Deep Multi-view Clustering Using Local Similarity Graphs

Shuli Jiang

CMU-CS-20-115
May 2020

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Artur Dubrawski (Chair)
Jeff Schneider

Submitted in partial fulfillment of the requirements for the degree of Master of Science.

Keywords: data mining, unsupervised learning, multi-view clustering, canonical correlation analysis, local similarity graphs, mutual K nearest neighbors, deep autoencoders

For my beloved child,
Chester Virgil Jiang.
I wish there is no panleukopenia in Heaven.

Abstract

Multi-view clustering involves clustering data with different, possibly distinct feature sets simultaneously. In many application domains, multi-view data arises naturally. For example, news articles can be described by both text and pictures, and multimedia segments can be described by their video signals from cameras and audio signals from voice recorders. Multi-view clustering has a wide range of potentially high impact applications. Yet, the benefits of using graph-based local similarity information to learn better representations of data for clustering, and the flexibility of incorporating pairwise constraints which may be accessible to improve clustering performance, are still under-explored in multi-view clustering.

In this thesis, we present Local Similarity Graph based Multi-view Clustering (LSGMC), a new and improved correlation-based multi-view clustering approach. The method leverages local similarity graphs constructed by mutual K nearest neighbors. LSGMC uses the graphs to guide the search for a better data representation through exploring first order proximity within views, and utilizing complementary information across views. We empirically show that LSGMC can efficiently use information from multiple views to improve clustering accuracy, and outperform state-of-the-art multi-view alternatives on a variety of benchmark and real world datasets, including image data for hand digit recognition, text data for language recognition and acoustic-articulatory data for speech recognition. We further show that LSGMC is flexible in incorporating pairwise constraints and thus it can be naturally extended to handle semi-supervised learning problems.

Acknowledgments

First, I would like to express my deepest gratitude to my advisor Professor Artur Dubrawski, for providing me with the wonderful opportunity to work at the Auton Lab, for opening me to the amazing world of research and for helping me grow as a researcher. I would also like to express my gratitude to Benedikt Boecking, a current CMU Robotics Ph.D. student and a former full-time staff researcher at the Auton Lab, for all the support and advice he gave me on my research throughout the year. Everything during my fifth-year master journey will not be made possible without Professor Dubrawski and Ben Boecking. I am so lucky to have the chance to work with both of them and I really had a wonderful time at the Auton Lab.

I would like to thank a group of wonderful people at the Auton Lab who helped me with my research and who gave me life advice, including Professor Barnabás Póczos, Jieshi (Jessie) Chen, Anthony Wertz, Dr. Eric Lei, Sibi Venkatesan, Chirag Nagpal, Xinyu (Rachel) Li, Nick Gisolfi, Vincent Jeanselme, Jiaxian (Chris) Sheng, Dr. Predrag Punosevac, and many others.

I would like to thank Professor Jeff Schneider at the Auton Lab for being my audience as my second thesis committee member. I am grateful to our program coordinator Tracy Farbacher who helped keep everything organized and coordinated with the oral presentation.

I would like to thank all my friends at CMU and my parents for their constant love and support.

Finally, I would like to give special thanks to my advisor Professor Dubrawski, Benedikt Boecking and Professor David P. Woodruff, for not only teaching me essential skills as a researcher but also for greatly influencing me to continue pursuing a career in research. This thesis marks the end of my student life at CMU, and the start of another wonderful journey as a graduate student researcher.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we first provide a general background on multi-view data and multi-view clustering. We argue the significance of multi-view clustering approaches. We state our motivation for developing a new multi-view clustering approach and summarize our contributions. At the end of this chapter, we provide a roadmap of this thesis.

1.1 Motivation

Multi-view clustering involves clustering data with different, possibly distinct feature sets simultaneously. In many application domains such multi-view data arises naturally. For example, the news can be described by both text and pictures [27], multimedia segments can be described by their video signals from cameras and audio signals from voice recorders [10], a person can be identified by his/her face, fingerprints, signature, or iris, with information obtained from multiple sources [36]. It has further been observed that even artificially splitting features to create multi-view data can improve the performance under multi-view learning compared to single view learning [24].

One might argue that a reasonable and simple way of performing multi-view clustering is to concatenate features from all views in order to convert multi-view clustering into a more familiar single-view setting. However, such concatenation may exacerbate the risk of over-fitting, especially with small training datasets, and it may diminish the interpretation of the resulting models since each view often has specific properties [36]. Thus, it is generally preferable to consider methods that can efficiently leverage information from multiple views.

Our motivation for developing a new multi-view clustering approach comes from the following gaps between single-view clustering and multi-view clustering:

1. While the use of local similarity graphs–often constructed by K nearest neighbors (KNN) or mutual K nearest neighbors (MKNN) –has been widely explored and shown to be effective in improving performance in single-view clustering, e.g. [2, 11, 20, 28, 29], its use and effect remain under-explored in multi-view clustering.

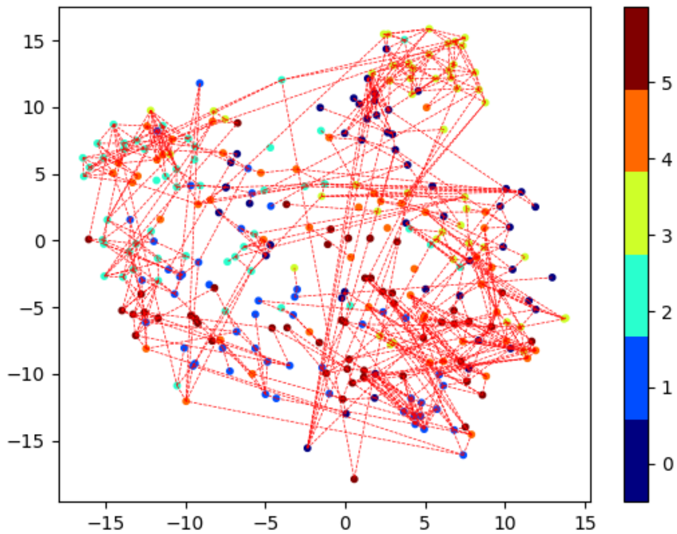


Figure 1.1: An example illustrating the usefulness of MKNN graphs: t-SNE plot on the first two components of 100 samples from a real world dataset with 6 classes. Each colored point indicates a sample from a class indicated by its color as shown by the color bar on the right. Each dotted red line represents an edge in the mutual K nearest neighbor graph (K=10) constructed on the data samples using cosine similarity measure.

To see how MKNN graphs might help clustering tasks, we consider an example in Figure 1.1, where each point represents the first two components in the t-SNE embedding of 100 samples from a real world dataset with 6 classes and each dotted red line represents edges in the MKNN graph constructed on the 100 samples using cosine similarity measure. We observe that the MKNN graph contains clique-like structures which could reveal true information about data clusters, even though such information might be noisy. For example, the edges at the upper right corner form a dense region over the green colored points, most of which come from class 3.

2. Sometimes we may have access to *must-link* and *cannot-link* pairs on certain datasets, which could also reveal true information about the clusters. Such constraints have been shown to be helpful in improving single-view clustering performance [6, 25], where the setting is often referred to as semi-supervised clustering or constrained clustering. Existing single-view clustering algorithms that are guided by a local similarity graph have been shown to adapt well to the semi-supervised clustering setting where the graph is augmented with known pairwise constraints [11, 29]. We would like to develop a new multi-view clustering approach that can also easily incorporate such constraints and improve the clustering performance when such constraints are available.

1.2 Contributions

In this thesis, we present Local Similarity Graph based Multi-view Clustering (LSGMC), an improved correlation based multi-view clustering approach. LSGMC learns an improved data representation in a lower dimensional embedding space through nonlinear maps. To guide this search for a better data representation, LSGMC draws on the following ideas:

1. The ability to reconstruct samples from the low dimensional embedding.
2. Correlation among data across views.
3. First order proximity which preserves the local structure of relationships among samples within views.
4. Complementary information across views through a unified similarity graph which is based on similarity graphs observed in the individual views.

LSGMC is able to naturally adapt to the semi-supervised setting in which we have prior knowledge about pairs of data elements that should belong to the same or different clusters. Our experiments demonstrate that the proposed approach outperforms state-of-the-art multi-view clustering approaches, including canonical correlation analysis (CCA) based deep clustering.

Our main contributions are as follows:

1. We explore the usage of local similarity graphs in multi-view clustering, which is under-explored in current literature.
2. We present a new multi-view clustering approach we term LSGMC.
3. We show that LSGMC is able to outperform state-of-art multi-view clustering alternatives on datasets of various types.
4. We further show the flexibility of LSGMC in incorporating pairwise constraints.

1.3 Thesis Organization

The rest of the thesis is organized as follows:

- In Chapter 2, we survey related work on non-centroid based clustering, multi-view clustering and the usage of local similarity graphs in single-view clustering. We further present the connection between related work and our proposed approach.
- In Chapter 3, we first formally define the multi-view clustering problem. We introduce one popular technique from multi-view clustering, Canonical Correlation Analysis (CCA) and one non-centroid based clustering method, Robust Continuous Clustering (RCC), as well as their deep extensions. We further discuss how we combine those ideas into a new, improved multi-view clustering approach.

- In Chapter 4, we describe in detail each component in our proposed approach LSGMC and motivations for designing the objective functions.
- In Chapter 5, we introduce our experiment settings for evaluating LSGMC. We introduce the datasets used in the experiments and state-of-the-art multi-view clustering alternatives we used as benchmarks for comparison. We further include implementation details and describe the evaluation metrics.
- In Chapter 6, we report results from all the experiments. We visualize the MKNN graphs and the learned data embeddings by all multi-view clustering methods for comparison on two chosen datasets. We discuss and analyze our results. We further include a set of experiments in semi-supervised setting showing the flexibility of LSGMC in incorporating pairwise constraints.
- In Chapter 7, we conclude the thesis work, discuss implications and limitations of LSGMC and motivate future work.

Chapter 2

Related Work

In this chapter, we survey related work on non-centroid based clustering, multi-view clustering and the usage of local similarity graphs in single-view clustering. We further show how related work connects to our work.

2.1 Non-centroid Based Clustering

K means clustering is one of the most widely applied clustering methods. Figure 2.1 shows an example of K means clustering on 9 synthetic data points from two clusters. During training, K means clustering maintains a set of center points, which is shown as the star points in the example. K means alternates between assigning data points to the closest center and updating the center points.

The clustering framework of the proposed algorithm is related to continuous, non-centroid based clustering approaches, proposed in works for single-view clustering such as [11, 29]. Unlike K Means clustering, there is no center points in non-centroid based clustering methods. Instead of learning a set of cluster centers, non-centroid based clustering methods maintain representatives for each data sample and learn to collapse those representations into clusters under the guidance of local similarity graphs. Figure 2.2 shows an example of such clustering method again on 9 synthetic data points. These clustering techniques enjoy several benefits, including the flexibility of incorporating pairwise constraints, reduced sensitivity to initialization and effectiveness in high dimensions [29]. However, the non-centroid clustering framework and the use of local similarity graphs are under-explored in multi-view setting. LSGMC draws on these ideas in a multi-view setting.

2.2 Multi-view Clustering Principles

In multi-view clustering, two general concepts are exploited: the *consensus* and *complementary* principles [36]. Within the consensus principle, the goal is to maximize agreement among

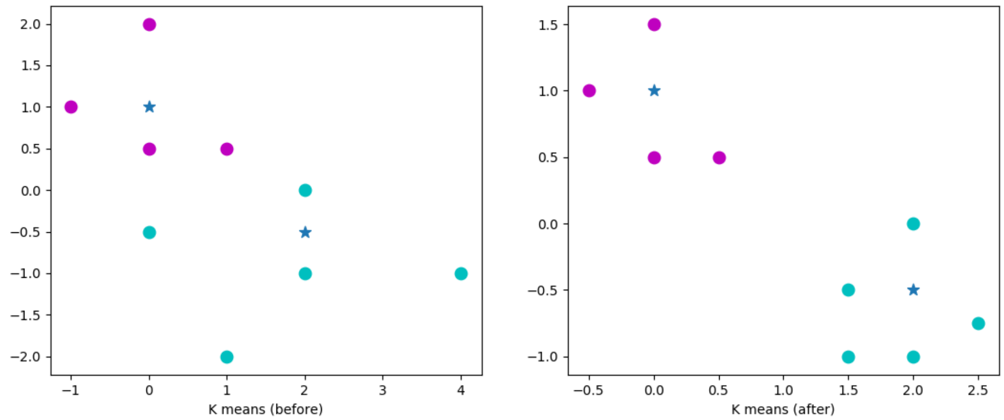


Figure 2.1: An example of K means clustering. The magenta points represent data samples from one class and the cyan points represent data samples from another class. The red edges represent edges in local similarity graphs. The stars represent center points for each cluster. The left figure represents samples before training and the right figure represents samples after training.

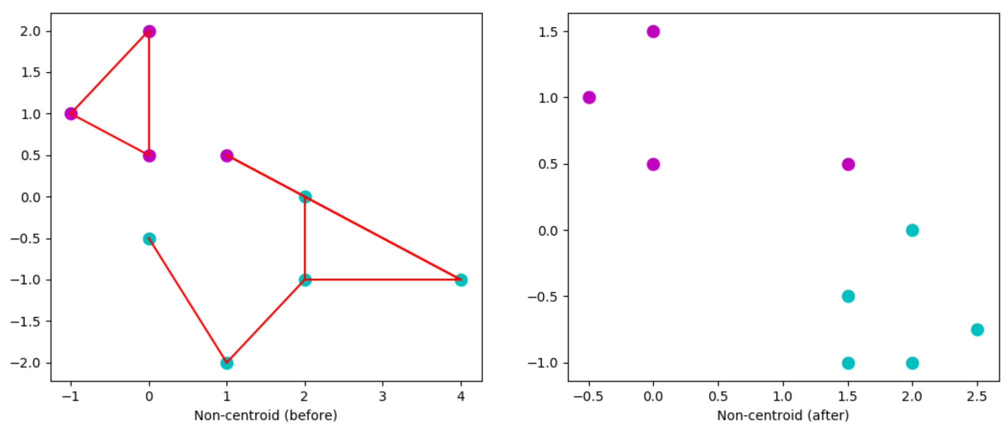


Figure 2.2: An example of non-centroid based clustering. The magenta points represent data samples from one class and the cyan points represent data samples from another class. The red edges represent edges in the local similarity graph constructed by mutual K nearest neighbors (K=3) using cosine similarity measure. The left figure represents samples before training and the right figure represents samples after training.

multiple views since they are obtained by simultaneously observing the same object. The complementary principle states that each view of the data may contain some knowledge that other views do not have [36]. [10] further divide multi-view clustering algorithms into generative approaches and discriminative approaches. According to this survey, discriminative approaches usually perform better than generative approaches in multi-view clustering [10]. The proposed LSGMC is an unsupervised discriminative approach for multi-view clustering and explores both the *consensus* and the *complementary* principle.

2.3 Popular Multi-view Clustering Methods

A popular strategy for multi-view clustering is to first project various feature spaces into similar lower dimensional embedding spaces (data representation) and then regularizes those embeddings or learns a unified embedding while keeping the consistency between views maximized. After that, we can apply basic unsupervised clustering algorithms, such as K means and spectral clustering, to the learned data representation to obtain a final cluster assignment. There are various approaches for maximizing agreement between views, including co-regularization and co-training based methods [16, 17, 34], matrix factorization based methods [19, 40, 41] and subspace clustering based methods under the general assumption that data from multiple views are generated from the same latent space [9, 12, 21, 37, 38]. Deep learning has also been explored in multi-view clustering. These deep methods usually learn a data representation through autoencoders or convolutional neural networks, which are able to extract nonlinear features from the original data space. Deep clustering approaches have been shown to outperform traditional clustering methods [1, 5, 13, 30, 31, 39].

2.4 Local Similarity Graphs

Existing work on single-view clustering has shown that graph based local similarity can be useful to improve clustering performance [2, 11, 20, 28, 29]. For example, the work in [29] and [11] demonstrates that a connectivity matrix built via mutual K nearest neighbors (MKNN) can bear a useful training signal and that clustering algorithms can overcome the noise contained in such a matrix.

Further, it has been shown that local similarity graphs constructed via MKNN perform better than K nearest neighbors (KNN) on a variety of data mining tasks. Consider two data points a and b, in KNN graphs, an edge is added to the graph if a is one of the K nearest neighbors to b or b is one of the K nearest neighbors to a. The neighborhood is determined based on similarity measures between a and b, e.g. cosine similarity or Euclidean distance. But in MKNN graphs, an edge is added if and only if a and b are both one of the K nearest neighbors to each other. To see the difference between KNN and MKNN graphs, we show an example in Figure 2.3, where the left plot shows a graph constructed by KNN and the right plot shows a graph constructed by MKNN. Consider two classes of data samples represented as magenta and cyan points in the plot. The blue edges indicate correct edges whose incident nodes belong to the same class. Such edges can provide correct information for clustering during training. The red edges indicate the wrong edge whose incident nodes belong to different classes. Such edges will provide noisy information during training. We can see that MKNN is more conservative in adding edges, potentially reduces the number of incorrect edges and could be more robust to outliers.

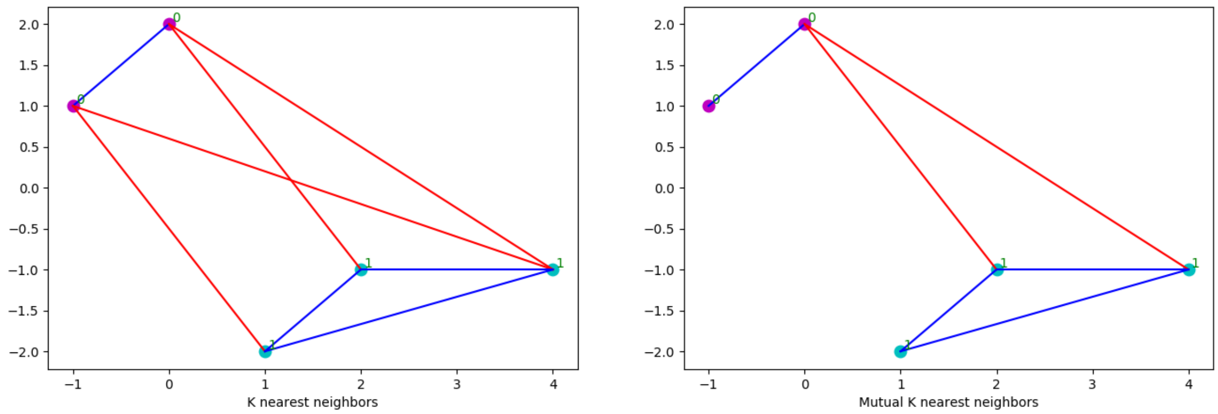


Figure 2.3: An example showing the difference between a mutual K nearest neighbors (MKNN) graph and a K nearest neighbors (KNN) graph. Both graphs are constructed on the same set of synthetic data points using cosine similarity measure. The two magenta points represent samples in class 0 while the other three cyan points represent samples from class 1. We plot edges in the KNN graph on the left and edges in the MKNN graph on the right. Blue edges represent edges whose incident nodes belong to the same class and red edges represent edges whose incident nodes belong to different classes.

2.5 Connections to Our Work

The following diagram shows how related work mentioned in previous sections connects to our proposed method, LSGMC:

Multi-view Clustering
View regularization (CCA)

Single-view Clustering
Non-centroid based clustering (RCC)
Local similarity graphs (MKNN)

New Multi-view Clustering
LSGMC

We draw ideas from techniques used in different domains. We use a popular technique for regularizing views, canonical correlation analysis (CCA), from multi-view clustering. We also use a non-centroid based clustering method, Robust continuous clustering (RCC), and local similarity graphs constructed by mutual K nearest neighbors (MKNN), in single-view clustering. We realize that current multi-view clustering approaches based on CCA can be improved with those ideas from single-view clustering.

In the next chapter, we first formally define the multi-view clustering problem. We then describe those techniques in detail. After that, we show how we combine those ideas into a new, improved multi-view clustering method.

Chapter 3

Background

In this chapter, we first formally define the multi-view clustering problem. We then introduce one single-view clustering approach, Robust Continuous Clustering (RCC) and one multi-view clustering approach, Canonical Correlation Analysis (CCA). We further introduce the deep extension of RCC: Clustering Driven Deep Embedding with Pairwise Constraints (CPAC) as well as two deep extensions of CCA: Deep Canonical Correlation Analysis (DCCA) and Deep Canonical Correlation Autoencoder (DCCAE). After that, we show how our proposed approach LSGMC improves upon DCCA and DCCAE based on ideas from RCC and CPAC.

3.1 Multi-view Clustering

We first formally define the multi-view clustering problem. Consider n data points available in V views. For the v-th view, $\boldsymbol{X}^{(v)} = [\boldsymbol{x}_1^{(v)}, \boldsymbol{x}_2^{(v)}, \ldots, \boldsymbol{x}_n^{(v)}] \in \mathbb{R}^{n \times d_v}$, where $\boldsymbol{x}_i^{(v)}$ is the i-th data point and d_v is the feature number of view v. Our task is to partition n data points into c clusters based on $\{\boldsymbol{X}^{(v)}\}_{v=1}^V$.

3.2 Robust Continuous Clustering

Robust Continuous Clustering (RCC) [29] is a single-view, non-centroid based clustering method which uses local information from nearest neighbor graphs based on a relaxation of convex clustering problems. Given n data points of a single view with d features $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, RCC optimizes an embedding space $\boldsymbol{U} \in \mathbb{R}^{n \times d}$ through the following objective:

$$\min_{\boldsymbol{U}} \frac{1}{2} \sum_{i=1}^{n} |\boldsymbol{x}^{(i)} - \boldsymbol{u}^{(i)}|_2^2 + \frac{\lambda}{2} \sum_{p,q \in \mathcal{E}} w_{p,q} \rho(|\boldsymbol{u}^{(p)} - \boldsymbol{u}^{(q)}|_2)$$

Here, $\mathcal{E}$ is a set of edges in a connectivity graph constructed by MKNN on $\mathbf{X}$. ρ is a penalty function on the regularization norm that can be chosen as, for example, the l_2 norm. $w_{p,q}$ is the weight balancing the contribution of each data point to the pairwise terms. λ is a parameter balancing the two terms in the objective.

Clustering-driven Deep Embedding with Pairwise Constraints (CPAC) [11] is an extension of RCC, which learns the embedding space $\boldsymbol{U}$ through a deep autoencoder and reconstruct samples from the embedding space through a decoder. CPAC has been shown to outperform RCC on several datasets. Compared to RCC, apart from the ability of directly extracting non-linear features from data through an autoencoder, one advantage of CPAC is that the learned embedding space $\boldsymbol{U}$ can be a lower dimensional space than the original $\boldsymbol{X}$. Let $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{U} \in \mathbb{R}^{n\times m}$, where $m < d$. CPAC reformulates the objective as follows, where $\boldsymbol{v}^{(i)}$ denotes the i-th reconstructed data samples through the decoder and $\boldsymbol{x}^{(i)}$ denotes the i-th data sample:

$$\min_{\theta,\delta} \frac{1}{d}\sum_{i=1}^{n} |\boldsymbol{v}^{(i)} - \boldsymbol{x}^{(i)}|_2^2 + \frac{\lambda}{m}\sum_{p,q\in\mathcal{E}} w_{p,q}\rho(|\boldsymbol{u}^{(p)} - \boldsymbol{u}^{(q)}|_2^2)$$

where $\mathcal{E}, \lambda, \rho, w$ have the same definition as in RCC, and θ, δ are parameters for the autoencoder and decoder respectively.

3.3 Canonical Correlation Analysis (CCA)

We introduce a general formulation of Canonical Correlation Analysis (CCA), a widely applied data analysis technique. Given a dataset with two views $\boldsymbol{X}_1, \boldsymbol{X}_2$, let $\boldsymbol{\Sigma}_1$, $\boldsymbol{\Sigma}_2$ be the covariance matrices for each view and $\boldsymbol{\Sigma}_{12}$ be the cross-covariance. CCA finds pairs of linear projections of the two views $(\boldsymbol{w}_1^T\boldsymbol{X}_1, \boldsymbol{w}_2^T\boldsymbol{X}_1)$ such that they are maximally correlated:

$$\max_{\boldsymbol{w}_1,\boldsymbol{w}_2} \frac{\boldsymbol{w}_1^T\boldsymbol{\Sigma}_{12}\boldsymbol{w}_2}{\sqrt{\boldsymbol{w}_1^T\boldsymbol{\Sigma}_1\boldsymbol{w}_1\boldsymbol{w}_2^T\boldsymbol{\Sigma}_2\boldsymbol{w}_2}}$$

CCA is more often reformulated as

$$\begin{aligned} &\max_{\boldsymbol{w}_1,\boldsymbol{w}_2} \boldsymbol{w}_1^T\boldsymbol{\Sigma}_{12}\boldsymbol{w}_2 \\ &\text{subject to } \boldsymbol{w}_1^T\boldsymbol{\Sigma}_1\boldsymbol{w}_1 = \boldsymbol{w}_2^T\boldsymbol{\Sigma}_2\boldsymbol{w}_2 = 1 \end{aligned}$$

We often consider finding multiple pairs of projections, where CCA is typically formulated to obtain such directions that subsequent projections are uncorrelated with previous ones:

$$\begin{aligned} &\max_{\boldsymbol{W}_1,\boldsymbol{W}_2} Tr(\boldsymbol{X}_1\boldsymbol{\Sigma}_{12}\boldsymbol{X}_2) \\ &\text{subject to } \boldsymbol{W}_1^T\boldsymbol{\Sigma}_1\boldsymbol{W}_1 = \boldsymbol{W}_2^T\boldsymbol{\Sigma}_2\boldsymbol{W}_2 = \boldsymbol{I} \\ &\qquad \boldsymbol{w}_1^{(i)T}\boldsymbol{\Sigma}_{12}\boldsymbol{w}_2^{(j)} = 0, \forall i,j \in \{1,2,\dots,n\}, i \neq j \end{aligned}$$

3.4 Deep Canonical Correlation Analysis

With recent advances in deep learning, traditional CCA is sometimes parameterized with autoencoders to better extract features and learn data representation. Deep Canonical Correlation

Analysis (DCCA) [5] is a deep extension of CCA. DCCA is shown to outperform traditional CCA and kernel CCA on two-view clustering tasks across a variety of datasets. Let $f_{1;\theta_1}, f_{2;\theta_2}$ denote the autoencoders for data from view 1 and view 2, and θ_1, θ_2 denote the parameters. We have the DCCA objective as follow:

$$\begin{aligned}
\max_{\boldsymbol{W}_1, \boldsymbol{W}_2, \theta_1, \theta_2} \; & \frac{1}{n} Tr(\boldsymbol{W}_1^T f_{1;\theta_1}(\boldsymbol{X}_1) f_{2;\theta_2}(\boldsymbol{X}_2)^T \boldsymbol{W}_2) \\
\text{subject to } & \boldsymbol{W}_1^T (\frac{1}{n} f_{1;\theta_1}(\mathbf{X}_1) f_{1;\theta_1}(\boldsymbol{X}_1)^T + r_1 \boldsymbol{I}) \boldsymbol{W}_1 = \boldsymbol{I} \\
& \boldsymbol{W}_2^T (\frac{1}{n} f_{2;\theta_2}(\boldsymbol{X}_2) f_{2;\theta_2}(\mathbf{X}_2)^T + r_2 \mathbf{I}) \boldsymbol{W}_2 = \boldsymbol{I} \\
& \boldsymbol{w}_1^{(i)T} f_{1;\theta_1}(\boldsymbol{X}_1) f_{2;\theta_2}(\boldsymbol{X}_2)^T \boldsymbol{w}_2^{(j)} = 0, \forall i, j \in \{1, 2, \ldots, n\}, i \neq j
\end{aligned}$$

where $r_1, r_2 > 0$ are regularization parameters.

An improved version of DCCA, Deep Canonical Correlation Autoencoder (DCCAE) [31] combines DCCA and reconstruction based regularization. The method optimizes both the correlation between the embedding of two views and the reconstruction error through decoders. DCCAE essentially offers a trade-off between the information captured in the projection to a lower dimensional embedding space and the information in the relationship across different views. Let $\boldsymbol{v}$ denote the reconstructed data, $\boldsymbol{x}^{(i)}$ denotes the i-th sample and δ_1, δ_2 denote the parameters for the decoders. The DCCAE objective is as follow:

$$\begin{aligned}
\max_{\boldsymbol{W}_1, \boldsymbol{W}_2, \theta_1, \theta_2, \delta_1, \delta_2} & \frac{1}{n} Tr(\boldsymbol{W}_1^T f_{1;\theta_1}(\boldsymbol{X}_1) f_{2;\theta_2}(\boldsymbol{X}_2)^T \boldsymbol{W}_2) \\
& + \frac{\lambda}{n} (\sum_{i=1}^{n} |\boldsymbol{x}_1^{(i)} - \boldsymbol{v}_1^{(i)}|_2^2 + |\boldsymbol{x}_2^{(i)} - \boldsymbol{v}_2^{(i)}|_2^2)
\end{aligned}$$

with the same set of constraints as DCCA. $\lambda > 0$ is a trade-off parameter.

3.5 Improvement Over Existing Approaches

We list and compare how the objective function changes over correlation based multi-view clustering approaches: CCA, DCCA, DCCAE and our proposed LSGMC. We highlight the improvement as the yellow part.

- CCA objective [1936]:

$$\max_{\boldsymbol{W}_1, \boldsymbol{W}_2} Tr(\boldsymbol{X}_1 \boldsymbol{\Sigma}_{12} \boldsymbol{X}_2)$$

- DCCA objective [2013]:

$$\max_{\boldsymbol{W}_1, \boldsymbol{W}_2, \theta_1, \theta_2} \frac{1}{n} Tr(\boldsymbol{W}_1^T f_{1;\theta_1}(\boldsymbol{X}_1) f_{2;\theta_2}(\boldsymbol{X}_2)^T \boldsymbol{W}_2)$$

- DCCAE objective [2015]:

$$\max_{\boldsymbol{W}_1,\boldsymbol{W}_2,\theta_1,\theta_2,\delta_1,\delta_2} \text{DCCA objective} + \frac{\lambda}{n}\left(\sum_{i=1}^{n} |\boldsymbol{x}_1^{(i)} - \boldsymbol{v}_1^{(i)}|_2^2 + |\boldsymbol{x}_2^{(i)} - \boldsymbol{v}_2^{(i)}|_2^2\right)$$

- LSGMC objective [proposed]:

$$\max_{\boldsymbol{W}_1,\boldsymbol{W}_2,\theta_1,\theta_2,\delta_1,\delta_2} \text{DCCAE objective} + \sum_{p,q\in\mathcal{E}} \sum_{v=1}^{2} |f_{v;\theta_v}(x_v^{(p)}) - f_{v;\theta_v}(x_v^{(q)})|_2^2$$

We have introduced techniques from both single-view clustering and multi-view clustering. We have further described how our proposed approach improves upon existing correlation based multi-view clustering approaches. In the next chapter, we describe our proposed multi-view clustering approach LSGMC in detail.

Chapter 4

Local Similarity Graph based Multi-view Clustering (LSGMC)

In this chapter we describe each component of LSGMC in detail. **Figure** 4.1 provides an overview of the network architecture. We learn a lower dimensional embedding for data in each view and maximize the consistency between data across views through canonical correlation analysis (CCA). Additionally, we construct view specific local similarity graphs through mutual K nearest neighbors based on the original data of each view. We describe how we use the graphs to explore first order proximity within views and complementary information across views, which provide additional signals for learning a better data representation. We focus our analysis on data with two views ($V = 2$), but our approach can be extended to more than two views. In the following paragraphs we detail the steps of the proposed clustering algorithm for which pseudo code is provided in **Algorithm** 1.

4.1 Learning Latent Data Representations

Since data from different views may stem from various different distributions–e.g. text data and image data–it is natural to learn a representation of the data in some lower dimensional subspace through nonlinear maps. This allows us to compute additional losses across views in the lower dimensional representation. Autoencoders have been demonstrated as an effective way of modeling feature nonlinearity on a variety of single-view clustering approaches [11, 35], and multi-view clustering approaches [5, 15, 31, 39]. In LSGMC, we use autoencoders to learn an embedding for each view. Let f_{θ_v}, g_{ϕ_v} denote the autoencoder and decoder for view v, with parameters θ_v, ϕ_v respectively. We fix the dimension of the embedding space to be p for all views. Let $\boldsymbol{U}^{(v)} = f_{\theta_v}(\boldsymbol{X}^{(v)}) \in \mathbb{R}^{n \times p}$ be the latent embedding for view v. We learn the embedding by the following reconstruction loss:

$$L_{rec}^{(v)} = |\boldsymbol{X}^{(v)} - g_{\phi_v}(\boldsymbol{U}^{(v)})|_2^2 \tag{4.1}$$

Note that the reconstruction loss is an important component to keep the embedding learned through LSGMC meaningful and to prevent inevitable false connections provided in local similarity graphs from corrupting the embedding.

Algorithm 1 LSGMC training procedure

Input: $\{\boldsymbol{X}^{(v)}\}_{v=1}^{V}$, hyperparameters λ_1, λ_2, common edge weight update interval c.
Output: cluster labels y.
for v in $1 \dots V$ **do**
    $\theta_v, \phi_v \leftarrow$ *weight_init()*
    Construct local similarity graph $\boldsymbol{G}^{(v)}$.
end for
Compute $\boldsymbol{G}_{union} = \cup_{v=1}^{V} \boldsymbol{G}^{(v)}$.
Compute $\boldsymbol{G}_{common} = \cap_{v=1}^{V} \boldsymbol{G}^{(v)}$.
for epoch in $1 \dots$ max epoch **do**
    for all sampled minibatch indices **do**
        Sample a batch of $\boldsymbol{X}^{(v)}$, $\forall v$.
        Compute $L_{rec}^{(v)}$, $\forall v$ by **Eq.** 4.1.
        Compute $L_{prox}^{(v)}$, $\forall v$ by **Eq.** 4.3.
        Compute L_{corr} by **Eq.** 4.2.
        $\theta_v \leftarrow \text{RMSprop}(\theta_v, -\nabla L_{rec}^{(v)}, -\nabla L_{prox}^{(v)}, -\nabla L_{corr})$
        $\phi_v \leftarrow \text{RMSprop}(\phi_v, -\nabla L_{rec}^{(v)}, -\nabla L_{prox}^{(v)}, -\nabla L_{corr})$
    end for
    if epoch mod $c == 0$ **then**: ▷ Optional
        Uniformly sample a subset of edges $\boldsymbol{G}'$ from $\boldsymbol{G}_{common}$.
        Compute L_{com} based on $\boldsymbol{G}'$ by **Eq.** 4.4.
        $\theta_v \leftarrow \text{RMSprop}(\theta_v, -\nabla L_{com})$
        $\phi_v \leftarrow \text{RMSprop}(\phi_v, -\nabla L_{com})$
    end if
end for
$y \leftarrow$ KMeans($\boldsymbol{U}^{(v)}$) for any v.

4.2 Consistency Between Data Across Views

One widely accepted assumption in multi-view clustering is that data from different views have certain forms of consistency. Through maximizing a linear correlation between data from two views, CCA is an effective way of maximizing data consistency across views. Since CCA has a closed form solution [5], we can maximize view consistency through the following correlation loss. Let

$$\bar{\boldsymbol{U}}^{(v)} = \boldsymbol{U}^{(v)} - \frac{1}{n}\boldsymbol{U}^{(v)}\mathbf{1}$$

be the re-centered embedding for view v,

$$\boldsymbol{\Sigma}^{(v)} = \frac{1}{n-1}\bar{\boldsymbol{U}}^{(v)T}\bar{\boldsymbol{U}}^{(v)}$$

the covariance matrix of view v's embedding, and

$$\boldsymbol{\Sigma}^{(v_1,v_2)} = \frac{1}{n-1}\bar{\boldsymbol{U}}^{(v_1)T}\bar{\boldsymbol{U}}^{(v_2)}$$

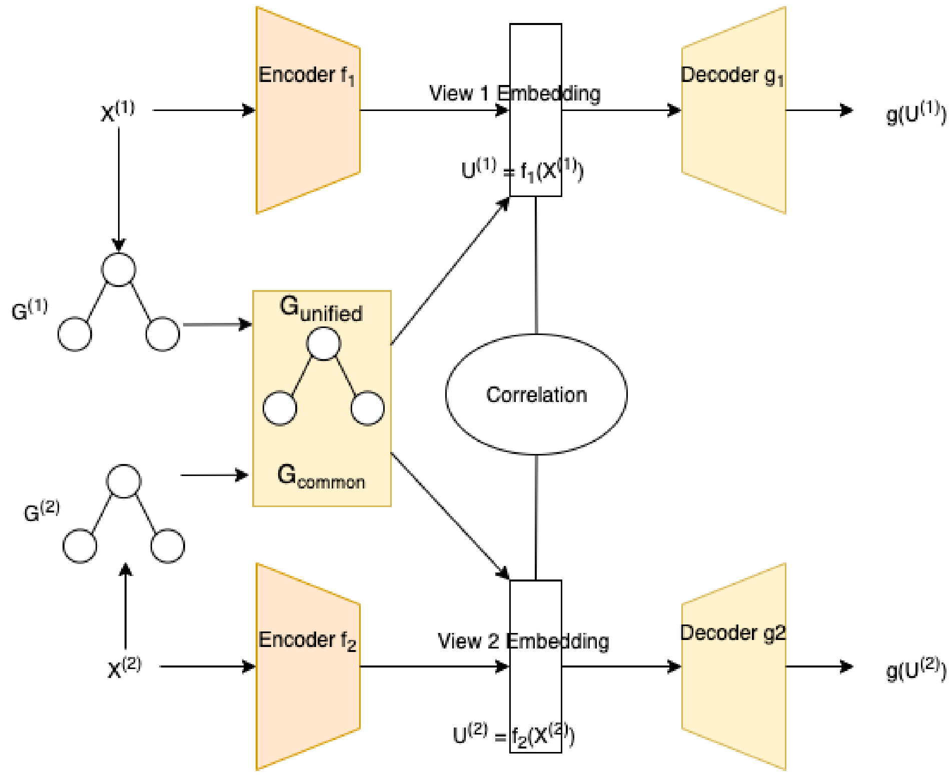


Figure 4.1: Architecture Overview: LSGMC uses one autoencoder per view (f_1, f_2) to project the data from the respective view to an embedding space ($\boldsymbol{U}^{(1)}$, $\boldsymbol{U}^{(2)}$) and a corresponding decoder (g_1, g_2) reconstructs the data representation. LSGMC further constructs local similarity graphs for each view ($\boldsymbol{G}^{(1)}$, $\boldsymbol{G}^{(2)}$) based on the input data. LSGMC merges local similarity graphs into a unified graph ($\boldsymbol{G}_{unified}$) and optionally constructs a graph with common edges between view specific local similarity graphs ($\boldsymbol{G}_{common}$). LSGMC trains the encoders and decoders to keep reconstruction error of each view small, maximizes correlation between embeddings of different views to enforce view consistency, all while explores first order proximity within each view and complementary information across views via $\boldsymbol{G}_{unified}$ and $\boldsymbol{G}_{common}$ to search for better data representations.

the cross-covariance matrix between embeddings of view v_1 and v_2. We further denote

$$\boldsymbol{T} = \boldsymbol{\Sigma}^{(v_1)-1/2}\boldsymbol{\Sigma}^{(v_1,v_2)}\boldsymbol{\Sigma}^{(v_2)-1/2}$$

and using the singular values of $\boldsymbol{T}$ we define the correlation loss

$$L_{corr} = -\sum_{i=1}^{p}\sigma_i \tag{4.2}$$

where the σ_i's are the top p singular values of $\boldsymbol{T}$, with p being the dimension of the latent embedding.

4.3 Utilizing Local Similarity Graphs

A desirable property of good data representations is that if two data points are close in the original space via some domain appropriate similarity function, their representations in the embedding space should also be close. Further, one might observe that if two data points are close to each other in the original space, they are more likely to be from the same cluster. We call a graph of pairs established via such a heuristic a local similarity graph and use it to guide the search for a better embedding since an edge in such a graph indicates a higher probability that two incident nodes belong to the same cluster.

To further improve the quality of the graph, we use a mutual nearest neighbor approach, meaning that two nodes are only connected if they are both nearest neighbors of each other in a K nearest neighbor graph. Prior research [8] has shown that graphs constructed through mutual K nearest neighbors can better capture local similarities in clustering tasks. We construct one such local similarity graph $\boldsymbol{G}^{(v)}$ for each view v and in our experiments use cosine similarity between data points in the original space.

4.4 Using Information From Graphs Across Views

We observe that view specific local similarity graphs encode view specific information as they differ across views due to different data representations in the respective original spaces. Thus, view specific graphs can provide complementary information. An edge connected in $\boldsymbol{G}^{(v_1)}$ might be disconnected in $\boldsymbol{G}^{(v_2)}$ but is expected to still provide valuable information in learning the embedding for view v_2. Thus we consider unifying all view specific graphs into one common graph $\boldsymbol{G} = \cup_{i=1}^{V} \boldsymbol{G}^{(v)}$ and use the common graph to guide the learning of embeddings for all views.

To encourage first order proximity in the embedding space, we define the following proximity loss, $\forall v$,

$$L_{prox}^{(v)} = \sum_{(i,j)\in \boldsymbol{G}} |\boldsymbol{U}_i^{(v)} - \boldsymbol{U}_j^{(v)}|_2^2 \tag{4.3}$$

It is important to note two potential drawbacks of the proximity loss in isolation. First, a trivial solution to minimizing the loss is to collapse all data representations in the embedding into a single cluster. The embedding thus fails to represent the actual relationship between data points. Second, due to relying on a heuristic such as K mutual nearest neighbors, view specific graphs will inevitably contain false connections. $L_{prox}^{(v)}$ by itself provides no mechanism to account for such mistakes. However, the reconstruction loss is able to counter such issues. By forcing a reconstructed sample to be close to its representation in the original space, the reconstruction loss encourages learning of a meaningful embedding and avoids collapse of all samples towards a single point.

4.5 Injecting Beliefs About Consensus

We may consider a complementary operation of unifying all view specific similarity graphs $\boldsymbol{G}^{(v)}$ to find the common edges among all graphs. A natural belief derived from the multi-view consensus clustering assumption is that edges occurring in all view specific local similarity graphs are more likely to provide true information that two incident nodes belong to the same cluster. In order to avoid reinforcing false information, we consider uniformly sampling a subset of the common edges. Let $\boldsymbol{G'} = \text{sample}(\cap_{v=1}^{V} \boldsymbol{G}^{(v)})$. We can encourage proximity through a subset of common edges as follows

$$L_{com} = \sum_{(i,j) \in G'} |\boldsymbol{U}_i^{(v)} - \boldsymbol{U}_j^{(v)}|_2^2 \tag{4.4}$$

every constant number of epochs, to reinforce the consensus belief.

4.6 Overall Objective and Optimization

Let $\Theta = \{\theta_v, \phi_v\}_{v=1}^{V}$ denote the overall parameters to train. The joint objective including all previously mentioned losses is

$$L_{\Theta} = \sum_{v=1}^{V} \left(L_{rec}^{(v)} + \lambda_1^{(v)} L_{prox}^{(v)} \right) + L_{corr} + \lambda_2 L_{com} \tag{4.5}$$

where λ_1 and λ_2 are trade-off hyperparameters to balance the losses. In our experiments, we set $\lambda_1^{(v)} = \frac{|X^{(v)}|_F}{\sigma}$, where $|\cdot|_F$ denotes the Frobenius norm and σ is the largest eigenvalue of the graph Laplacian based on the cosine similarity matrix we used to construct $G^{(v)}$. See [29] for details. We set $\lambda_2 = 1$. The overall LSGMC learning procedure is presented in **Algorithm** 1. Equation 4.5 can be minimized through standard backpropagation algorithms to update parameters in the encoder and the decoder for each view. In our implementation we use RMSprop to learn Θ and apply K Means to obtain the final clustering based on the embeddings that were learned. We note that the correlation loss can be optimized efficiently only if the gradient is estimated using a sufficiently large minibatch [31].

4.7 Extension to Semi-supervised Clustering

In some applications one may have access to reliable pairwise information for a small portion of observations from ground truth or meta data. Such pairs usually come in the form of pairwise *must-link* and *cannot-link* constraints. LSGMC is able to incorporate such information by augmenting all local similarity graphs.

We have presented our proposed LSGMC. In the next chapter, we describe the experiment settings for evaluating LSGMC.

Chapter 5

Experiment Setup

In this chapter, we present the experiment settings to evaluate LSGMC. We begin by introducing four benchmark and real world datasets for evaluating multi-view clustering approaches from different domains that will be used in our experiments. We then introduce five state-of-the-art multi-view clustering alternatives that our LSGMC will be compared against. We further describe our implementation details and introduce four commonly applied evaluation metrics for clustering.

5.1 Datasets

We use four widely used multi-view clustering benchmarks and real world datasets for LSGMC evaluation. **Figure** 5.1 and **Figure** 5.2 show examples from the Noisy MNIST dataset and the Digit/MNIST-USPS dataset respectively. **Table** 5.1 presents a summary of all datasets used in the experiments.

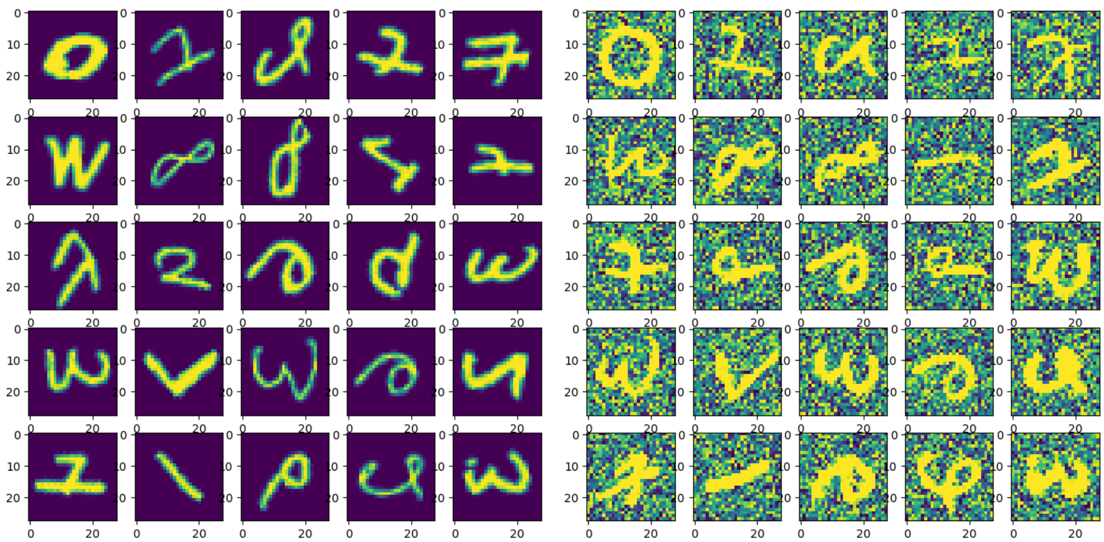
Dataset	# samples	# features view 1	# features view 2	# clusters	type
Noisy MNIST	4000	784	784	10	image
Digit/MNIST-USPS	4000	784	256	10	image
BBC+The Guardian	169	3560	3631	6	text
XRMB	20,000	273	112	20	acoustic-articulatory

Table 5.1: A summary of experiment datasets.

5.1.1 Noisy MNIST

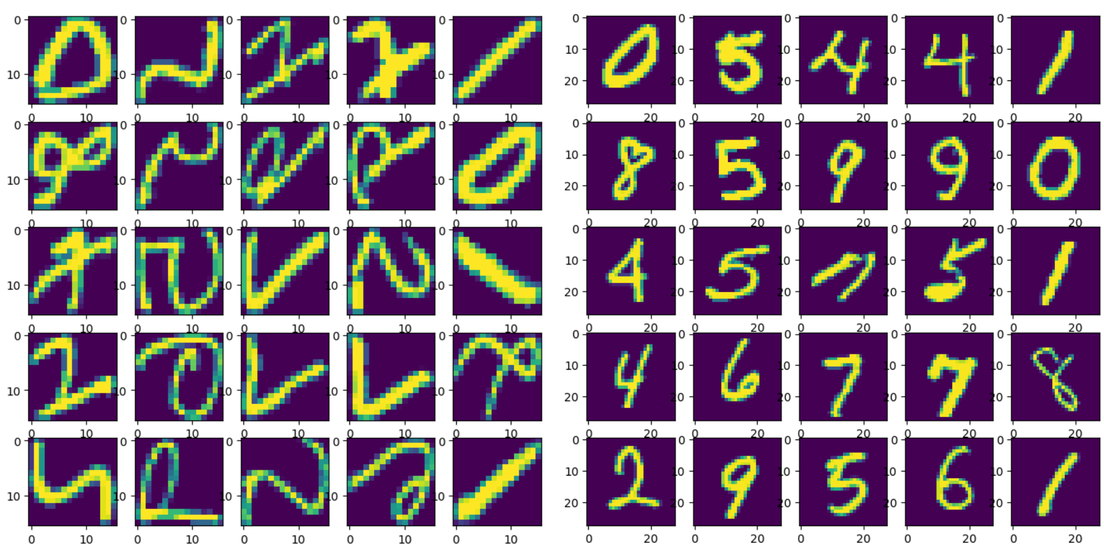
[1] Following [31], we create two noisy views from MNIST handwritten digits [18]. We randomly select a small subset of 4000 samples out of the entire 70K samples, with 400 samples per class. The first view is a random rotation of the original digit with angles sampled uniformly within range $[-\frac{\pi}{4}, \frac{\pi}{4}]$. The corresponding second view is randomly selected from the same class

[1] https://csc.lsu.edu/~saikat/n-mnist/



(a) Noisy MNIST view 1 (b) Noisy MNIST view 2

Figure 5.1: Examples of Noisy MNIST data.



(a) Digit/MNIST-USPS view 1 (b) Digit/MNIST-USPS view 2

Figure 5.2: Examples of Digit/MNIST-USPS data.

from the original dataset. The pixel values are scaled to $[0, 1]$, masked with i.i.d noise uniformly drawn from $[0, 1]$, and subsequently truncated to $[0, 1]$. Each view has a dimension of 784. The dataset has 10 classes.

5.1.2 Digit/MNIST-USPS

Following [26] we randomly select 4000 samples from MNSIT and USPS, with 400 samples per class. We randomly match digits from MNIST with digits of the same class from USPS to form a two view handwritten digits data. The first view (MNIST) has a dimension of 784 and the second view (USPS) has a dimension of 256. The dataset has 10 classes.

5.1.3 BBC+The Guardian

[2] The Three Sources Dataset is a multi-view text dataset consisting of news articles collected from three online news sources: BBC, The Guardian, and Reuters. All articles are represented as bag-of-words and each article is annotated with at least one of six topics. Following [7], we use 169 articles that are available in all three sources. We use one annotation for each article and use BBC and The Guardian as the first and second view. The first view (BBC) has a dimension of 3560 and the second view (the Guardian) has a dimension of 3631. The dataset has 6 classes.

5.1.4 XRMB

[3] The Wisconsin X-ray Microbeam (XRMB) data consists of simultaneously recorded speech and articulation measurements from 47 American English speakers. We use a processed version from [32]. The first view (acoustic measurements) has 39 features consisting of mel-frequency cepstral coefficients (MFCCs) and their first and second derivatives, and the second view (articulation measurements) has horizontal/vertical displacement of 8 pellets attached to different parts of the vocal tract. Both features are concatenated over a 7-frame window around each frame, resulting in a total of 273 features for the first view and 112 features for the second view. The task is to cluster phones based on two set of features. We randomly sample 20 phones from 35 speakers, with each phone 1000 samples, resulting in a total of 20,000 samples.

5.2 Methods for Comparison

We use **LSGMC** to denote our proposed approach without injecting beliefs about consensus among edges across local similarity graphs from different views. We use **LSGMC+** to denote our proposed approach with injecting beliefs about consensus. In our experiments using **LSGMC+**, we uniformly sample 20% common edges and compute L_{com} every 5 epochs. We compare **LSGMC** and **LSGMC+** against the following state-of-the-art multi-view clustering approaches:

5.2.1 Deep Canonical Correlation Analysis (DCCA)

Deep Canonical Correlation Analysis (DCCA) [5] is a multi-view clustering method based on canonical correlation analysis (CCA) for regularizing data from two views. DCCA improves

[2] `https://github.com/mbrbic/Multi-view-LRSSC`

[3] `https://ttic.uchicago.edu/~klivescu/XRMB_data/full/README`

CCA by using deep autoencoders to better extract nonlinear features from data. DCCA is shown to outperform CCA and kernel CCA on various datasets.

5.2.2 Deep Canonically Correlated Autoencoders (DCCAE)

Deep Canonically Correlated Autoencoders (DCCAE) [31] is an improved version of DCCA. DCCAE uses not only CCA parameterized by deep autoencoders but also uses decoders and reconstruction errors to improve the learned data embedding. DCCAE is able to outperform DCCA on several datasets.

5.2.3 Deep Matrix Factorization (DMF)

Deep Matrix Factorization (DMF) [41] explores complementary information across views by using semi-nonnegative matrix factorization to learn the hierarchical semantics of multi-view data in a layer-wise fashion, with graph regularizers to represent intrinsic geometric structure in each view data.

5.2.4 Low-rank Sparse Subspace Clustering (LRSSC)

Multi-view Low-rank Sparse Subspace Clustering (LRSSC) [7] learns a joint subspace representation by constructing the similarity matrix shared among all views while encouraging sparsity and low-rankness solutions at the same time. Note that LRSSC has four different variations. We report the highest scores among the four variations in our experiments.

5.2.5 Deep Multimodal Subspace Clustering (DMSC)

Deep Multimodal Subspace Clustering (DMSC) [1] is a convolutional neural network based deep multimodel subspace clustering mainly for image datasets. DMSC uses encoders and decoders for learning a lower dimensional data representation and explores different affinity fusion techniques to regularize data across views. The DMSC implementation does not allow the clustering of multi-view data with different dimensionality for each view. We are thus only able to report the results of DMSC on dataset where the number of features in both views is the same.

5.3 Implementation Details

In order to enable a fair comparison of LSGMC against DCCA and DCCAE, we use the same set of hyperparameters and optimizer across the three approaches. In computing correlation loss, we use the top p singular values, where p is the same as the embedding dimension. We evaluate all clustering methods in an unsupervised clustering setting using K means clustering on the learned data representation for each view and report the best score among two views. For our approach, to construct the local similarity graph for each view, we fix the number of nearest neighbors to 10 to create a mutual K nearest neighbors graph for each dataset. [22] suggests a lower bound on the number of neighbors in a mutual K nearest neighbors graph to successfully identify clusters as K $\propto \log(n)$ where n is the number of samples. In our experiments, our samples range from about 200 to 4000, meaning the lower bound lies in $[7.5, 12]$.

We use the RMSprop optimizer with a weight decay of $1e-5$ and a learning rate of $1e-3$. We fix the dimension of the embedding space to be 10, which is the same as the original setting used in both DCCA and DCCAE. The autoencoder for each view has three hidden layers and each layer has 1024 units. We run the experiment for at most 100 epochs for Noisy MNIST, BBC+The Guardian and XRMB dataset, at most 300 epochs for Digit/MNIST-USPS dataset. We use a batch size of 800 in each minibatch update. We use random initialization of all parameters in the encoders and decoders without pre-training. We did not find a significant improvement in the clustering performance using pre-training.

5.4 Evaluation Metrics

Consistent with relevant literature [4, 23], we use four extrinsic metrics, which compare the output of the clustering algorithm and a ground truth (true classes/labels), for evaluating the clustering performance: Normalized Mutual Information (NMI), Adjusted RAND Index (ARI), F1 score and Purity for evaluating clustering performance. Those four widely applied metrics in clustering literature captures different aspects of the clustering algorithms. NMI is a metric based on entropy. The entropy of a predicted cluster reflects how the members of different true classes are distributed within each cluster. ARI is a metric which considers statistics over pairs of items. Purity and F1 score are metrics based on set matching. Both assume a one to one mapping between predicted clusters and true classes, and used precision and recall for comparison. We give a detailed description of each metric below.

Since we are comparing the quality of learned data embedding, we assume the true number of clusters, k, is known a priori during evaluation. Let $Y = \{y_1, \ldots, y_k\}$ be the set of true classes, $C = \{c_1, \ldots, c_k\}$ be the set of predicted clusters and N be the number of samples.

5.4.1 Normalized Mutual Information (NMI)

NMI is an information theory based metric. NMI is defined as

$$
\begin{aligned}
\mathrm{NMI}(Y, C) &= \frac{2 \times I(Y;C)}{H(Y) + H(C)} \\
\mathrm{I}(Y;C) &= \sum_i \sum_j \Pr[y_i \cap c_j] \log \frac{\Pr[y_i \cap c_j]}{\Pr[y_i]\Pr[c_j]} \\
&= \sum_i \sum_j \frac{|y_i \cap c_j|}{N} \log \frac{N|y_i \cap c_j|}{|y_i||c_j|} \\
\mathrm{H}(Y) &= -\sum_i \Pr[y_i] \log \Pr[y_i] \\
&= -\sum_i \frac{|y_i|}{N} \log \frac{|y_i|}{N}
\end{aligned}
$$

where $\Pr[y_i]$, $\Pr[c_j]$ and $\Pr[y_i \cap c_j]$ denote the probability of a data sample being in class y_i, predicted cluster c_j and in the intersection of y_i and c_j, respectively. $\mathrm{I}(\cdot;\cdot)$ is the mutual information and $\mathrm{H}(\cdot)$ is the entropy. $\mathrm{H}(C)$ is defined similarly as $\mathrm{H}(Y)$.

Mutual information (MI) $\mathrm{I}(Y;C)$ measures the amount of information by which our knowledge about the true class increases when we are told what the predicted clusters are. The minimum of $\mathrm{I}(Y;C)$ is 0 if the clustering is random. The maximum of $\mathrm{I}(Y;C)$ is when C matches Y exactly. A larger number of clusters will result in a larger $\mathrm{I}(Y;C)$. In order to have a more direct comparison when the number of clusters varies, MI is normalized by $\frac{\mathrm{H}(Y)+\mathrm{H}(C)}{2}$ since a larger number of clusters results in an increase in the entropy. $\frac{\mathrm{H}(Y)+\mathrm{H}(C)}{2}$ is a tight upper bound of $\mathrm{I}(Y;C)$ and thus $\mathrm{NMI}(Y;C) \in [0,1]$.

5.4.2 Adjusted RAND Index (ARI)

ARI measures the similarity between cluster assignments based on pairs of samples. ARI is an adjusted version of the RAND index (RI) to account for the fact that RI sometimes fails to capture the quality of the clustering assignment C due to randomness. We provide a detailed derivation of ARI from RI in this section based on [3, 14]. We begin by introducing RI, the drawback of RI under randomness and then discuss how ARI comes into the picture.

Consider the contingency table between true classes Y and predicted clusters C, where $n_{ij} = |c_i \cap y_j|$, $n_{i\cdot} = |c_i|$ and $n_{\cdot j} = |y_j|$ as follow

	y_1	y_2	$\dots$	y_k	
c_1	n_{11}	n_{12}	$\dots$	n_{1k}	$n_{1\cdot}$
c_2	n_{21}	n_{22}	$\dots$	n_{2k}	$n_{2\cdot}$
$\dots$			$\dots$		
c_k	n_{k1}	n_{k2}	$\dots$	n_{kk}	$n_{k\cdot}$
	$n_{\cdot 1}$	$n_{\cdot 2}$	$\dots$	$n_{\cdot k}$	N

Table 5.2: The contingency table between true classes Y and predicted clusters C.

For each pair of samples among $\binom{N}{2}$ all possible pairs, one of the four cases: $\{$*True Positive*, *False Positive*, *False Negative*, *True Negative*$\}$ applies. We calculate the total number of samples in each case. Further notice that total # FP, total # FN and total # TN are all constant linear transformation of total # TP. We adopt part of the calculation and notation from [3]. Let $P = \sum_{i=1}^{k} n_{i\cdot}^2 - N$, $Q = \sum_{j=1}^{k} n_{\cdot j}^2 - N$.

1. *True Positive (TP)*: two samples from the same true class are assigned to the same cluster.

$$\begin{aligned}\text{total \# TP} &= \sum_{i=1}^{k}\sum_{j=1}^{k}\binom{n_{ij}}{2} \\ &= \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}n_{ij}^2 - \frac{N}{2}\end{aligned}$$

2. *False Positive (FP)*: samples from two different classes are assigned to the same cluster.

$$\begin{aligned}\text{total \# FP} &= \sum_{i=1}^{k}\binom{n_{i\cdot}}{2} \\ &= \frac{1}{2}\sum_{i=1}^{k}n_{i\cdot}^2 - \frac{1}{2}\sum_{i=1}^{k}\sum_{i=1}^{k}n_{ij}^2 \\ &= \frac{1}{2}P - [\text{total \# TP}]\end{aligned}$$

3. *False Negative (FN)*: samples from the same class are assigned to different clusters.

$$\begin{aligned}\text{total \# FN} &= \sum_{j=1}^{k}\binom{n_{\cdot j}}{2} \\ &= \frac{1}{2}\sum_{j=1}^{k}n_{\cdot j}^2 - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}n_{ij}^2 \\ &= \frac{1}{2}Q - [\text{total \# TP}]\end{aligned}$$

4. *True Negative (TN)*: two samples from different classes are assigned to different clusters.

$$\begin{aligned}\text{total \# TN} &= \binom{N}{2} - (\text{TP} + \text{FP} + \text{FN}) \\ &= \frac{1}{2}(N^2 + \sum_{i=1}^{k}\sum_{j=1}^{k}n_{ij}^2 - (\sum_{i=1}^{k}n_{i\cdot}^2 + \sum_{j=1}^{k}n_{\cdot j}^2)) \\ &= \binom{N}{2} - \frac{1}{2}(P+Q) + [\text{total \# TP}]\end{aligned}$$

The RAND index (RI) denotes the percentage of pairs of samples which have an agreed assignment in Y and C among all pairs – i.e. both samples in a pair are assigned to the same class in Y and C (TP) or both samples in a pair are assigned to different classes in Y and C (TN). Formally, RI is defined as

$$\begin{aligned}\text{RI} &= \frac{\text{total \# TP} + \text{total \# TN}}{\binom{N}{2}} \\ &= \frac{\binom{N}{2} + 2\sum_{i=1}^{k}\sum_{j=1}^{k}\binom{n_{ij}}{2} - (\sum_{i=1}^{k}\binom{n_{i\cdot}}{2} + \sum_{j=1}^{k}\binom{n_{\cdot j}}{2})}{\binom{N}{2}}\end{aligned}$$

Since RI counts the number of pairs that are assigned to different classes in Y and C, if the number of classes k is large, then there will be a higher chance for a pair that comes from different classes in Y to be assigned to different clusters in C, even when C is a bad cluster assignment. For example, consider two samples from different true classes $x_1 \in y_1$ and $x_2 \in y_2$. Consider a uniformly random clustering assignment C. When $k = 2$, x_1, x_2 are assigned to different clusters in C with probability 0.5. When $k = 100$, however, the probability of x_1, x_2 being assigned to different clusters becomes 0.99. Thus randomness might result in a high RI and RI might fail to reflect the true quality of C. We need a better measurement that can account for this fact.

Adjusted RAND Index (ARI) is an "adjusted for chance" correction to RI proposed by Hubert and Arabic in 1985 [14], under the assumption that contingency table is constructed from the generalized hypergeometric distribution, i.e. Y and C partitions are picked at random, subject to having the original number of classes and objects in each. A general form of a statistic index – in our case, the RAND index – corrected for chance is

$$\text{Corrected Index} = \frac{\text{Index} - \text{Expected Index}}{1 - \text{Expected Index}}$$

where the Expected Index represents the probability of a pairing due to randomness with fixed marginal counts and is calculated under the null distribution.

We begin by calculating the expected RAND index with fixed sets of marginal counts. As an analogy to the expected number of samples in the cell (i, j) in the contingency table being $\mathbb{E}[n_{ij}] = \frac{n_{i\cdot} \times n_{\cdot j}}{N}$, the expected number of TP pairs that are both assigned to y_i and c_j in the cell (i, j), is defined as

$$\mathbb{E}[\binom{n_{ij}}{2}] = \frac{\binom{n_{i\cdot}}{2} \times \binom{n_{\cdot j}}{2}}{\binom{N}{2}}$$

By linearity,

$$\mathbb{E}[\text{total \# TP}] = \mathbb{E}[\sum_{i=1}^{k}\sum_{j=1}^{k}\binom{n_{ij}}{2}] = \frac{\sum_{i=1}^{k}\binom{n_{i\cdot}}{2}\sum_{j=1}^{k}\binom{n_{\cdot j}}{2}}{\binom{N}{2}}$$

As we noted above that total # FP, total # FN, total # TN are all constant linear transformations of total # TP, we can derive $\mathbb{E}[\text{total \# TP} + \text{total \# TN}]$ directly from $\mathbb{E}[\text{total \# TP}]$. Therefore, we have

$$
\begin{aligned}
\mathbb{E}[\mathrm{RI}] &= \frac{\mathbb{E}[\text{total \# TP} + \text{total \# TN}]}{\binom{N}{2}} \\
&= \frac{\mathbb{E}[\binom{N}{2} + 2\sum_{i=1}^{k}\sum_{j=1}^{k}\binom{n_{ij}}{2} - (\sum_{i=1}^{k}\binom{n_{i\cdot}}{2} + \sum_{j=1}^{k}\binom{n_{\cdot j}}{2})]}{\binom{N}{2}} \\
&= \frac{\binom{N}{2} + 2\mathbb{E}[\sum_{i=1}^{k}\sum_{j=1}^{k}\binom{n_{ij}}{2}] - (\sum_{i=1}^{k}\binom{n_{i\cdot}}{2} + \sum_{j=1}^{k}\binom{n_{\cdot j}}{2})}{\binom{N}{2}} \\
&= 1 + 2\frac{\sum_{i=1}^{k}\binom{n_{i\cdot}}{2}\sum_{j=1}^{k}\binom{n_{\cdot j}}{2}}{\binom{N}{2}^2} - \frac{\sum_{i=1}^{k}\binom{n_{i\cdot}}{2} + \sum_{j=1}^{k}\binom{n_{\cdot j}}{2}}{\binom{N}{2}}
\end{aligned}
$$

With RI and $\mathbb{E}[\mathrm{RI}]$, we can now calculate ARI as follow

$$
\begin{aligned}
\mathrm{RI} - \mathbb{E}[\mathrm{RI}] = 1 + 2\frac{\sum_{i=1}^{k}\sum_{j=1}^{k}\binom{n_{ij}}{2}}{\binom{N}{2}} - \frac{\sum_{i=1}^{k}\binom{n_{i\cdot}}{2} + \sum_{j=1}^{k}\binom{n_{\cdot j}}{2}}{\binom{N}{2}} - \\
(1 + 2\frac{\sum_{i=1}^{k}\binom{n_{i\cdot}}{2}\sum_{j=1}^{k}\binom{n_{\cdot j}}{2}}{\binom{N}{2}^2} - \frac{\sum_{i=1}^{k}\binom{n_{i\cdot}}{2} + \sum_{j=1}^{k}\binom{n_{\cdot j}}{2}}{\binom{N}{2}}) \\
= 2(\frac{\sum_{i=1}^{k}\sum_{j=1}^{k}\binom{n_{ij}}{2}}{\binom{N}{2}} - \frac{\sum_{i=1}^{k}\binom{n_{i\cdot}}{2}\sum_{j=1}^{k}\binom{n_{\cdot j}}{2}}{\binom{N}{2}^2})
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{ARI} &= \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{1 - \mathbb{E}[\mathrm{RI}]} \\
&= \frac{2(\frac{\sum_{i=1}^{k}\sum_{j=1}^{k}\binom{n_{ij}}{2}}{\binom{N}{2}} - \frac{\sum_{i=1}^{k}\binom{n_{i\cdot}}{2}\sum_{j=1}^{k}\binom{n_{\cdot j}}{2}}{\binom{N}{2}^2})}{\frac{\sum_{i=1}^{k}\binom{n_{i\cdot}}{2}+\sum_{j=1}^{k}\binom{n_{\cdot j}}{2}}{\binom{N}{2}} - 2\frac{\sum_{i=1}^{k}\binom{n_{i\cdot}}{2}\sum_{j=1}^{k}\binom{n_{\cdot j}}{2}}{\binom{N}{2}^2}} \\
&= \frac{\sum_{i=1}^{k}\sum_{j=1}^{k}\binom{n_{ij}}{2} - [\sum_{i=1}^{k}\binom{n_{i\cdot}}{2}\sum_{j=1}^{k}\binom{n_{\cdot j}}{2}]/\binom{N}{2}}{\frac{1}{2}[\sum_{i=1}^{k}\binom{n_{i\cdot}}{2} + \sum_{j=1}^{k}\binom{n_{\cdot j}}{2}] - [\sum_{i=1}^{k}\binom{n_{i\cdot}}{2}\sum_{j=1}^{k}\binom{n_{\cdot j}}{2}]/\binom{N}{2}}
\end{aligned}
$$

The maximum ARI is 1 when Y and C have exactly the same cluster assignments. Notice that ARI can be negative under randomness. A random cluster assignment will get an expected ARI of 0.

5.4.3 F1 score

F1 score is a trade-off between clustering correctly all samples from the same true category into the same predicted cluster and making sure that each predicted cluster contains points from only one category. Let Precision and Recall for a true category y_i and a predicted cluster c_j defined as follow

$$\text{Precision}(y_i, c_j) = \frac{|c_j \cap y_i|}{|c_j|}$$
$$\text{Recall}(y_i, c_j) = \frac{|c_j \cap y_i|}{|y_i|}$$

Let F denote the harmonic mean of Precision and Recall. We have the F1 score as

$$\text{F}(y_i, c_j) = \frac{2 \times \text{Recall}(y_i, c_j)\text{Precision}(y_i, c_j)}{\text{Recall}(y_i, c_j) + \text{Precision}(y_i, c_j)}$$
$$\text{F1 score}(Y, C) = \sum_i \frac{|y_i|}{N} \max_j F(y_i, c_j)$$

A bad clustering will have a low F1 score, while an exact match between Y and C will get a F1 score of 1.

5.4.4 Purity

Purity is a simple and straight-forward evaluation metric based on set matching. When calculating purity, each predicted cluster c_j is assigned to the category y_i which is most frequent in the cluster and then the accuracy of this assignment is measured by counting the number of correctly assigned samples and dividing by N. Specifically, Purity is defined as

$$\text{Purity}(Y, C) = \frac{1}{N} \sum_j \max_i |y_i \cap c_j|$$

A bad clustering will get purity 0 and an exact matching between Y and C will get purity 1. However, high purity can also be achieved when the number of cluster is large. Thus we might also need to consider alternative evaluation metrics when assessing the clustering performance.

We have presented our experiment setups. In the next chapter, we present the experiment results. We further discuss and analyze our results.

Chapter 6

Results

In this chapter, we present our experiment results. We first compare evaluation metrics numerically on all datasets. We then visualize local similarity graphs used during training on two datasets, i.e. Noisy MNIST and BBC+The Guardian. We further visually compare the learned embeddings on the two datasets. We characterize the local similarity graphs used during training on all datasets. After that, we discuss and analyze our results. Finally we describe our results in semi-supervised setting.

6.1 Performance Comparison

We conduct 5 random runs of each experiment and report mean and standard deviation of values of the best under each metric from each run. **Figure** 6.1, **Figure** 6.2, **Figure** 6.3 and **Figure** 6.4 provide comparisons of the clustering performance of LSGMC against other state-of-art approaches on each respective dataset. We represent the performance of each multi-view clustering method as a column with mean value (in percentage) achieved under four evaluation metrics (NMI, ARI, F1 Score and Purity) described in the previous chapter across 5 runs. We represent the standard deviation as a black bar on top of each column. Under each metric, the two rightmost columns represent LSGMC (colored brown) and LSGMC+ (colored pink). We first compare LSGMC against related methods and then compare LSGMC against LSGMC+.

LSGMC, DCCA and DCCAE perform better than the other clustering methods on the image datasets. On Noisy MNIST, LSGMC significantly outperforms DCCA and DCCAE, across all performance metrics we compute, see **Figure** 6.1. On Digit/MNIST-USPS, LSGMC also outperforms all related approaches, but the gap in performance is less pronounced as the methods achieve high quality clustering results, see **Figure** 6.2.

On the text data, LSGMC and LRSSC perform better than the related methods. LSGMC significantly outperforms LRSSC as can be seen in **Figure** 6.3. On the acoustic-articulatory data, LSGMC and DCCA perform better than the related methods. LSGMC slightly outperforms DCCA as can be seen in **Figure** 6.4.

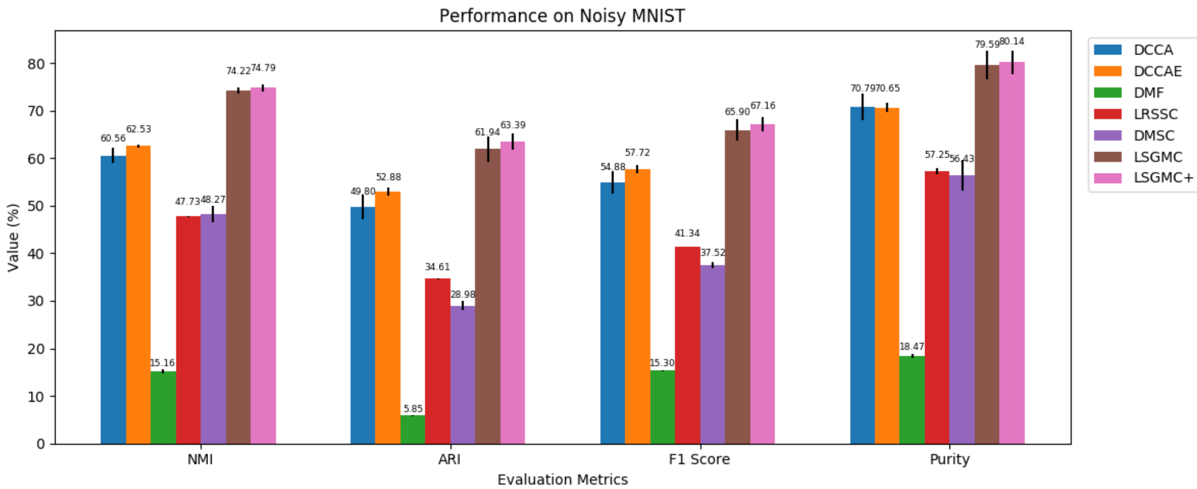


Figure 6.1: Performance on **Noisy MNIST** dataset.

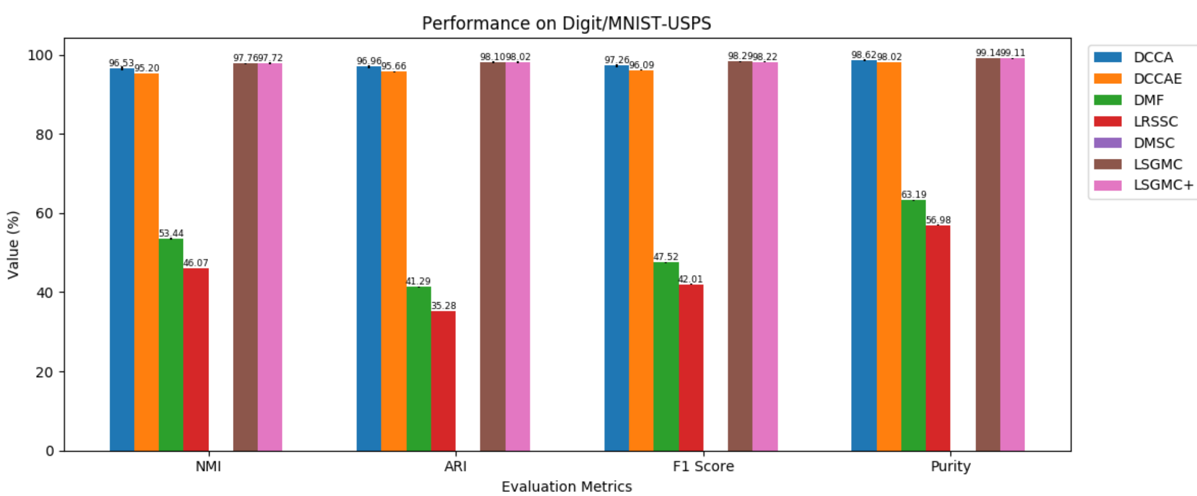


Figure 6.2: Performance on **Digit/MNIST-USPS** dataset.

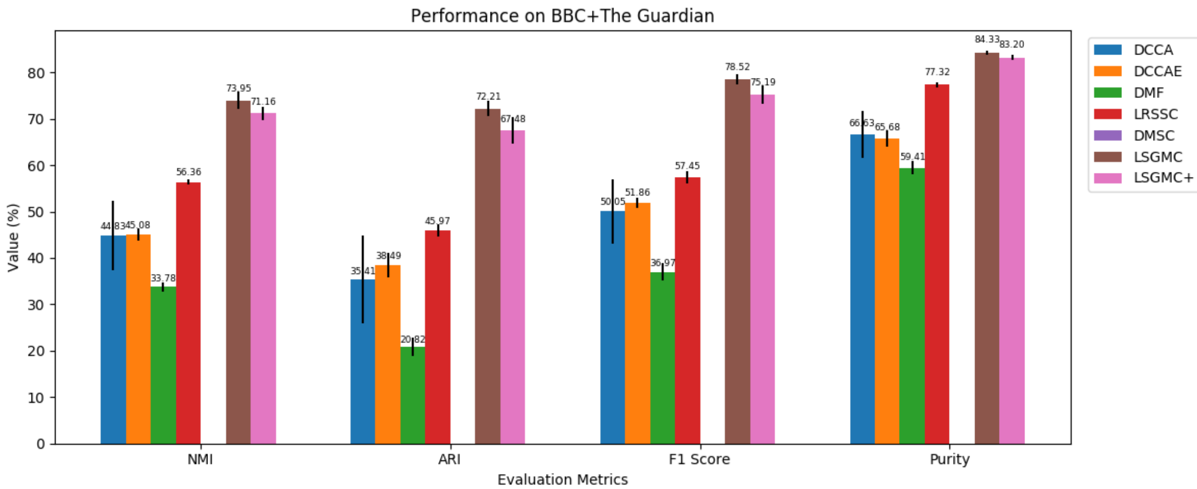


Figure 6.3: Performance on **BBC+The Guardian** dataset.

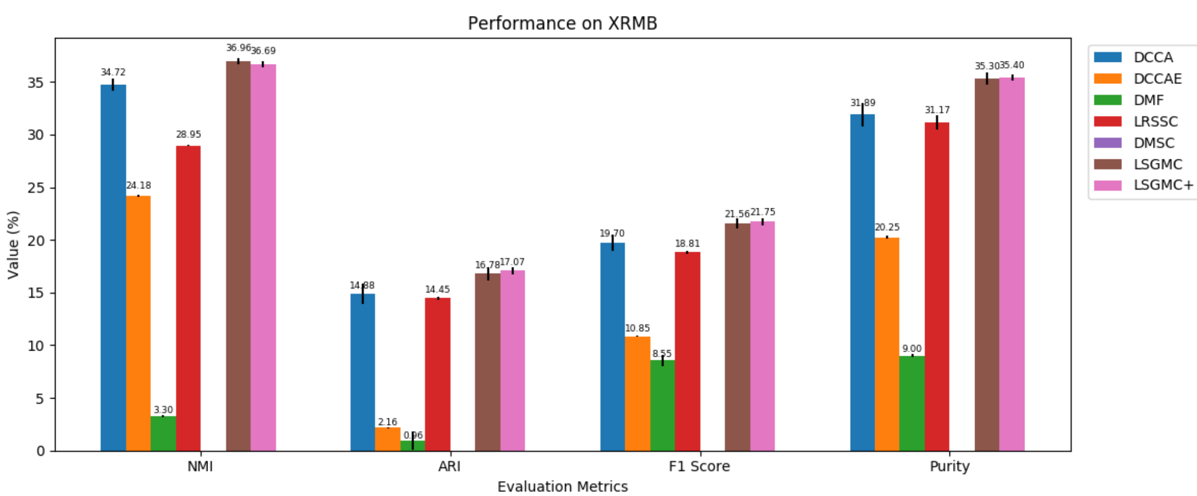
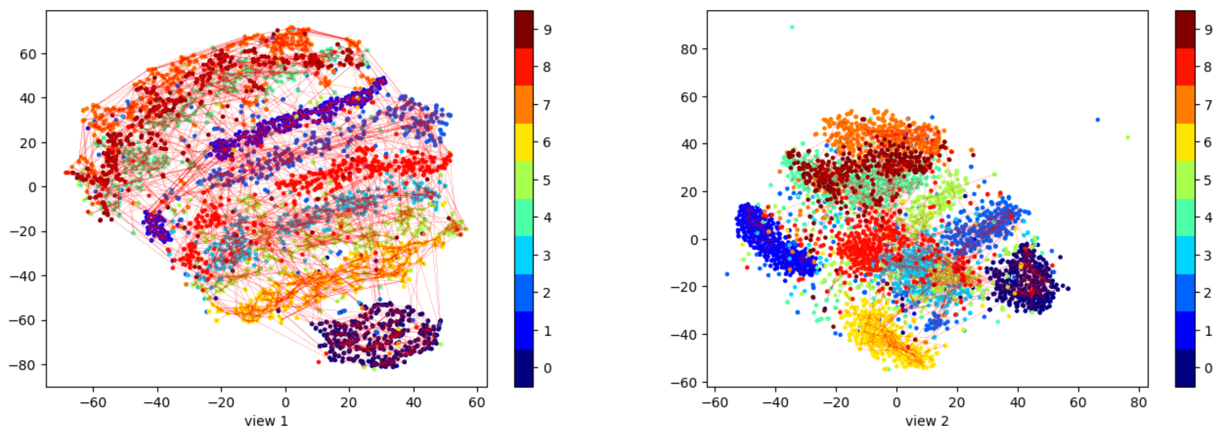


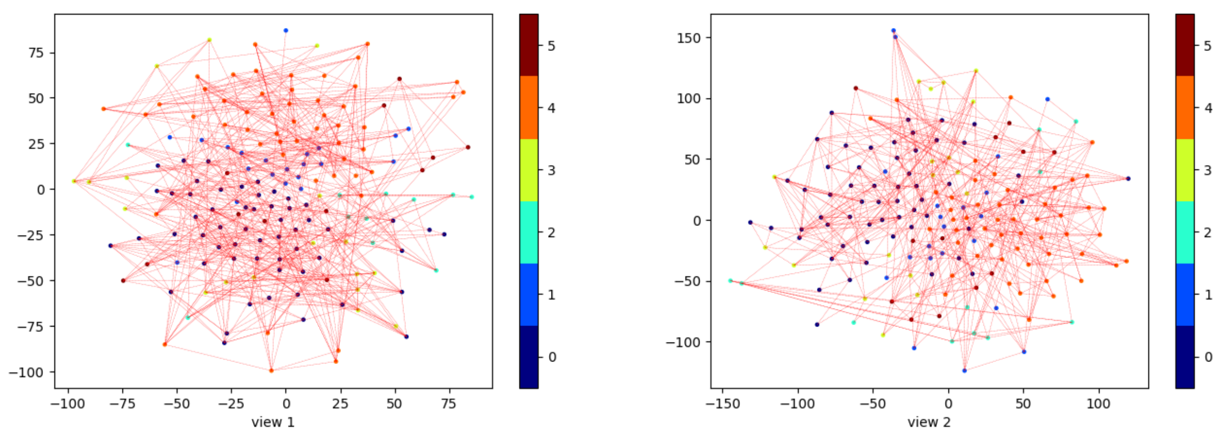
Figure 6.4: Performance on **XRMB** dataset.

The results comparing our LSGMC and LSGMC+ methods are inconclusive. LSGMC+, which takes into account additional consensus information, slightly outperforms LSGMC on the Noisy MNIST data, but not significantly. LSGMC+ has the same performance as LSGMC on Digit/MNIST-USPS and XRMB dataset, and performs slightly worse on the small BBC+The Guardian dataset.

6.2 Visualization of Local Similarity Graphs



(a) NoisyMNIST



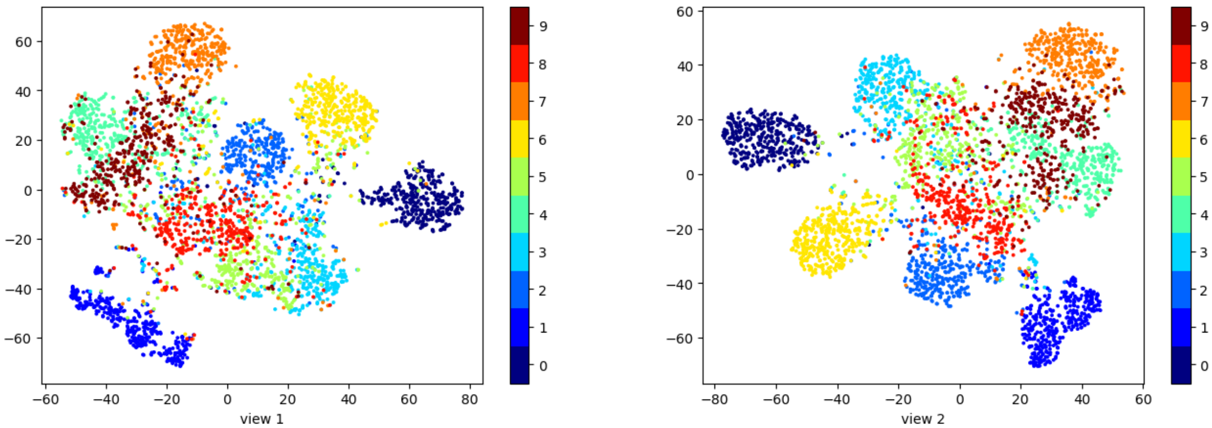
(b) BBC+The Guardian

Figure 6.5: Visualization of edges constructed by Mutual K nearest graphs in each view on the two components of t-SNE embeddings of the original data.

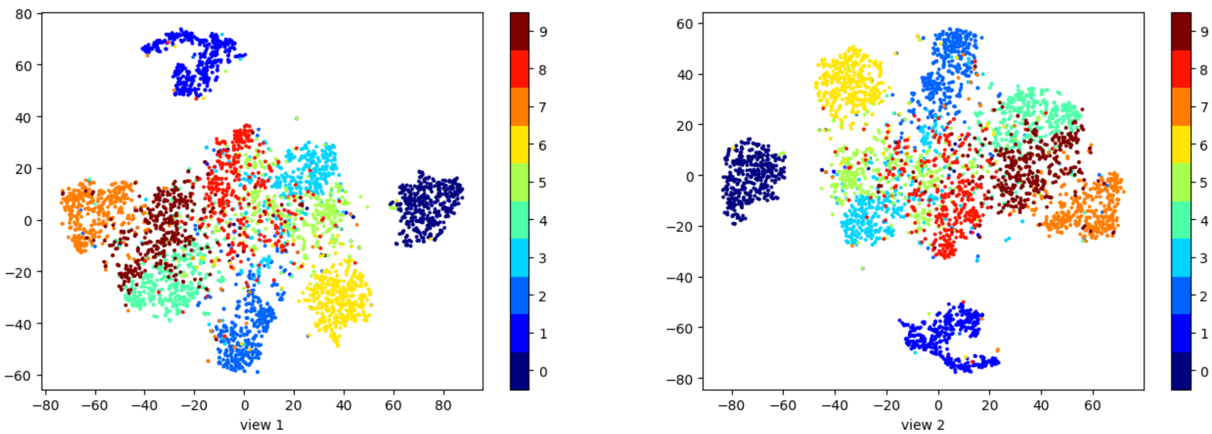
To better understand the local similarity graphs constructed by Mutual K nearest neighbors (MKNN), in **Figure** 6.5, we plot the view specific MKNN graphs, on two components of t-SNE embeddings of the original data, on the Noisy MNIST and BBC+The Guardian datasest. Each dot in the plot represents a data point in the original space, and each red dashed line represents an edge in the MKNN graph. We observe that the density of MKNN edges provide some information about the clusters.

6.3 Visualization of the Embeddings

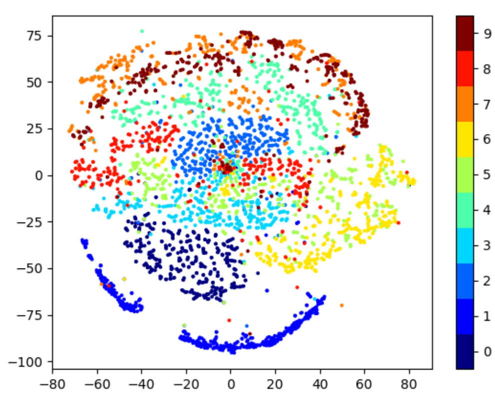
In **Figure** 6.6 and **Figure** 6.7 we visualize and compare t-SNE plots of the low dimensional embeddings learned on the Noisy MNIST and BBC+The Guardian dataset, where LSGMC greatly outperforms the other multi-view clustering methods. The plots strongly suggest that the representation learned by our proposed LSGMC is more separable. The t-SNE plots provide visual evidence suggesting that local connectivity graph can be a very useful signal in unsupervised learning on multi-view data.

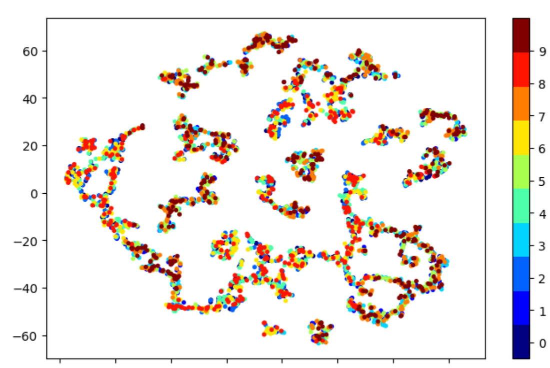


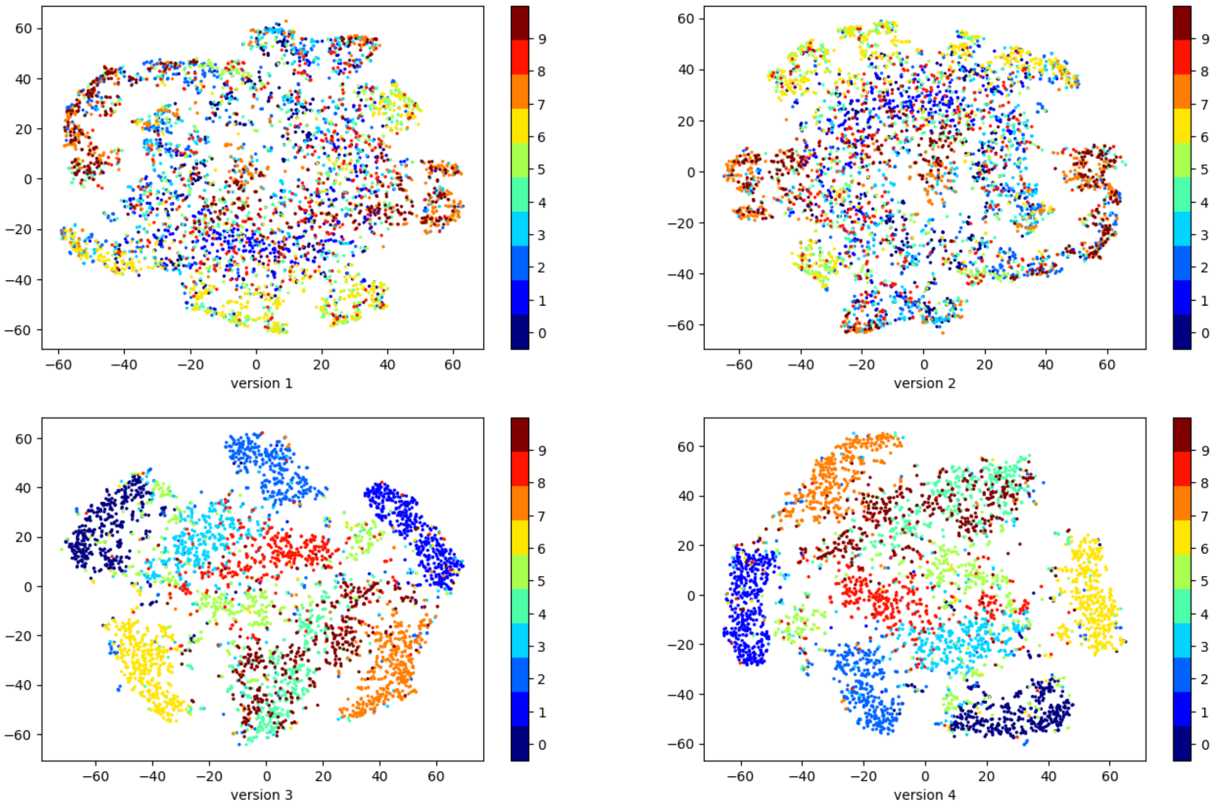
(a) DCCA (NMI: 60.56, ARI: 49.80, F1 score: 54.88, Purity: 70.79)



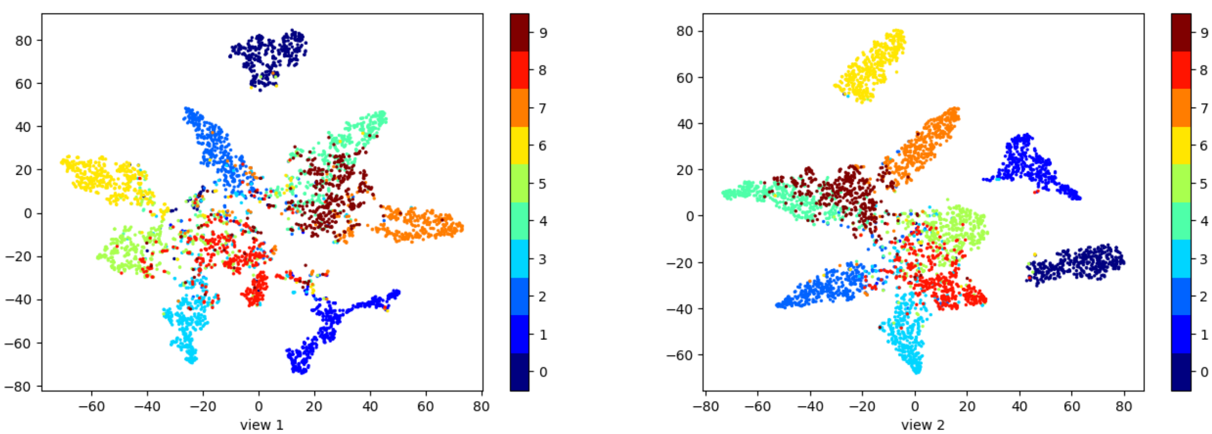
(b) DCCAE (NMI: 62.53, ARI: 52.88, F1 score: 57.72, Purity: 70.65)


(c) DMSC (NMI: 48.27, ARI: 28.98, F1 score: 37.52, Purity: 56.43)


(d) DMF (NMI: 15.16, ARI: 5.85, F1 score: 15.30, Purity: 18.47)

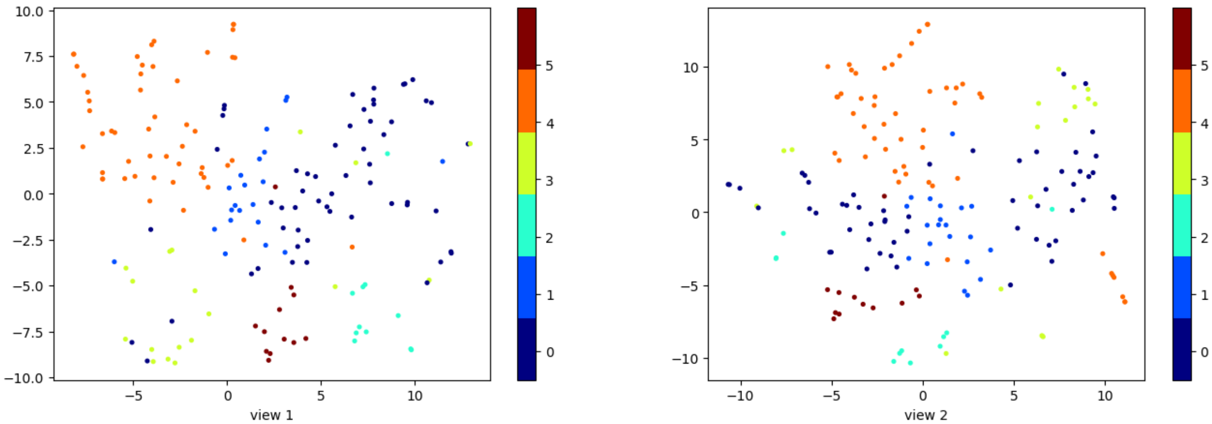


(e) LRSSC (highest: NMI: 47.73, ARI: 34.61, F1 score: 41.34, Purity: 57.25)

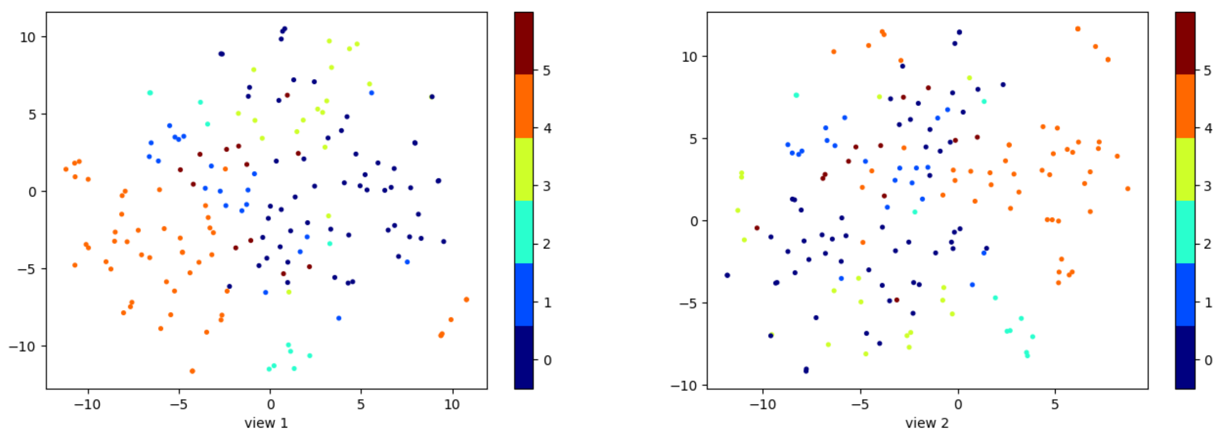


(f) LSGMC (NMI: 74.22, ARI: 61.94, F1 score: 65.90, Purity: 79.59)

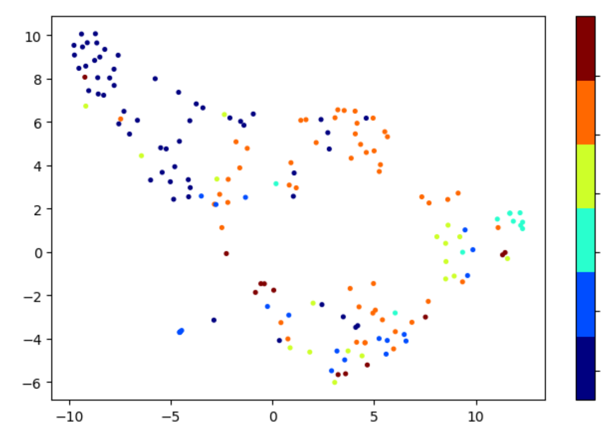
Figure 6.6: t-SNE plots of the learned data representation (embedding) on 4000 samples Noisy MNIST dataset (10 classes) by different multi-view clustering approaches. The embedding of view 1 is on the left and view 2 on the right. Note that DMSC applies fusion algorithms on the learned data representation from different views to gain a unified representation. We only plot the final data representation learned by DMSC.



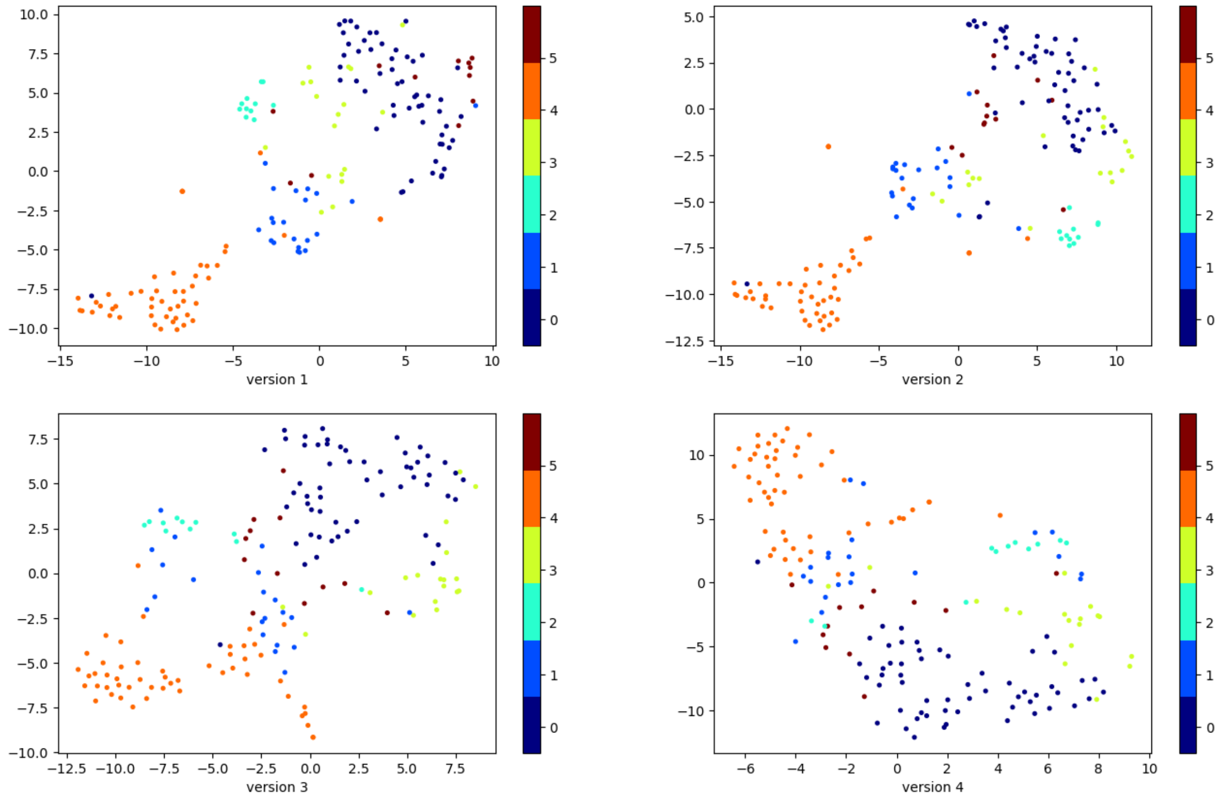
(a) DCCA (NMI: 44.83, ARI: 35.41, F1 score: 50.05, Purity: 66.63)



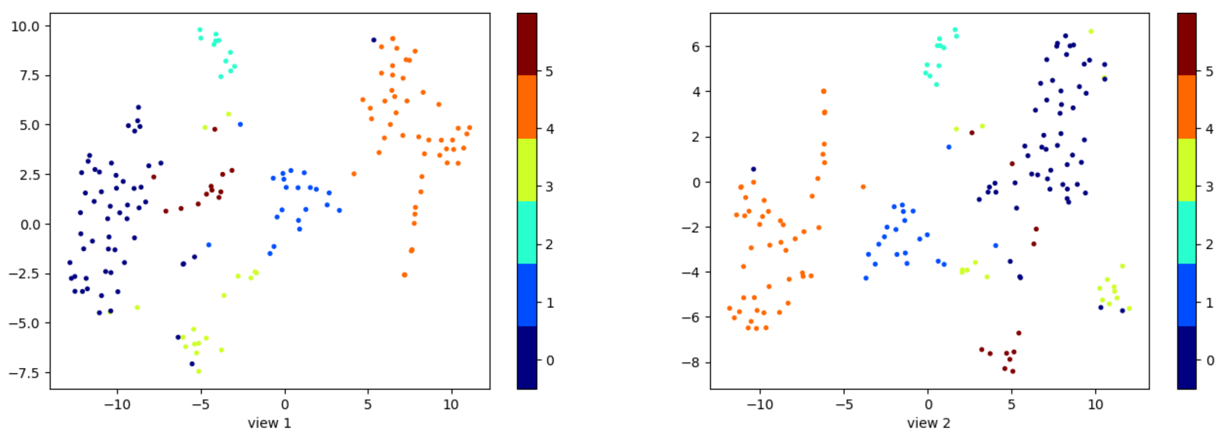
(b) DCCAE (NMI: 45.08, ARI: 38.49, F1 score: 51.86, Purity: 65.68)



(c) DMF (NMI: 33.78, ARI: 20.82, F1 score: 36.97, Purity: 59.41)



(d) LRSSC (highest: NMI: 56.36, ARI: 45.97, F1 score: 57.45, Purity: 77.32)



(e) LSGMC (NMI: 73.95, ARI: 72.21, F1 score: 78.52, Purity: 84.33)

Figure 6.7: t-SNE plots of the learned data representation (embedding) on 169 samples BBC+The Guardian dataset (6 classes) by different multi-view clustering approaches. The embedding of view 1 is on the left and view 2 on the right.

6.4 Discussion

Our experiments suggest that the proposed LSGMC outperforms state-of-the-art multi-view clustering alternatives consistently on popular image and text benchmark datasets as well as a real world acoustic-articulatory dataset. While correlations between data from different views can be powerful in regularizing view consistency, the guidance of local similarity graphs can significantly improve the performance of a clustering algorithm. This is evidenced by the performance of LSGMC over related correlation based multi-view clustering methods, DCCA and DCCAE.

We report three characteristics of the local similarity graphs on each dataset in **Table** 6.1. To illustrate the three characteristics, consider an example in **Figure** 6.8. In the example, we plot 5 synthetic data samples, where the two magenta points represent samples in class 0 while the other three cyan points represent samples from class 1. We also plot all edges in the graph across 5 data samples. We use red edges to denote the ones selected by MKNN and blue edges to denote the non-selected ones.

We first consider the percentage of edges (% Total) selected by MKNN among all possible edges in a graph. In the toy example, this corresponds to $\frac{6}{10}$ since there are 6 red edges among a total of 10 edges. We then report the number of true/correct edges, whose incident nodes belong to the same cluster, among the selected edges by MKNN (# True). In this example, edges $\{$(a), (d), (e), (f)$\}$ are the correct ones and thus the number of correct edges is 4 in this case. We also report the percentage of true edges (% True) among all edges selected by MKNN, which is 4 among 6 edges in the example.

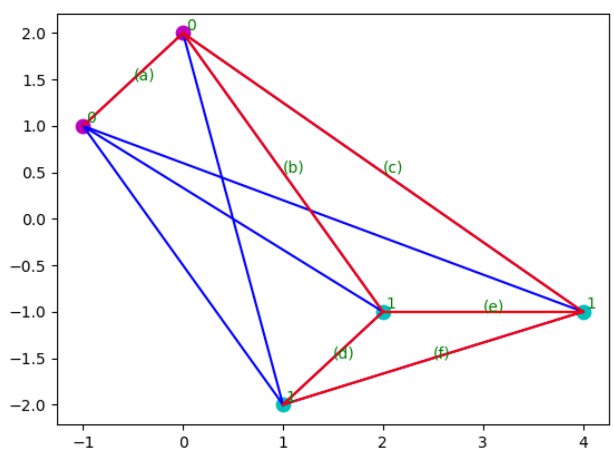


Figure 6.8: An example illustrating three characteristics of the local similarity graphs we report on each dataset. The two magenta points represent samples from class 0 while the other three cyan points represent samples from class 1. We plot a complete graph, where the red edges are the ones selected by MKNN graph while the blue edges are the ones not selected.

Dataset	View 1			View 2			Unified			Common		
	%Total	#True	%True	%Total	#True	%True	%Total	#True	%True	%Total	#True	%True
Noisy MNIST	0.14	9765	88.62	0.07	3773	67.29	0.21	13498	81.39	5e-4	40	95.24
Digit	0.14	10669	95.18	0.13	9764	93.17	0.27	20291	94.17	1.8e-3	142	100.00
BBC+Gua	4.09	469	80.72	3.68	424	81.07	5.78	653	79.53	1.99	240	84.81
XRMB	0.027	31012	57.34	0.026	27323	53.42	0.051	55227	54.21	0.0017	3108	92.50

Table 6.1: Characterization of local similarity graphs from view 1, view 2, unified graph and graph with common edges on different datasets. %Total, #True and %True denote the percentage of the number of edges in the graph of interest among all edges in a complete graph, the total number of correct edges in the graph of interest and the percentage of correct edges in the graph of interest.

We report the above three characteristics of four graphs: the local similarity graphs from view 1, view 2, the unified graph and the common edges in **Table** 6.1. We observe that the percentage of correct edges of unified graphs is $\gtrsim 80\%$ on each dataset except XRMB on which the percentage of correct edges is only 54%, while covering only a small portion of all possible edges. We further observe that the number of edges selected by MKNN is extremely small. The results demonstrate that a small amount of information from the local similarity graphs with a reasonable quality can be used as a powerful training signal to guide the search for a better embedding space. Our experiments and results demonstrate that this is indeed the case for several different data types and common multi-view benchmark datasets. We did not tune the number of neighbors for each dataset to improve the quality of the graphs but rather chose a fixed value in a range suggested by the analysis presented in [8].

Additionally, the characteristics of local similarity graphs provide insights into the performance difference between LSGMC and LSGMC+ on different datasets.

On Noisy MNIST, the quality of view 2's local similarity is low, with only 67.29% correct edges, compared to view 1's local similarity graph, with 88.62% correct edges. As a result, the unified graph has only 81.39% correct edges, while there are 95.24% correct edges among the common edges between view 1 and view 2. On XRMB, the quality of both view 1 and view 2's local similarity is low, with only 57.34% and 53.42% correct edges respectively. The unified graph has only 54.21% correct edges while there are 92.50% correct common edges. Since LSGMC uses the unified graph to learn the embedding for both views, the common edge graph with high correctness could provide valuable additional information to the learning process. This explains why we observe > 1% increase in ARI and F1 score in LSGMC+ on Noisy MNIST, and a slight increase in ARI, F1 score and Purity in LSGMC+ on XRMB, even if we only use the information of 20% common edges selected uniformly randomly every 5 epochs in training. This case corresponds to the general belief on consensus that common edges could provide more reliable training signals.

On the other two datasets, however, the difference between the percentage of correct edges in the unified graph and the graph with common edges is small, with 5.83% and 5.28% on Digit/MNIST-USPS and BBC+The Guardian respectively. Common edges do not provide much more information than edges in the unified graph. The slight drop in performance of LSGMC+ on the two datasets may be explained by the fact that the algorithm is biased towards a small subset of common edges despite them not being much more accurate than other edges. This case is contrary to the belief that common edges could provide more reliable training signals. In conclusion, whether a graph with common edges across views should be used as an additional training signal to the unified graph, depends on the inherent local structure of the data.

6.5 Extension to Semi-supervised Clustering

We demonstrate the use of LSGMC in the semi-supervised clustering setting and demonstrate the flexibility it offers in incorporating pairwise *must-link* and *cannot-link* constraints, on the Noisy MNIST dataset.

We set up the experiment by first constructing a unified MKNN graph on all available data samples, and split our data samples into a training set and a testing set. We then incorporate *must-link* or *cannot-link* constraints to the training set only and evaluate the performance on the testing set. To illustrate the experiment setting, consider an example in **Figure** 6.9. We construct the MKNN graph based on all samples, as shown by red edges in the plot. In the example, the training set consists of data samples $\{$(a), (b), (h), (i)$\}$. We add *must-link* constraints through subsampling a small amount of true edges across data samples in the training set. In the example, we could add the edge connecting nodes $\{$(a), (b)$\}$, or the edge connecting nodes $\{$(h), (i)$\}$. We add *cannot-link* constraints through subsampling a small amount of false edges among training data samples in the ones selected by MKNN graph and remove them. In this example, we could remove the edge connecting nodes $\{$(a), (i)$\}$.

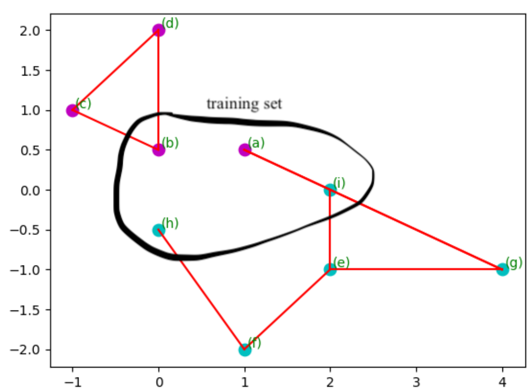


Figure 6.9: An example illustrating the experiment setting for evaluating semi-supervised clustering. The magenta and cyan points represent data samples from two classes. The red edges represent edges in the MKNN graph. Points in the black circle represent samples in the training set and the rest of the points represent samples in the testing set.

In the experiment, we consider {50% training - 50% testing, 60% training - 40% testing, 70% training - 30% testing} split of the entire data samples. As in the unsupervised setting, we conduct 5 random runs of each experiment and report the mean and standard deviation of values achieved under each metric across 5 runs.

For pairwise *must-link* constraints, we randomly select {500, 1000, 5000, 10000} pairs of samples within the training set which belong to the same cluster and which do not appear in the unified local similarity graph.

For pairwise *cannot-link* constraints, we randomly remove {100, 500, 1000} pairs from the unified similarity graph within the training set which belong to different clusters.

The results for incorporating pairwise *must-link* constraints are presented in **Figure** 6.10, 6.11 and 6.12. The results for removing pairwise *cannot-link* constraints are presented in **Figure** 6.13, 6.14 and 6.15.

We observe a slight increase across all performance metrics in most cases, as the number of *must-link* or *cannot-link* constraints increase. However, it is not necessary that as the number of pairwise constraints increases, the performance will increase for sure. Sometimes there might even be a slight drop in performance when the number of pairwise constraints increases. For example, on 50% training - 50% testing split, incorporating more pairwise constraints will not necessarily increase the performance across all metrics. The performance achieves the maximum value when we add 1000 pairs of *must-link* constraints and the performance drops when we add 5000 and 10000 *must-link* constraints. On 60% training - 40% testing split, the performance drops slightly to be below the baseline as we use 100 *cannot-link* constraints across three metrics: ARI, F1 score and Purity. However, in most cases, incorporating pairwise constraints performs no worse than the baseline or slightly outperforms the baseline.

This shows LSGMC can make use of pairwise information about the data, when such information is available to some extent. However, compared to unsupervised setting, LSGMC only has a slight advantage by incorporating pairwise constraints.

We have presented results from the experiments and have shown empirically that LSGMC can outperform several state-of-the-art multi-view clustering alternatives on a variety of datasets. We further show the flexibility of LSGMC in incorporating pairwise constraints to some extent. In the next chapter, we conclude the thesis work and pose several future directions.

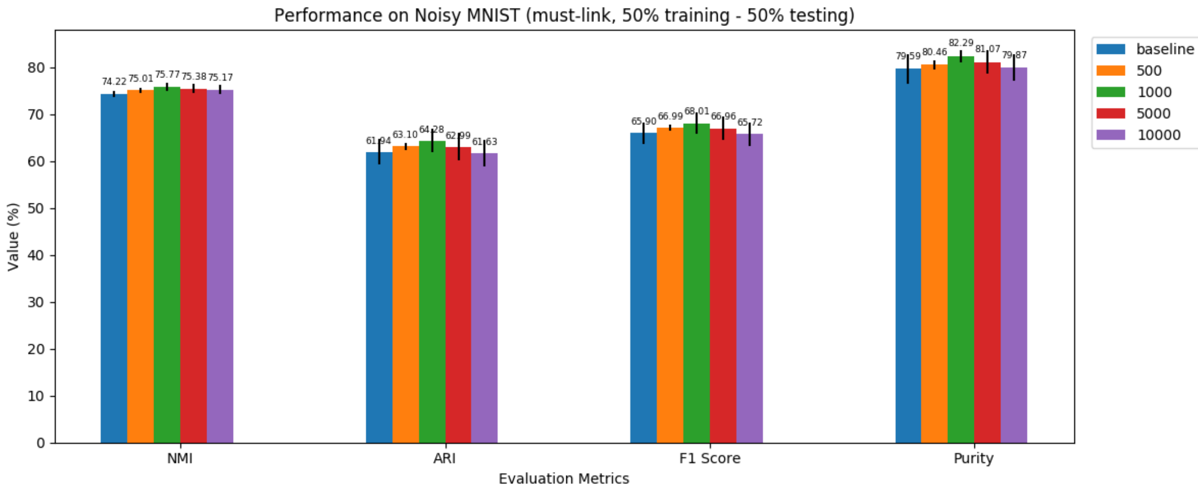


Figure 6.10: Performance on **Noisy MNIST** dataset with *must-link* constraints, 50% training data, 50% testing data.

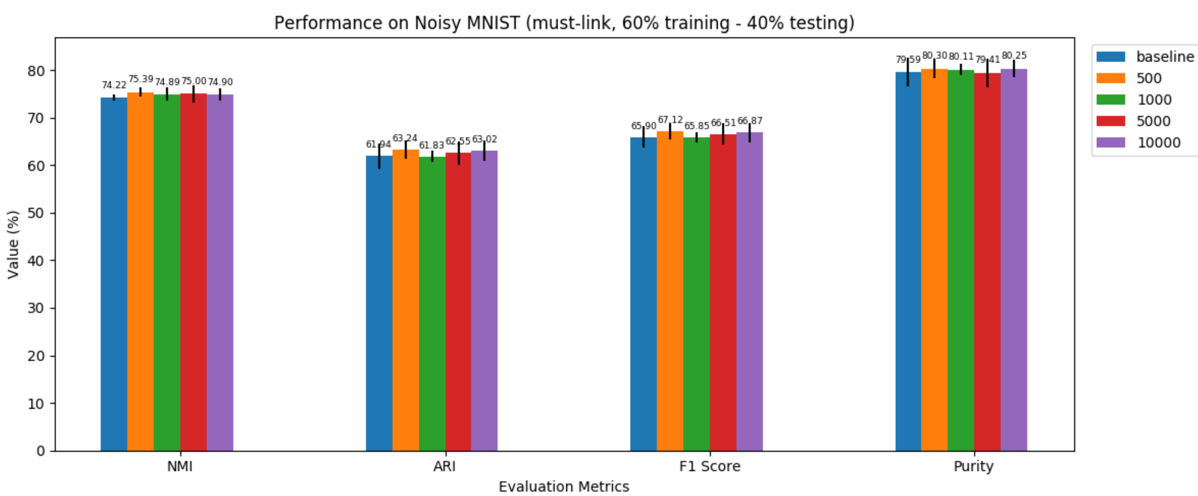


Figure 6.11: Performance on **Noisy MNIST** dataset with *must-link* constraints, 60% training data, 40% testing data.

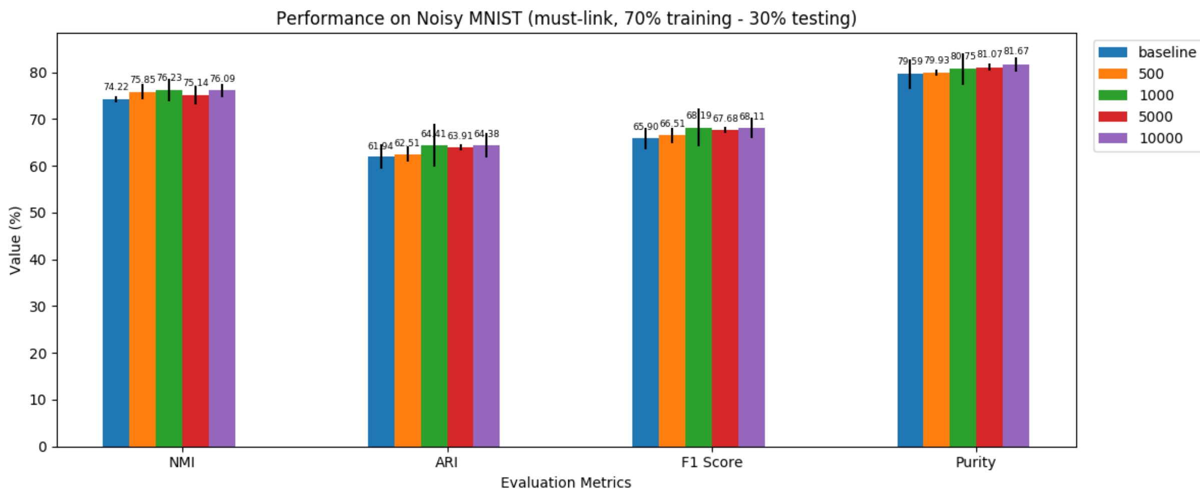


Figure 6.12: Performance on **Noisy MNIST** dataset with *must-link* constraints, 70% training data, 30% testing data.

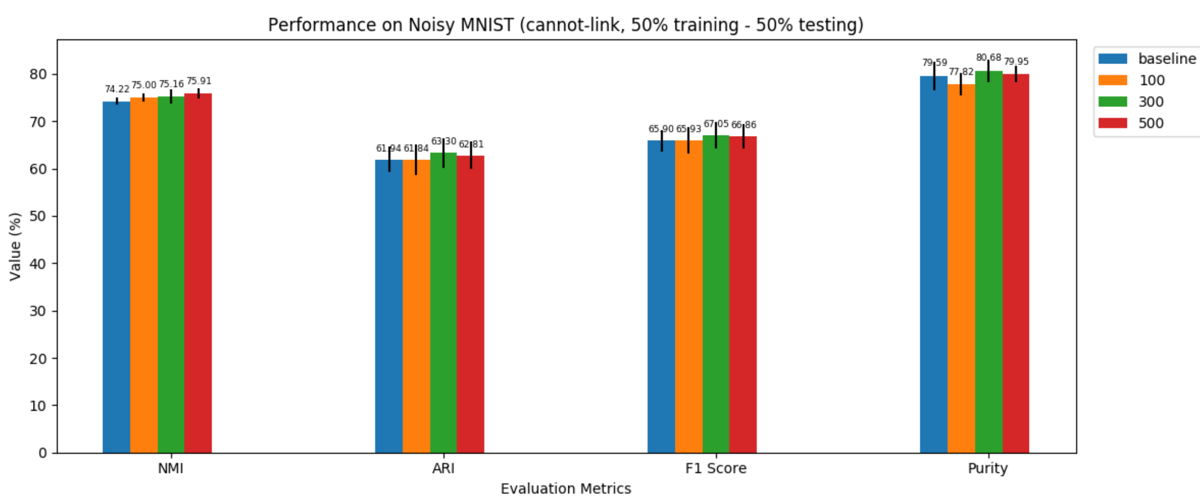


Figure 6.13: Performance on **Noisy MNIST** dataset with *cannot-link* constraints, 50% training data, 50% testing data.

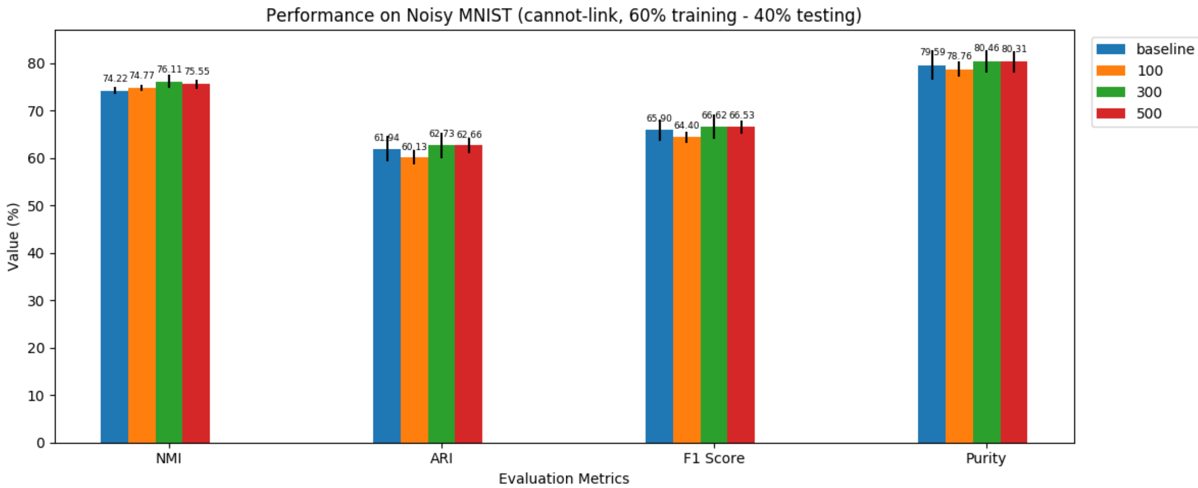


Figure 6.14: Performance on **Noisy MNIST** dataset with *cannot-link* constraints, 60% training data, 40% testing data.

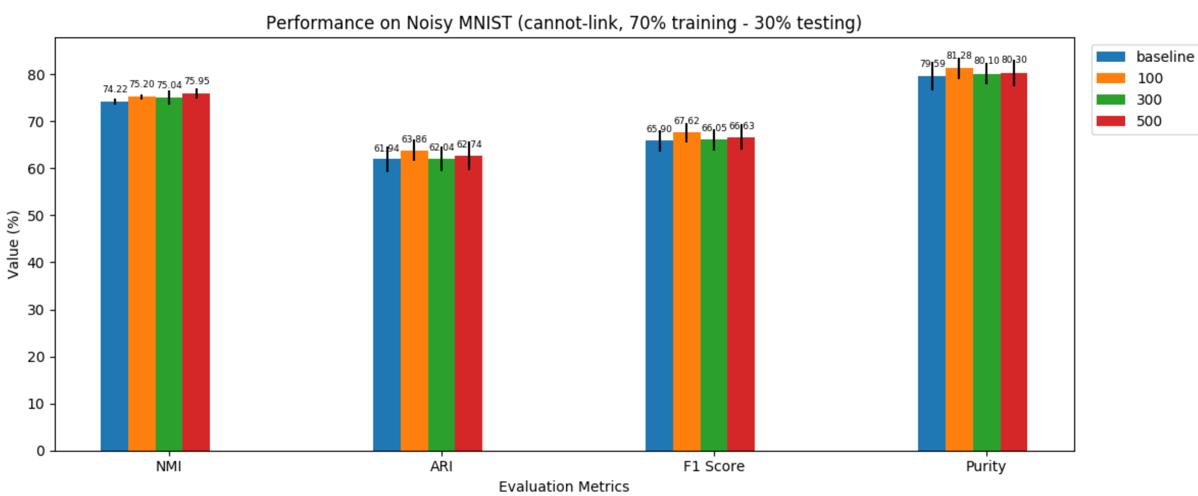


Figure 6.15: Performance on **Noisy MNIST** dataset with *cannot-link* constraints, 70% training data, 30% testing data.

Chapter 7

Conclusion

In this chapter, we first summarize the thesis work and discuss implications of LSGMC. We further discuss several limitations of our current work, how we could potentially address those limitations and propose future directions.

7.1 Conclusion

In this thesis, we study an important data mining task – clustering of multi-view data – which often arises naturally in many application domains. It has been shown that sometimes artificially splitting features to create multi-view data can improve the performance. Simply concatenating data from different views, which could come from very different distributions, in order to convert multi-view clustering into single-view clustering may exacerbate the risk of over-fitting and further diminish the interpretation of the resulting models. Thus developing multi-view clustering approaches which leverage data from various views to improve clustering performance could be important and beneficial for many applications.

We observe that the use of local similarity graphs and the flexibility of incorporating sometimes available *must-link* and *cannot-link* constraints, often referred to as semi-supervised clustering, are well studied in single-view clustering but under-explored in multi-view clustering. In this thesis, we present LSGMC, an improved correlation based multi-view clustering method which explores first order proximity within each view and complementary information across views. The proposed approach does so by using a unified graph based on local similarity graphs from each view. The informative local similarities can be a powerful learning signal in unsupervised methods and LSGMC unifies this information with correlation-based representation learning. LSGMC also allows for flexibility in incorporating pairwise constraints and can thus be easily extended to semi-supervised clustering.

Results from experiments presented in this thesis suggest that LSGMC is able to leverage extremely sparse local similarity information to improve clustering performance and outperforms a large number of existing state-of-the-art multi-view clustering approaches on image, text and acoustic-articulatory datasets. Results from experiments further show that LSGMC is able to incorporate pairwise information to slightly improve the learned data representation when such information is available.

LSGMC demonstrates the usefulness of local similarity graphs in multi-view clustering, even though such training signal is extremely sparse. This implies we might be able to combine other view-regularization techniques – for example, instead of using CCA to find directions that the correlation between the two embeddings of two views is maximized, we could co-regularize view embeddings into a unified representation – with training signals from view specific local similarity graphs. Those view specific local similarity graphs provide complement information across views. By incorporating this piece of information, we could potentially improve a broad class of existing multi-view clustering approaches.

7.2 Future Work

Our current work on LSGMC has several limitations. We discuss the implication of those limitations in practice and how we could potentially address these limitations as directions for future work.

1. LSGMC is unable to handle data with more than two views due to the limitation of CCA. This means we are currently not able to apply LSGMC on data with more than two views, which limits the application of LSGMC. We might want to consider multi-view CCA, which finds linear projections between every pair of views (e.g. [33]), or attempt other view regularization techniques (e.g. regularizing views towards a common representation).
2. LSGMC is only able to remove *cannot-link* constraints from the unified local similarity graphs and thus cannot handle such constraints if the links are not present in the graph under semi-supervised learning. We might want to add additional regularization terms to the objective function to force margins between pair of samples with *cannot-link* constraints. For example, we might want to penalize some *cannot-link* pair if the distance between the two data samples is below some threshold.
3. The experiments show that LSGMC has a large advantage over other multi-view clustering methods in unsupervised setting but has only a slight advantage in semi-supervised setting when we incorporate pairwise constraints. The current LSGMC is more suitable for unsupervised clustering but not semi-supervised clustering. This means that directly augmenting the local similarity graph might not be an efficient way of propagating pairwise information across all clusters. We might want to consider other ways of propagating such information in semi-supervised literature and compare against other semi-supervised clustering methods.
4. The quality of local similarity graphs could have a large impact on the clustering performance. The more accurate (i.e. correct number of pairs) the local similarity graph is and the more extensive the information (i.e. pairwise information indicated by edges across various data samples) the local similarity graph contains, the better LSGMC is able to perform. In an actual unsupervised clustering task, we would have no idea how good

the quality of the constructed local similarity graph is and how much improvement in the performance LSGMC is able to achieve. Therefore, we might want to pre-determine the quality of such local similarity graphs and predict how well LSGMC could perform to see whether LSGMC is a suitable multi-view clustering approach on certain datasets. We might want to check whether there is a correlation between some graph properties of the local similarity graphs, e.g. eigenvalues of the graph Laplacian, and the performance of LSGMC.

Bibliography

[1] Mahdi Abavisani and Vishal M. Patel. Deep multimodal subspace clustering networks. In *IEEE Journal of Selected Topics in Signal Processing*, volume 12, pages 1601–1614, 2018. doi: 10.1109/JSTSP.2018.2875385. 2.3, 5.2.5

[2] Mohamed Abbas and Amin Shoukry. Cmune: A clustering using mutual nearest neighbors algorithm. In *the 11th International Conference on Information Science, Signal Processing and their Applications*, ISSPA '12, pages 1192–1197, 07 2012. ISBN 978-1-4673-0381-1. doi: 10.1109/ISSPA.2012.6310472. 1, 2.4

[3] Ahmed N. Albatineh, Magdalena Niewiadomska-Bugaj, and Daniel Mihalko. On similarity indices and correction for chance agreement. *Journal of Classification*, 23:301–313, 02 2006. doi: 10.1007/s00357-006-0017-z. 5.4.2, 5.4.2

[4] Enrique Amigó, Julio Gonzalo, Javier Artiles, and M. Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12:461–486, 10 2009. doi: 10.1007/s10791-008-9066-8. 5.4

[5] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *ICML '13*, pages 1247–1255, Atlanta, Georgia, USA, 06 2013. 2.3, 3.4, 4.1, 4.2, 5.2.1

[6] Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 59–68, New York, NY, USA, 2004. ISBN 1581138881. doi: 10.1145/1014052.1014062. 2

[7] Maria Brbić and Ivica Kopriva. Multi-view low-rank sparse subspace clustering. *Pattern Recognition*, 73:247–258, 2018. doi: https://doi.org/10.1016/j.patcog.2017.08.024. 5.1.3, 5.2.4

[8] MR Brito, EL Chavez, AJ Quiroz, and JE Yukich. Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Statistics & Probability Letters*, 35(1): 33–42, 08 1997. 4.3, 6.4

[9] Xiaochun Cao, Fu Huazhu Zhang, Changqing, Si Liu, and Hua Zhang. Diversity-induced multi-view subspace clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '15, pages 586–594, June 2015. doi: 10.1109/CVPR.2015.7298657. 2.3

[10] Guoqing Chao, Shiliang Sun, and Jinbo Bi. A survey on multi-view clustering. 2017. URL

http://arxiv.org/abs/1712.06246. 1.1, 2.2

[11] Sharon Fogel, Hadar Averbuch-Elor, Jacob Goldberger, and Danielr Cohen-Or. Clustering-driven deep embedding with pairwise constraints. *IEEE Computer Graphics and Applications*, 39(04):16–27, 07 2019. ISSN 1558-1756. doi: 10.1109/MCG.2018.2881524. 1, 2, 2.1, 2.4, 3.2, 4.1

[12] Hongchang Gao, Feiping Nie, Xuelong Li, and Heng Huang. Multi-view subspace clustering. In *IEEE International Conference on Computer Vision*, pages 4238–4246, 12 2015. doi: 10.1109/ICCV.2015.482. 2.3

[13] Zhenyu Huang, Joey Tianyi Zhou, Xi Peng, Changqing Zhang, Hongyuan Zhu, and Jiancheng Lv. Multi-view spectral clustering network. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI '19, pages 2563–2569, 07 2019. doi: 10.24963/ijcai.2019/356. 2.3

[14] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1): 193–218, 1985. 5.4.2, 5.4.2

[15] Yangbangyan Jiang, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Dm2c: Deep mixed-modal clustering. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 5888–5892. Curran Associates, Inc., 2019. 4.1

[16] Abhishek Kumar and Hal Daume III. A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML '11, page 393–400, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195. 2.3

[17] Abhishek Kumar, Piyush Rai, and Hal Daumé III. Co-regularized multi-view spectral clustering. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NeurIPS '11, page 1413–1421, Red Hook, NY, USA, 2011. Curran Associates Inc. ISBN 9781618395993. 2.3

[18] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5.1.1

[19] Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, ICDM '13, pages 252–260, 2013. doi: 10.1137/1.9781611972832.28. 2.3

[20] Małgorzata Lucińska and Sławomir Wierzchoń. Spectral clustering based on k-nearest neighbor graph. In *the 11th International Conference on Computer Information Systems and Industrial Management*, CISIM '12, pages 254–265, 09 2012. 1, 2.4

[21] Shirui Luo, Changqing Zhang, Wei Zhang, and Xiaochun Cao. Consistent and specific multi-view subspace clustering. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, AAAI '18, 2018. 2.3

[22] Markus Maier, Matthias Hein, and Ulrike von Luxburg. Optimal construction of k-nearest neighbor graphs for identifying noisy clusters. In *Theoretical Computer Science*, page 1749–1764. Elsevier, 04 2009. 5.3

[23] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduc-*

tion to Information Retrieval. Cambridge University Press, Cambridge, UK, 2008. ISBN 978-0-521-86571-5. URL `http://nlp.stanford.edu/IR-book/information-retrieval-book.html`. 5.4

[24] Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the 9th International Conference on Information and Knowledge Management*, CIKM '00, page 86–93, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581133200. doi: 10.1145/354756.354805. 1.1

[25] Dan Pelleg and Dorit Baras. K-means with large and noisy constraint sets. In *Proceedings of the 18th European conference on Machine Learning*, ECML '07, pages 674–682, Berlin, Heidelberg, 09 2007. Springer Berlin Heidelberg. ISBN 978-3-540-74958-5. doi: 10.1007/978-3-540-74958-5_67. 2

[26] Xi Peng, Zhenyu Huang, Jiancheng Lv, Hongyuan Zhu, and Joey Tianyi Zhou. COMIC: Multi-view clustering without parameter selection. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *ICML '19*, pages 5092–5101, Long Beach, California, USA, 06 2019. PMLR. 5.1.2

[27] Mingjie Qian and Chengxiang Zhai. Unsupervised feature selection for multi-view clustering on text-image web news data. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, page 1963–1966, New York, NY, USA, 2014. ISBN 9781450325981. doi: 10.1145/2661829.2661993. 1.1

[28] Divya Sardana and Raj Bhatnagar. Graph clustering using mutual k-nearest neighbors. pages 35–48. Springer International Publishing, 08 2014. ISBN 978-3-319-09911-8. doi: 10.1007/978-3-319-09912-5_4. 1, 2.4

[29] Sohil Atul Shah and Vladlen Koltun. Robust continuous clustering. *Proceedings of the National Academy of Sciences*, 114(37):9814–9819, 2017. ISSN 0027-8424. doi: 10.1073/pnas.1700770114. 1, 2, 2.1, 2.4, 3.2, 4.6

[30] Xiaoliang Tang, Xuan Tang, Wanli Wang, Li Fang, and Xian Wei. Deep multi-view sparse subspace clustering. In *Proceedings of the 2018 VII International Conference on Network, Communication and Computing*, ICNCC '18, page 115–119, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450365536. doi: 10.1145/3301326.3301391. 2.3

[31] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, volume 37 of *ICML '15*, page 1083–1092, 2015. 2.3, 3.4, 4.1, 4.6, 5.1.1, 5.2.2

[32] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '15, pages 4590–4594, 04 2015. doi: 10.1109/ICASSP.2015.7178840. 5.1.4

[33] Daniela Witten and Robert Tibshirani. Extensions of sparse canonical correlation analysis with applications to genomic data. In *Statistical applications in genetics and molecular biology*. Berkeley Electronic Press, 06 2009. doi: 10.2202/1544-6115.1470. 1

[34] Rongkai Xia, Yan Pan, Lei Du, and Jian Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, AAAI '14, page 2149–2155, 2014. 2.3

[35] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, volume 48 of *ICML'16*, page 478–487. JMLR.org, 2016. 4.1

[36] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. 2013. URL `http://arxiv.org/abs/1304.5634`. 1.1, 2.2

[37] Nan Xu, Yanqing Guo, Jiujun Wang, Xiangyang Luo, and Xiangwei Kong. Multi-view clustering via simultaneously learning shared subspace and affinity matrix. *International Journal of Advanced Robotic Systems*, 14(6):1729881417745677, 2017. doi: 10.1177/1729881417745677. 2.3

[38] Changqing Zhang, Qinghua Hu, Huazhu Fu, Pengfei Zhu, and Xiaochun Cao. Latent multi-view subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '17, pages 4333–4341, 2017. 2.3

[39] Dejiao Zhang, Yifan Sun, Brian Eriksson, and Laura Balzano. Deep unsupervised clustering using mixture of autoencoders. 12 2017. URL `http://arxiv.org/abs/1712.07788`. 2.3, 4.1

[40] Zheng Zhang, Li Liu, Fumin Shen, Heng Tao Shen, and Ling Shao. Binary multi-view clustering. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, number 7, 2019. 2.3

[41] Handong Zhao, Zhengming Ding, and Yun Fu. Multi-view clustering via deep matrix factorization. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, AAAI '17, 2017. 2.3, 5.2.3