 35.22]
Tags Silhouette Index	0.08 ± 0.013[0.08, 0.08, 0.10]	0.08 ± 0.005[0.08, 0.09, 0.10]	<b>0.10 ± 0.01</b> [0.08, 0.09, 0.12]
Tags Dunn Index	0.0001 ± 0[0.0001, 0.0001, 0.0001]	0.0001 ± 0[0.0001, 0.0001, 0.0001]	<b>0.0001 ± 0</b> [0.0001, 0.0001, 0.0001]
Data type	Visual domain		
Algorithm	Mixed set	Incompatible set	Compatible Set
DB Index	2.07 ± 0.09[1.90, 2.05, 2.28]	<b>1.99 ± 0.04</b> [1.93, 1.99, 2.06]	2.18 ± 0.13[2.04, 2.17, 2.50]
Silhouette Index	0.10 ± 0.005[0.09, 0.10, 0.10]	0.11 ± 0.004[0.105, 0.112, 0.12]	<b>0.11 ± 0.01</b> [0.08, 0.11, 0.12]
Dunn Index	0.035 ± 0.02[0.02, 0.02, 0.07]	<b>0.055 ± 0.03</b> [0.008, 0.062, 0.10]	0.046 ± 0.013[0.015, 0.05, 0.05]
Tags DB Index	35.48 ± 4.91[27.46, 35.59, 42.69]	41.67 ± 3.54[33.11, 42.64, 45.75]	<b>24.40 ± 7.24</b> [14.49, 24.56, 35.22]
Tags Silhouette Index	0.08 ± 0.01[0.07, 0.08, 0.08]	0.09 ± 0.005[0.08, 0.095, 0.10]	<b>0.09 ± 0.01</b> [0.008, 0.086, 0.10]
Tags Dunn Index	0.0001 ± 0[0.0001, 0.0001, 0.0001]	0.0001 ± 0[0.0001, 0.0001, 0.0001]	<b>0.0001 ± 0</b> [0.0001, 0.0001, 0.0001]

**Table 4.23:** Clustering results of multiview clustering for the mixed, incompatible, and compatible sets (10 runs,  $k = 16$  clusters per domain).



**Figure 4.18:** An example illustrating the two cluster-based annotation schemes, with final number of tags,  $f_{max} = 2$ .

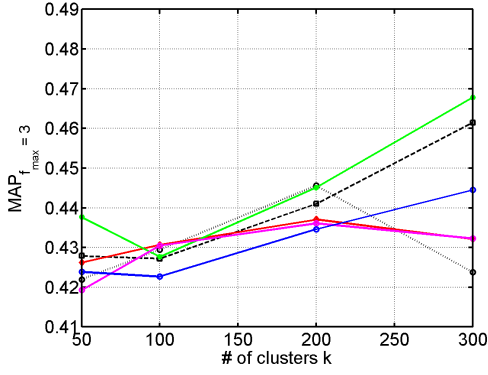
The performance of all clustering methods was evaluated using the average mean precision (MAP), defined in the standard way:

$$MAP_{f_{max}} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{f_{max}} \sum_{f=1}^{f_{max}} \frac{|T_r(f) \cap T_g|}{|T_r(f)|} \quad (4.11)$$

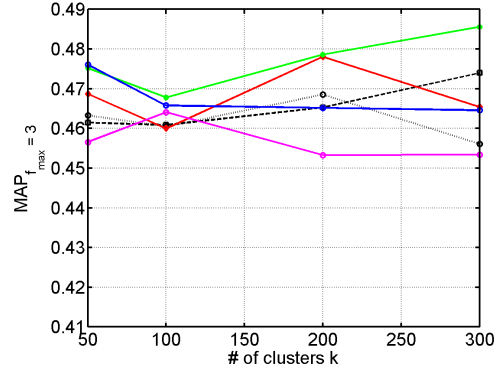
where  $Q$  is the test set of query images,  $T_g$  is the ground-truth set of tags associated with the query image, and  $f_{max}$  takes the discrete values from 1 to 5 tags.

The results for both experimental setups are presented below:

- Table 4.24 shows the  $MAP_{f_{max}=3}$  and  $MAP_{f_{max}=5}$  results for the seed-based IDS (for each seed exchange mechanism), constraint-based IDS, conversion, multiview, ensemble, and splitting clustering algorithms for both validation options for the first experimental setup. For the



(a) Option 1: the 10-NN to a cluster's centroid annotation scheme.



(b) Option 2: the 10-NN to the query image in the same cluster annotation scheme.

**Figure 4.19:** Value of  $MAP_{f_{max}=3}$  for the image auto-annotation of the MIRFlickr data set for the different validation options: seed-based IDS with normal seed exchange (solid red diamonds), constraint-based IDS (solid green stars), conversion clustering (dotted black circles), multiview clustering (solid magenta circles), ensemble clustering (solid blue circles), and splitting clustering (dashed black squares). See the value of  $MAP_{f_{max}=3}$  and  $MAP_{f_{max}=5}$  in Table 4.24.

first validation option, the constraint-based IDS outperforms all other clustering methods for  $k = 50, 200$ , and  $300$  clusters, for both values of MAP, while the seed-based IDS shows a better clustering result with  $k = 100$ . For the second validation option, again, the constraint-based IDS outperforms the other clustering techniques in both values of MAP. Note that the ensemble clustering shows similar results to the constraint-based IDS results for  $MAP_{f_{max}=3}$  but yields in terms of  $MAP_{f_{max}=5}$ . Figures 4.19a and 4.19b show the value of  $MAP_{f_{max}=3}$  with respect to the different number of clusters for the annotation options 1 and 2, respectively.

- Table 4.25 shows  $MAP_{f_{max}=3}$  and  $MAP_{f_{max}=5}$  results for the seed-based IDS (for each seed exchange mechanism), constraint-based IDS, conversion, multiview, ensemble, and splitting clustering algorithms for both validation options with the compatibility analysis. For the first validation option, the seed-based IDS with DB index-based seed exchange mechanism outperforms all other methods with  $k = 50, 80$ , and  $100$  clusters. The constraint-based IDS shows a better clustering result with  $k = 16$ , but yields to the seed-based IDS with normal seed exchange when  $k = 32$ . For the second validation option, the results are less consistent, showing overall improvement of the proposed IDS framework over other clustering methods. Figure 4.20a and 4.20b show the value of the  $MAP_{f_{max}=3}$  with respect to the different number of cluster for the validation option 1 and 2, respectively.

Algorithm	Clustered domain(s)	Number of clusters			
		50	100	200	300
Seed-based IDS with the DB index-based seed exchange mechanism	Text, Visual	0.4127, 0.3907	<b>0.4315</b> , 0.4033	0.4267, 0.4002	0.4351, 0.4069
Seed-based IDS with the XB index-based seed exchange mechanism	Text, Visual	0.4156, 0.3910	0.4260, 0.3987	0.4262, 0.3983	0.4309, 0.4076
Seed-based IDS with normal seed exchange mechanism	Text, Visual	0.4262, 0.3981	0.4307, <b>0.4062</b>	0.4371, 0.4082	0.4321, 0.4014
Constraint-based IDS	Text, Visual	<b>0.4377</b> , <b>0.4029</b>	0.4277, 0.4032	<b>0.4451</b> , <b>0.4154</b>	<b>0.4678</b> , <b>0.4348</b>
Conversion	Text, Visual	0.4220, 0.3963	0.4295, 0.3975	0.4456, 0.4129	0.4238, 0.3969
Multiview k-means	Text, Visual	0.4193, 0.3957	0.4305, 0.4057	0.4361, 0.4054	0.4323, 0.4006
Ensemble voting	Text, Visual	0.4239, 0.3948	0.4227, 0.3983	0.4346, 0.4076	0.4445, 0.4128
Splitting	Visual	0.4279, 0.3996	0.4272, 0.4006	0.4410, 0.4093	0.4615, 0.4278

(a) Option 1: the 10-NN to a cluster's centroid validation scheme.

Algorithm	Clustered domain(s)	Number of clusters			
		50	100	200	300
Seed-based IDS with the DB index-based seed exchange mechanism	Text, Visual	0.4573, 0.4293	0.4562, 0.4287	0.4528, 0.4250	0.4572, 0.4265
Seed-based IDS with the XB index-based seed exchange mechanism	Text, Visual	0.4585, 0.4290	0.4585, 0.4252	0.4711, 0.4371	0.4569, 0.4285
Seed-based IDS with normal seed exchange mechanism	Text, Visual	0.4687, 0.4365	0.4600, 0.4297	0.4781, 0.4407	0.4654, 0.4328
Constraint-based IDS	Text, Visual	<b>0.4761, 0.4437</b>	<b>0.4678, 0.4355</b>	<b>0.4786, 0.4415</b>	<b>0.4856, 0.4549</b>
Conversion	Text, Visual	0.4633, 0.4331	0.4605, 0.4319	0.4686, 0.4368	0.4561, 0.4243
Multiview k-means	Text, Visual	0.4566, 0.4275	0.4641, 0.4317	0.4533, 0.4246	0.4534, 0.4204
Ensemble voting	Text, Visual	<b>0.4761, 0.4417</b>	0.4658, 0.4370	0.4652, 0.4327	0.4646, 0.4326
Splitting	Visual	0.4615, 0.4310	0.4609, 0.4309	0.4653, 0.4340	0.4740, 0.4385

(b) Option 2: the 10-NN to a query image in a same cluster validation scheme.

**Table 4.24:** Value of the  $MAP_{f_{max}=3}$  and  $MAP_{f_{max}=5}$  for the image auto-annotation of the MIRFlickr data set for the different validation schemes.

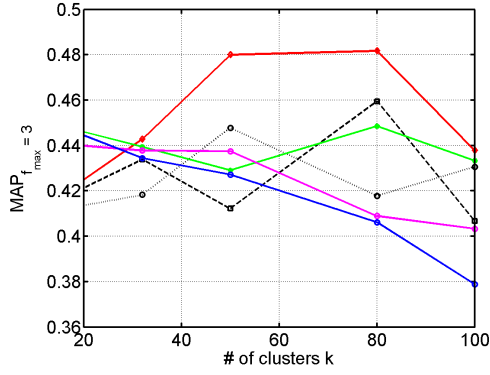
Algorithm	Clustered domain(s)	Training Set	Number of clusters			
			16	32	50	80
Seed-based IDS with the DB index-based seed exchange mechanism	Text, Visual	Compatible	0.4189, 0.3787	0.4428, 0.4009	<b>0.4800, 0.4244</b>	<b>0.4817, 0.4244</b>
Seed-based IDS with the XB index-based seed exchange mechanism	Text, Visual	Compatible	0.4472, 0.3932	0.4272, 0.3895	0.4561, 0.4188	0.4550, 0.4174
Seed-based IDS with normal seed exchange mechanism	Text, Visual	Compatible	0.4239, 0.3845	<b>0.4639, 0.4099</b>	0.4344, 0.3888	0.4606, 0.4175
Constraint-based IDS	Text, Visual	Compatible	<b>0.4482, 0.4089</b>	0.4394, 0.4049	0.4290, 0.3713	0.4485, 0.4058
Conversion	Text, Visual	Mixed	0.4122, 0.3794	0.4183, 0.3856	0.4478, 0.4053	0.4178, 0.3874
Multiview k-means	Text, Visual	Mixed	0.4406, 0.3909	0.4378, 0.4026	0.4372, 0.4007	0.4089, 0.3741
Ensemble voting	Text, Visual	Mixed	0.4478, 0.4080	0.4344, 0.4031	0.4272, 0.3868	0.4061, 0.3758
Splitting	Visual	Mixed	0.4172, 0.3827	0.4339, 0.4031	0.4122, 0.3802	0.4595, 0.4171

(a) Option 1: the 10-NN to a cluster's centroid annotation scheme.

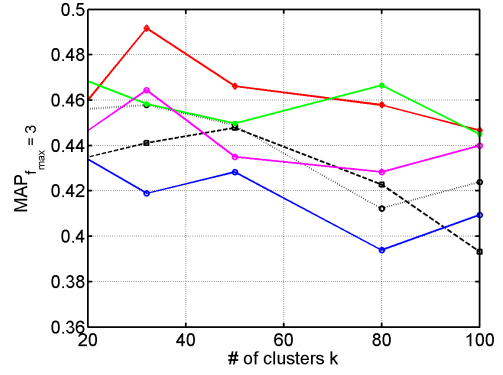
Algorithm	Clustered domain(s)	Training Set	Number of clusters			
			16	32	50	80
Seed-based IDS with the DB index-based seed exchange mechanism	Text, Visual	Compatible	0.4494, 0.4160	<b>0.4917, 0.4342</b>	0.4661, 0.4284	0.4578, 0.4185
Seed-based IDS with the XB index-based seed exchange mechanism	Text, Visual	Compatible	0.4706, <b>0.4296</b>	0.4511, 0.4155	<b>0.4861, 0.4407</b>	0.4539, 0.4252
Seed-based IDS with normal seed exchange mechanism	Text, Visual	Compatible	0.4667, 0.4270	0.4528, 0.4032	0.4833, 0.4390	<b>0.4756, 0.4274</b>
Constraint-based IDS	Text, Visual	Compatible	<b>0.4717, 0.4275</b>	0.4583, 0.4198	0.4497, 0.4197	0.4665, 0.4262
Conversion	Text, Visual	Mixed	0.4556, 0.4174	0.4578, 0.4174	0.4489, 0.4036	0.4122, 0.3819
Multiview k-means	Text, Visual	Mixed	0.4406, 0.4043	0.4644, 0.4200	0.4350, 0.3969	0.4283, 0.3919
Ensemble voting	Text, Visual	Mixed	0.4389, 0.4079	0.4189, 0.3873	0.4283, 0.3903	0.3939, 0.3631
Splitting	Visual	Mixed	0.4328, 0.3986	0.4411, 0.4070	0.4478, 0.4033	0.4228, 0.3932

(b) Option 2: the 10-NN to the query image in the same cluster annotation scheme.

**Table 4.25:** Value of  $MAP_{f_{max}=3}$  and  $MAP_{f_{max}=5}$  for the MIRFlickr data set with the compatible and mixed training sets for the different validation schemes.



(a) Option 1: the 10-NN to a cluster's centroid validation scheme.



(b) Option 2: the 10-NN to a query image in a same cluster validation scheme.

**Figure 4.20:** Value of the  $MAP_{f_{max}=3}$  for the image auto-annotation with the compatibility analysis of the MIRFlickr data set for the different validation options: seed-based IDS with DB index-based seed exchange (solid red diamonds), constraint-based IDS (solid green stars), conversion clustering (dotted black circles), multiview clustering (solid magenta circles), ensemble clustering (solid blue circles), and splitting clustering (dashed black squares). See the value of the  $MAP_{f_{max}=3}$  and  $MAP_{f_{max}=5}$  in Table 4.20.

## 4.5 Summary of the Chapter

In this chapter, we presented experimental results for the three seed exchange mechanisms for the seed-based IDS: DB index-based, XB index-based, and normal seed exchange mechanisms. The proposed seed-based IDS algorithm with the DB index-based seed exchange mechanism shows a better clustering results, algorithm's stability and a faster convergence, than the seed-based IDS with the XB index-based and normal seed exchange mechanisms in terms of the internal and external validity measures. For the proposed constraint-based IDS, we found an optimal number of constraints for each data set, we observe that the constraint-based IDS clustering generally results in an improvement over the splitting algorithm in the following aspects:

- over a wide range of the algorithm parameters,
- for different data sets with different sizes and number of features,
- in different validation measures,
- for some data sets, the improvement are asymmetric, with one domain contributing more to guide the other,
- for some data sets, validation in terms of an external (NMI) and internal (DB) validity measures gives opposite results.

This reflects some disagreement between the internal structure and external labels. Of course, the external validity option is generally impossible without external “true” class labels, which is the case with most real-life data.

Next, we presented experimental results comparing the proposed seed-based and constraint-based IDS clustering approaches with the splitting, conversion, ensemble clustering (with voting and post-clustering as consensus functions), and multiview clustering algorithms. The results are different for the different data sets in the different domains, but overall, we observed that the proposed IDS clustering approaches obtain significantly better clustering results than the other techniques. We observe, how one domain guides the clustering process in another domain, helping it to reach a better clustering. We also noted extremely low minimum values of the DB and high values of the Silhouette indices of the seed-based IDS in both domains in several data sets, indicating the superior potential capabilities of the IDS approach in reaching highly favorable optima.

We also presented experimental results for the proposed compatibility analysis. The experiments on the MIRFlickr data set showed the importance of the compatibility analysis and confirmed the role of mutual supervision in inter-domain clustering for data with mixed domains. Our results for the image auto-annotation experiments show that the proposed IDS clustering approaches outperform other clustering techniques, taking advantage of the inter-domain mutual supervision, domain compatibility, and algorithm stability over a wide range of the different number of clusters.

In the following chapter, we conclude our work and present potential future research.

## CHAPTER 5

### CONCLUSIONS AND FUTURE WORK

#### 5.1 Summary

We proposed an inter-domain supervision (IDS) clustering framework to handle diverse data formats, mixed-type attributes and different sources of data. This approach can be used for combining diverse representations of the data, in particular where data comes from different sources, some of which may be unreliable or uncertain. Our contributions can be summarized as follows:

- We proposed a seed-based inter-domain supervision clustering approach to transfer knowledge discovered from the clustering in one domain to help guide the clustering in the other domains (Section 3.1).
- We proposed different seed exchange mechanisms for the seed-based IDS, in order to control the selectivity of the exchanged knowledge, based on linear-complexity unsupervised internal cluster validity indices (Section 3.1.2).
- We proposed a constraint-based inter-domain supervision clustering approach to handle inconsistent partitions between different domains, which can now be combined into a consistent clustering result (Section 3.2).
- We proposed a domain compatibility analysis approach for a more effective clustering of heterogeneous data, that exploits the synergy between the different domains, even when parts of the data descriptions are incompatible in the different domains (Section 3.4).

The results of our experiments show that the proposed IDS-based heterogeneous data clustering framework tends to yield better clustering results in both domains, over a wide range of parameters. Thus the seeds or constraints obtained from clustering one domain tend to provide additional helpful knowledge to another domain. This information may in turn be used to avoid local minima and obtain a better clustering in the target domain. Moreover, by first distinguishing between the data depending on whether the different domains describe the data in a compatible manner, the IDS

approach was able to compute an even better clustering compared to the conventional methods. Finally, we presented a real life application of our IDS clustering approach to the automated image annotation problem and presented evaluation results on a benchmark data set, consisting of images described with their visual content along with noisy text descriptions, generated by users on the social media sharing website, Flickr.

## 5.2 Current Status and Future Prospects

Future work can expand this work to address some current limitations, by further:

- exploring the effect of parametrized distortion measures that can be incorporated within the proposed constraint-based IDS clustering framework for heterogeneous data.
- devising a better method to estimate the confidence levels of the points contributing to the created constraints, and then using them to obtain better informed constraint violation cost weights  $W$  and  $\bar{W}$  in the HMRF K-means penalty terms.
- better handling of the complexity of the HMRF-initialization, currently based on transitive closure. This could be handled by sampling, thus avoiding full transitive closure. Other approaches could be investigated.

## REFERENCES

- [Abhishek and Hal, 2011] Abhishek, K. and Hal, D. I. (2011). A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th International Conference on Machine Learning*, pages 393–400.
- [Abhishek et al., 2011] Abhishek, K., Piyush, R., and Hal, D. I. (2011). Co-regularized multi-view spectral clustering. In *Proceedings of the 25th Neural Information Processing Systems*, pages 1412–1421.
- [Ahmad and Dey, 2007] Ahmad, A. and Dey, L. (2007). A k-mean clustering algorithm for mixed numeric and categorical data. *Data and Knowledge Engineering*, 63(2):503 – 527.
- [Banerjee et al., 2005] Banerjee, A., Dhillon, I. S., Ghosh, J., Sra, S., and Ridgeway, G. (2005). Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6:2005.
- [Basu et al., 2002a] Basu, S., Banerjee, A., and Mooney, R. (2002a). Semi-supervised clustering by seeding. In *Proceedings of 19th International Conference on Machine Learning (ICML-2002)*.
- [Basu et al., 2002b] Basu, S., Banerjee, A., and Mooney, R. J. (2002b). Semi-supervised clustering by seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning, ICML '02*, pages 27–34. Morgan Kaufmann Publishers Inc.
- [Basu et al., 2004] Basu, S., Bilenko, M., and Mooney, R. J. (2004). A probabilistic framework for semi-supervised clustering. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '04*, pages 59–68.
- [Ben-Hur et al., 2001] Ben-Hur, A., Horn, D., Siegelmann, H. T., and Vapnik, V. (2001). A support vector method for clustering. In *Advances in Neural Information Processing Systems 13*, pages 367–373. MIT Press.
- [Ben-Hur et al., 2002] Ben-Hur, A., Horn, D., Siegelmann, H. T., and Vapnik, V. (2002). Support vector clustering. *J. Mach. Learn. Res.*, 2:125–137.

- [Bezdek, 1981] Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.
- [Bezdek et al., 1984] Bezdek, J. C., Ehrlich, R., and Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers and Geosciences*, 10(2–3):191–203.
- [Bickel and Scheffer, 2004] Bickel, S. and Scheffer, T. (2004). Multi-view clustering. In *Proceedings of the Fourth IEEE International Conference on Data Mining, ICDM '04*, pages 19–26.
- [Blum and Mitchell, 1998] Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory, COLT' 98*, pages 92–100, New York, NY, USA. ACM.
- [Boley et al., 1999] Boley, D., Gini, M., Gross, R., Han, E.-H. S., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., and Moore, J. (1999). Partitioning-based clustering for web document categorization. *Decision Support Systems*, 27(3):329 – 341.
- [Caicedo et al., 2012] Caicedo, J. C., BenAbdallah, J., Gonzalez, F. A., and Nasraoui, O. (2012). Multimodal representation, indexing, automated annotation and retrieval of image collections via non-negative matrix factorization. *Neurocomputing*, 76(1):50 – 60.
- [Cohn et al., 2003] Cohn, D., Caruana, R., and McCallum, A. (2003). Semi-supervised clustering with user feedback. Technical report.
- [Cover and Hart, 1967] Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27.
- [Cover, 1965] Cover, T. M. (1965). Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *Electronic Computers, IEEE Transactions on*, EC-14:326–334.
- [Davies and Bouldin, 1979] Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-1(2):224 –227.
- [de Sa, 2005] de Sa, V. (2005). Spectral clustering with two views. In *Proceedings of the ICML 2005 Workshop on Learning with Multiple Views, ICML '05*, pages 20–27, Bonn, Germany.
- [Dhillon and Modha, 2001] Dhillon, I. S. and Modha, D. S. (2001). Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42:143–175.

- [Dimitriadou et al., 2001] Dimitriadou, E., Weingessel, A., and Hornik, K. (2001). Voting-merging: An ensemble method for clustering. In Dorffner, G., Bischof, H., and Hornik, K., editors, *Artificial Neural Networks - ICANN 2001*, volume 2130 of *Lecture Notes in Computer Science*, pages 217–224. Springer Berlin / Heidelberg.
- [Domeniconi and Al-Razgan, 2009] Domeniconi, C. and Al-Razgan, M. (2009). Weighted cluster ensembles: Methods and analysis. *ACM Trans. Knowl. Discov. Data*, 2(4):17:1–17:40.
- [Dudoit and Fridlyand, 2003] Dudoit, S. and Fridlyand, J. (2003). Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9):1090–1099.
- [Dunn, 1973] Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.
- [Dunn, 1974] Dunn, J. C. (1974). Well separated clusters and optimal fuzzy partitions. *J. Cybern.*, 4:95–104.
- [Equitz, 1989] Equitz, W. (1989). A new vector quantization clustering algorithm. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(10):1568 –1575.
- [Espinoza et al., 2005] Espinoza, M., Joye, C., Belmans, R., and DeMoor, B. (2005). Short-term load forecasting, profile identification, and customer segmentation: A methodology based on periodic time series. *Power Systems, IEEE Transactions on*, 20(3):1622 – 1630.
- [Ester et al., 1996] Ester, M., peter Kriegel, H., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231. AAAI Press.
- [Fern and Lin, 2008] Fern, X. Z. and Lin, W. (2008). Cluster ensemble selection. *Statistical Analysis and Data Mining*, 1(3):128–141.
- [Filippone et al., 2008] Filippone, M., Camastra, F., Masulli, F., and Rovetta, S. (2008). A survey of kernel and spectral methods for clustering. *Pattern Recognition*, 41:176 – 190.
- [Fisher, 1987] Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2):139–172.
- [Frakes and Baeza-Yates, 1992] Frakes, W. B. and Baeza-Yates, R. (1992). *Information Retrieval: Data Structures and Algorithms*. Prentice Hall PTR.

- [Frank, 2005] Frank, A. (2005). On kuhn’s hungarian method - a tribute from hungary. *Naval Research Logistics (NRL)*, 52(1):2–5.
- [Frank and Asuncion, 2010] Frank, A. and Asuncion, A. (2010). UCI machine learning repository.
- [Frigui et al., 2007] Frigui, H., Hwang, C., and Rhee, F. C.-H. (2007). Clustering and aggregation of relational data with applications to image database categorization. *Pattern Recognition*, 40(11):3053 – 3068.
- [Gal and Cohen-Or, 2006] Gal, R. and Cohen-Or, D. (2006). Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.*, 25(1):130–150.
- [Gan et al., 2007] Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics, Philadelphia, PA.
- [Ganti et al., 1999] Ganti, V., Gehrke, J., and Ramakrishnan, R. (1999). Cactus - clustering categorical data using summaries. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD ’99, pages 73–83, New York, NY, USA. ACM.
- [Girolami, 2002] Girolami, M. (2002). Mercer kernel-based clustering in feature space. *Neural Networks, IEEE Transactions on*, 13(3):780 –784.
- [Guha et al., 1998] Guha, S., Rastogi, R., and Shim, K. (1998). Cure: an efficient clustering algorithm for large databases. *SIGMOD Rec.*, 27(2):73–84.
- [Guha et al., 2000] Guha, S., Rastogi, R., and Shim, K. (2000). Rock: A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5):345 – 366.
- [Halkidi et al., 2002] Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2002). Cluster validity methods: Part i. *SIGMOD Rec.*, 31(2):40–45.
- [Hammuda and Kamel, 2006] Hammuda, K. and Kamel, M. (2006). Collaborative document clustering. In *In Proceedings of the Sixth SIAM International Conference on Data Mining (SDM06)*, pages 453–463, Bethesda, MD.
- [He et al., 2005] He, Z., Xu, X., and Deng, S. (2005). Clustering Mixed Numeric and Categorical Data: A Cluster Ensemble Approach. *eprint arXiv:cs/0509011*.
- [Hore et al., 2009] Hore, P., Hall, L. O., and Goldgof, D. B. (2009). A scalable framework for cluster ensembles. *Pattern Recognition*, 42(5):676 – 688.

- [Hsu and Chen, 2007] Hsu, C.-C. and Chen, Y.-C. (2007). Mining of mixed data with application to catalog marketing. *Expert Systems with Applications*, 32(1):12 – 23.
- [Huang, 1998a] Huang, Z. (1998a). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2:283–304.
- [Huang, 1998b] Huang, Z. (1998b). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2:283–304.
- [Huiskes and Lew, 2008] Huiskes, M. J. and Lew, M. S. (2008). The mir flickr retrieval evaluation. In *MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval*, New York, NY, USA. ACM.
- [Inokuchi and Miyamoto, 2004] Inokuchi, R. and Miyamoto, S. (2004). Lqv clustering and som using a kernel function. In *Fuzzy Systems, 2004. Proceedings. 2004 IEEE International Conference on*, volume 3, pages 1497 – 1500 vol.3.
- [Jain and Dubes, 1988] Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- [Johnson, 1967] Johnson, S. (1967). Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254.
- [Karypis and Kumar, 1998] Karypis, G. and Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392.
- [Kaufman and Rousseeuw, 1990] Kaufman, L. and Rousseeuw, P. (1990). *Finding Groups in Data An Introduction to Cluster Analysis*. John Wiley & Sons, Inc, New York.
- [Kershaw, 1978] Kershaw, D. S. (1978). The incomplete cholesky - conjugate gradient method for the iterative solution of systems of linear equations. *Journal of Computational Physics*, 26(1):43 – 65.
- [Krishnapuram and Keller, 1993] Krishnapuram, R. and Keller, J. (1993). A possibilistic approach to clustering. *Fuzzy Systems, IEEE Transactions on*, 1(2):98–110.
- [Kuhn, 1955] Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97.
- [Lakhina et al., 2005] Lakhina, A., Crovella, M., and Diot, C. (2005). Mining anomalies using traffic feature distributions. *SIGCOMM Comput. Commun. Rev.*, 35(4):217–228.

- [MacDonald and Fyfe, 2000] MacDonald, D. and Fyfe, C. (2000). The kernel self-organising map. In *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on*, volume 1, pages 317–320 vol.1.
- [MacQueen, 1967] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.
- [Manning et al., 2008] Manning, C. D., Raghavan, P., and Schtze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- [Monay and Gatica-Perez, 2007] Monay, F. and Gatica-Perez, D. (2007). Modeling semantic aspects for cross-media image indexing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(10):1802–1817.
- [Muller et al., 2001] Muller, K.-R., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *Neural Networks, IEEE Transactions on*, 12(2):181–201.
- [Nasraoui and Frigui, 2000] Nasraoui, O. and Frigui, H. (2000). Extracting web user profiles using relational competitive fuzzy clustering.
- [Nasraoui and Krishnapuram, 1996] Nasraoui, O. and Krishnapuram, R. (1996). A robust estimator based on density and scale optimization and its application to clustering. In *Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on*, volume 2, pages 1031–1035 vol.2.
- [Nasraoui and Krishnapuram, 2002] Nasraoui, O. and Krishnapuram, R. (2002). One step evolutionary mining of context sensitive associations and web navigation patterns. In *in SIAM conference on Data Mining*, pages 531–547.
- [Nasraoui et al., 1999] Nasraoui, O., Krishnapuram, R., and Joshi, A. (1999). Relational clustering based on a new robust estimator with application to web mining. In *Fuzzy Information Processing Society, 1999. NAFIPS. 18th International Conference of the North American*, pages 705–709.
- [Ng et al., 2001] Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press.

- [Pedrycz and Rai, 2008] Pedrycz, W. and Rai, P. (2008). Collaborative clustering with the use of fuzzy c-means and its quantification. *Fuzzy Sets and Systems*, 159(18):2399 – 2427.
- [Pedrycz and Rai, 2009] Pedrycz, W. and Rai, P. (2009). A multifaceted perspective at data analysis: a study in collaborative intelligent agents. *Trans. Sys. Man Cyber. Part B*, 39(4):834–844.
- [Plant and Böhm, 2011] Plant, C. and Böhm, C. (2011). Inconco: interpretable clustering of numerical and categorical objects. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 1127–1135, New York, NY, USA. ACM.
- [Qin and Suganthan, 2004] Qin, A. K. and Suganthan, P. N. (2004). Kernel neural gas algorithms with application to cluster analysis. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4 - Volume 04*, ICPR '04, pages 617–620. IEEE Computer Society.
- [Remm et al., 2001] Remm, M., Storm, C. E., and Sonnhammer, E. L. (2001). Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *Journal of Molecular Biology*, 314(5):1041 – 1052.
- [Rissanen, 1978] Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5):465 – 471.
- [Rousseeuw, 1987] Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20:53–65.
- [Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423.
- [Sheath and Sokal, 1973] Sheath, P. H. A. and Sokal, R. C. (1973). *Numerical taxonomy. The principles and practice of numerical classification*. San Francisco : W.H. Freeman.
- [Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.
- [Steinbach et al., 2000] Steinbach, M., Karypis, G., and Kumar, V. (2000). A comparison of document clustering techniques. In *In KDD Workshop on Text Mining*.
- [Strehl and Ghosh, 2003] Strehl, A. and Ghosh, J. (2003). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617.

- [Strehl et al., 2002] Strehl, A., Ghosh, J., and Cardie, C. (2002). Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617.
- [Tan et al., 2005] Tan, P.-N., Steinbach, M., and Kumar, V. (2005). *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [Tang et al., 2009] Tang, W., Lu, Z., and Dhillon, I. S. (2009). Clustering with multiple graphs. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, pages 1016–1021, Washington, DC, USA. IEEE Computer Society.
- [Topchy et al., 2005] Topchy, A., Jain, A., and Punch, W. (2005). Clustering ensembles: models of consensus and weak partitions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1866 –1881.
- [Topchy et al., 2004a] Topchy, A., Jain, A. K., and Punch, W. (2004a). A Mixture Model for Clustering Ensembles. In *Proceedings of the SIAM International Conference on Data Mining*.
- [Topchy et al., 2004b] Topchy, A., Minaei-Bidgoli, B., Jain, A., and Punch, W. (2004b). Adaptive clustering ensembles. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 1, pages 272 – 275 Vol.1.
- [Ungar et al., 1998] Ungar, L., Foster, D., Andre, E., Wars, S., Wars, F. S., Wars, D. S., and Whispers, J. H. (1998). Clustering methods for collaborative filtering. AAAI Press.
- [Wagstaff et al., 2001] Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. (2001). Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, pages 577–584. Morgan Kaufmann Publishers Inc.
- [Wang et al., 1999] Wang, K., Xu, C., and Liu, B. (1999). Clustering transactions using large items. In *Proceedings of the eighth international conference on Information and knowledge management, CIKM '99*, pages 483–490, New York, NY, USA. ACM.
- [Xie and Beni, 1991] Xie, X. L. and Beni, G. (1991). A validity measure for fuzzy clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(8):841–847.
- [Xing et al., 2002] Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. (2002). Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press.

- [Yang et al., 2002] Yang, Y., Guan, X., and You, J. (2002). Clope: a fast and effective clustering algorithm for transactional data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 682–687, New York, NY, USA. ACM.
- [Zhang and Chen, ] Zhang, D. and Chen, S. Fuzzy clustering using kernel method. In *in The 2002 International Conference on Control and Automation, 2002. ICCA, 2002*, pages 162–163.
- [Zhong and Ghosh, 2003] Zhong, S. and Ghosh, J. (2003). A unified framework for model-based clustering. *Journal of Machine Learning Research*, 4:1001–1037.
- [Zhou and Burges, 2007] Zhou, D. and Burges, C. J. C. (2007). Spectral clustering and transductive learning with multiple views. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 1159–1166. ACM.

## CURRICULUM VITAE

NAME	Artur Abdullin
E-MAIL	ar.abdullin@gmail.com
EDUCATION	MS, Computer Engineering and Computer Science University of Louisville, USA 1/2008 - 5/2009 BS and MS, Physics Perm State University, Russia 9/2002 - 6/2007
WORK EXPERIENCE	Data Scientist Resonate Networks, Inc 5/2013 - Present Research Assistant University of Louisville, Knowledge Discovery and Web Mining Lab 1/2010 - 5/2013 Research Assistant University of Louisville, Computational Intelligence Laboratory 1/2008 - 12/2009
AWARDS, MEMBERSHIPS AND SERVICE	Winner of the Graduate Dean's Citation Award at UofL, 2013 Winner of the Computer Science and Engineering Doctoral Award at UofL, 2013 Winner of the Third Annual Graduate Research Symposium at UofL, 2011 Winner of the 2011 Engineering Expo Student Research Competition at UofL. Winner of the Research Tuition Award 2010. Winner of the Hearst Analytics Challenge 2010, out of 700 teams from nearly 60 countries. Cash prize \$25,000.
SELECTED PUBLICATIONS AND RESEARCH PROJECTS	1. Jacek M. Zurada, Maciej A. Mazurowski, Artur Abdullin, Rammohan Ragade, Janusz Wojtusiak, James E. Gentle, Building Virtual Community in Computational Intelligence and Machine Learning, Computational Intelligence Magazine, 4, 1, 43-54. 2. Abdullin, A., Nasraoui, O.: "Clustering heterogeneous data with mutual semi-supervision," in Proceedings of SPIRE 2012 - International Symposium on String Processing and Information Retrieval, October 2012. 3. Abdullin, A., Nasraoui, O.: "A semi-supervised learning framework to cluster mixed data types". In: Proceedings of KDIR 2012 - International Conference on Knowledge Discovery and Information Retrieval, October 2012 4. A. Abdullin and O. Nasraoui, "Clustering Heterogeneous Data Sets", Web Congress (LA-WEB), 2012 Eighth Latin American, 2012