Order N^o : ../../..

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
UNIVERSITY KASDI MERBAH OUARGLA

Faculty of New Information Technologies and Communication
Department of Electronics and Telecommunications

THESIS

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

3^{rd} **Cycle LMD Doctorate**

Option : **Telecommunications Systems**

By : **Khamis HOUFAR**

Theme

Collaborative Representation Learning From Multiple Image Descriptors : Algorithms and Applications to Visual Data Analysis

Publicly defended on : *May* $25^{th}, 2023$ before the jury composed of :

Mourad CHAA	*MCA*	*at University Ouargla*	*President*
Djamel SAMAI	*MCA*	*at Ouargla University*	*Thesis Director*
Abdelmalik TALEB-AHMED	*Pr*	*at Valenciennes University*	*Thesis Co-director*
Riadh AJGOU	*Pr*	*at Eloued University*	*Examiner*
Abderrazak BENCHABANE	*MCA*	*at Ouargla University*	*Examiner*
Abdelhai LATI	*MCA*	*at Ouargla University*	*Examiner*

Quote : “Clustering is in the eye of the beholder”

هُوَ ٱلَّذِي جَعَلَ ٱلشَّمۡسَ ضِيَآءٗ وَٱلۡقَمَرَ نُورٗا وَقَدَّرَهُۥ مَنَازِلَ لِتَعۡلَمُواْ عَدَدَ ٱلسِّنِينَ وَٱلۡحِسَابَۚ مَا خَلَقَ ٱللَّهُ ذَٰلِكَ إِلَّا بِٱلۡحَقِّۚ يُفَصِّلُ ٱلۡأٓيَٰتِ لِقَوۡمٖ يَعۡلَمُونَ

It is He who made the sun a shining light and the moon a derived light and determined for it phases – that you may know the number of years and account [of time]. Allah has not created this except in truth. He details the signs for a people who know

With great happiness and gratitude, I dedicate this thesis to:

My lovely parents **Fatiha** *and* **Saad**, *who always take care of me and support me through their prayers.*

"My dad passed away on February 28th, 2023; May Allah rest his soul and grant him the highest level of Paradise."

My beloved brothers and sister, **Ali,Ramzi,Saida,** *and* **Mohammed.**

To my honey wife **Toumia**, *and my sweets* **Amira** *and* **Saad.**

To all my big family members, may Allah bless you all , and keep you safe, and full of happiness.

All the beloved MOBILIS employees and my friends who always support me and bring me happiness.

Khamis HOUFAR

Acknowledgments

First and foremost, I thank **ALLAH** the Most Gracious and Most Merciful for giving me the strength and ability to complete this study.

Furthermore, I would like to thank from my heart my best advisors, ***Dr. Djamel Samai (University Kasdi Merbah of Ouargla), and Pr. Abdelmalik Taleb Ahmed (Université Polytechnique Hauts de France, Université de Lille, Valenciennes, France)*** *who have shared their experiences with me during my Ph.D. journey, as well as for their graciously unconditional support and everlasting attention.*

In particular, I would like to express my deepest gratitude and how fortunate I was to meet ***Dr. Fadi Donaika (University of the Basque Country UPV/EHU, Spain)****, the amazing doctor, for his assistance, professional advice, and his interminable support.*

I'm incredibly grateful to my best team members, ***"The twin-kings," Dr. Azeddine Benlamoudi, and Dr. Khalid Bensid (University Kasdi Merbah of Ouargla)****, for their patience and invaluable help and whom I had a pleasure to work with, in an excellent research atmosphere.*

I would like to acknowledge my thesis committee ***Dr. Mourad CHAA, Pr. Riadh AJGOU, Dr. Abderrazak BENCHABANE, and Dr. Abdelhai LATI****, for accepting and evaluating my thesis work.*

I would also like to express my gratitude and appreciation toward my generous teachers : ***Pr. Mohamed Lamine Kherfi, Dr. Oussama Aiadi, Dr. Khadra Bouanane, Dr. Maarouf Korichi, Dr. Mebarka Allaoui*** *who have directly or indirectly encouraged my research work in this thesis.*

To all my colleges and friends– Thank you so much.

Abstract

Multi-view is a kind of rich data thanks to the several features that are laid out in the form of multiple data matrices ; these matrices are obtained through different measurements, experiments, or transformations. Multi-view data analysis involves dealing with different features simultaneously to get the most out of their information and achieve a practical decision. In many real-life applications, multi-view data is naturally raised, particularly in computer vision and image analysis. Each feature representation corresponds to a view and may represent distinct formulations or statistical properties. It is agreed that multi-view data share common information as well as contain some complementary information ; the crucial step in machine learning and knowledge discovery is learning how these two understandings can be appropriately manipulated and leveraged, and since we are focusing on one of the unsupervised learning techniques, namely clustering, the objectives of this thesis are managed by grasping the basic concepts to build a multi-view clustering framework based on mining and learning from multi-view data. Regarding the high dimensionality and data complexity, it is better to study the extrinsic non-linearity of the data structures by carrying multi-view data into a reliable feature space. Thus, an anchor-based kernelization takes place based on the similarity measures to map the original data points into higher dimensional space and increase the understandability and linear separability. In contrast to the real-valued paradigms that make most of the clustering approaches non-scalable, not to mention that they may suffer from computational complexity and memory costs. Hashing is another flexible and worth-noting strategy to efficiently compact the dimensionality and enable working in hamming space, which offers more robustness to noise and outliers. In our research direction, we employed the kernelization process and the common binary codes learning to develop an effective approach called "Automatically Weighted Binary Multi-View Clustering via deep initialization (AW-BMVC)." The work introduced a one-step joint learning model by synthesizing two components, the common discrete representation and the binary matrix factorization. A self-estimation of each view/sample during the learning process is considered. The problem is perfectly solved using an alternating optimization scheme, which has been positively influenced by our innovative deep binary matrix initialization. Experimental results on several challenging datasets demonstrate the effectiveness and superiority of the proposed approach over state-of-the-art methods.

Keywords : Multi-view Clustering, Large Scale, Anchors, Machine Learning, Discrete Representation, BD-FFT, Dimentionality Reduction, Binary Hashing, Deep Learning.

Résumé

La multivue est définie comme une sorte de données riches grâce aux nombreuses fonctionnalités présentées sous la forme de plusieurs matrices de données ; ces matrices sont obtenues par différentes mesures, expériences ou transformations. L'analyse de données multivues implique de traiter simultanément l'ensemble des différentes fonctionnalités dans le but de tirer le meilleur parti de leurs informations afin de prendre une décision efficace. Dans de nombreuses applications réelles, les données multivues sont naturellement générées, en particulier, en vision par ordinateur et en analyse d'image. Chaque représentation de caractéristique correspond à une vue et peut représenter une formulation ou des propriétés statistiques distinctes. Il est convenu que les données multivues partagent une information commune ainsi que contiennent des informations complémentaires ; l'étape cruciale de l'apprentissage automatique et de la découverte des connaissances consiste à apprendre comment ces deux compréhensions peuvent être correctement manipulées et exploitées, et puisque nous nous concentrons sur l'une des techniques d'apprentissage non supervisé, à savoir le clustering, les objectifs de cette thèse sont gérés en saisissant les concepts de base pour construire un framework de multivues clustering basé sur l'exploration et l'apprentissage à partir de données multivues. En ce qui concerne la haute dimensionnalité et la complexité des données, il est préférable d'étudier la non-linéarité extrinsèque des structures de données en transportant les données multivues dans un espace de caractéristiques fiable. Ainsi, une kernelisation en fonction des points d'ancrages est effectuée sur la base des mesures de similarité afin de mapper les points de données d'origine dans un espce de dimension supérieure et d'augmenter la compréhensibilité et la séparabilité linéaire. Contrairement aux paradigmes à valeur réelle qui rendent la plupart des approches de clustering non évolutives sans parler de la complexité de calcul et des coûts de mémoire. Le hachage est une autre stratégie flexible à noter qui permet de compacter efficacement la dimensionnalité et permettre de travailler dans l'espace de Hamming qui offre plus de robustesse au bruit et aux valeurs aberrantes. Dans notre direction de recherche, nous avons employé le processus de kernelisation ainsi que les codes binaires communs en apprenant à développer une approche efficace appelée Automatically Weighted Binary Multi-View Clustering via deep initialization (AW-BMVC). Le travail en question a introduit un modèle d'apprentissage conjoint en une étape en synthétisant deux composants, la représentation discrète commune et la factorisation de la matrice binaire. Une auto estimation de chaque vue/sample pendant le processus d'apprentissage est prise en considération. Le problème est parfaitement résolu en utilisant un système d'optimisation alterné, qui a été positivement influencé par notre initialisation innovante de matrice binaire profonde. Les résultats expérimentaux menés sur plusieurs ensembles

de données défiants démontrent l'efficacité et la supériorité de l'approche proposée par rapport aux méthodes de la littérature.

Mots-clés : Multivues clustering, Grande échelle, Points d'ancrages, Apprentissage machine, Représentation discrète, BD-FFT, Réduction de la dimensionnalité, Hachage binaire, Apprentissage en profondeur.

الملخص

يُعرَفْ العرض المتعدد بأنه نوع من البيانات الثرية بفضل الميزات العديدة المقدمة في شكل مصفوفات بيانات متعددة ؛ يتم الحصول على هذه المصفوفات عن طريق قياسات أو تجارب أو تحويلات مختلفة. يتضمن تحليل البيانات متعدد العروض معالجة جميع الميزات المختلفة في وقت واحد بهدف الحصول على أقصى استفادة من معلوماتهم من أجل اتخاذ قرار فعال. في العديد من تطبيقات العالم الحقيقي ، يتم توليد بيانات العرض المتعدد بشكل طبيعي ، لا سيما في رؤية الكمبيوتر وتحليل الصور. يتوافق تمثيل كل ميزة مع طريقة عرض ويمكن أن يمثل صياغة أو خصائص إحصائية مختلفة. تم الاتفاق على أن البيانات متعددة العروض تتقاسم فيما بينها معلومات مشتركة وكذلك تحتوي على معلومات تكميلية ؛ تتمثل الخطوة الحاسمة في التعلم الآلي واكتشاف المعرفة في تعلم كيف يمكن التلاعب بهذين المفهومين واستغلالهما بشكل صحيح. وبما أننا نركز على إحدى تقنيات التعلم غير الخاضعة للإشراف ، وهي التجميع ، تتم إدارة أهداف هذه الأطروحة من خلال استيعاب المفاهيم الأساسية لبناء نموذج تجميع متعدد العروض يعتمد على الإستخلاص والتعلم من بيانات العرض المتعدد. فيما يتعلق بالأبعاد العالية وتعقيد البيانات ، من الأفضل استكشاف اللاخطية المتأصلة في هياكل البيانات عن طريق نقل بيانات العرض المتعدد إلى فضاء ميزة يعول عليه. وبالتالي ، يتم إجراء الكرنل المستند إلى نقاط إرتكاز بناءً على مقاييس التشابه من أجل تحويل نقاط البيانات الأصلية إلى فضاء ذو أبعاد أعلى ورفع قابلية الفهم و الفصل الخطي. على النقيض من النماذج ذات القيمة الحقيقية التي تجعل معظم مناهج التجميع غير قابلة للتطوير ناهيك على أنها قد تعاني من التعقيد الحسابي وتكاليف الذاكرة .تعتبر التجزئة إستراتيجية أخرى مرنة جديرة بالملاحظة تأتي جنبًا إلى جنب مع ضغط الأبعاد بكفاءة وتمكين العمل في فضاء هامنج ، مما يوفر مزيدًا من الثبات للضوضاء و القيم المتطرفة. في اتجاه بحثنا ، قمنا بتوظيف عملية الكرنلة بالإضافة إلى تعلم الرموز الثنائية المشتركة لتطوير نهج فعال يسمى Automatically Weighted Binary Multi-View Clustering via deep initialization (AW-BMVC) . قدم العمل المعني نموذجًا تعليميًا مشتركًا من خطوة واحدة ، عن طريق توليف مكونين ، التمثيل المتقطع المشترك وتفكيك المصفوفة الثنائية إلى معاملين. يتم الأخذ في الإعتبار، التقدير الذاتي لكل عرض / عينة، أثناء عملية التعلم. تم حل المسألة بشكل مثالي باستخدام خطة التحسين المتناوب، والتي كان لها تأثير إيجابي من خلال تهيئتنا المبتكرة للمصفوفة الثنائية العميقة. تظهر النتائج التجريبية التي أجريت على العديد من مجموعات البيانات ذات التحدي، كفاءة وتفوق النهج المقترح على أحدث الأساليب.

الكلمات المفتاحية التجميع متعدد العروض، النطاق الواسع، نقاط إرتكاز، تدريب الآلة، تمثيل متقطع، BD-FFT ، تقليل الأبعاد، التجزئة الثنائية ، التعلم العميق.

Contents

List of Figures

List of Tables

Abbreviations

ACC :	Accuracy
ADMM :	Alternating Direction Method of Multipliers
ADPLM :	Adaptive Discrete Proximal Linearized Minimization
BD-FFT :	Bidirectional Fast Fourier Transform
CCA :	Canonical Correlation Analysis
CH :	Color Histogram
CM :	Color Moment
CNN :	Convolutional Neural Network
CORR :	Color Correlation
DBSCAN :	Density-based Spatial Clustering of Applications with Noise
ED :	Edge Distribution
H_{dis} :	Hamming Distance
HOG :	Histogram of Oriented Gradients
kNN :	k-nearest Neighbor
LBP :	Local Binary Pattern
MVC :	Multi-view Clustering
NMF :	Non-negative Matrix Factorization
NMI :	Normalized Mutual Information
Optics :	Ordering Points to Identify Cluster Structure
PAM :	Partitioning Around Medoids
PCA :	Principal Component Analysis
PHOG :	Pyramid Histogram of Oriented Gradients
RBF :	Radial Basis Function
SIFT :	Scale Invariant Feature Transform
SMRS :	Sparse Modeling Representative Selection
VGG-16 :	Convolutional Neural Network architecture from Visual Geometry Group
WM :	Wavelet Moment
WT :	Wavelet Texture
XOR :	Logical Exclusive OR Operation

Part I

Context And Motivations

Chapter 1

General Introduction

In the era of big data, raw data is considered the core building block for many applications and companies. Witten et al.[1], described in their book "Data Mining", that data must be analyzed and interpreted to explore valuable structures or hidden information. The mining task can generate a new vision and better decisions for future trend prediction. In the past decades, data mining has been known as an entire field of study that has been investigated intensively [2].

One of the exciting topics in data mining is clustering. It is the process of grouping data based on similarity, correlation, or other statistical properties [3], with the help of techniques and algorithms. As the source and channels of information evolved, data could be extracted from multiple applied areas, and observed by various models, where the same physical object, concept, or phenomenon [4], is described from different sources or perspectives. For example, an image comprises different features (descriptors); pictures on the web have tags and narratives attached to them; news originating from multiple news organizations; Sensor information consists of time and frequency. All these cases are just a sub-part of big data known as multi-source or multi-view data [5].

The view can represent the same data captured from a different perspective, called multi-view, or from a different type of sensor, called multi-source. These data types show heterogeneous characteristics, but it is still possible to find links between them. Accordingly, improvements were made to the traditional clustering algorithms for multi-view data clustering.

The key to the success of the clustering process heavily relies on how multiple-view data is appropriately integrated and leveraged. Meanwhile, machine learning tasks become challenging since are built upon empirical measurements where the data may be corrupted or incomplete due to the noise.

To step further, the easy and natural idea of multi-view data integration is concatenating different views into a single data matrix. In fact, dependencies between features within one view are likely to be more relevant than dependencies between two views [6]. Not to mention that it is not wise to ignore the diversities and just consider multiple distinct views as a single representation once all features are simply connected [7]. Consequently, it is better to treat multi-view data independently as different constructed similarity matrices (kernels) and get the summation of kernels for multi-view clustering.

The kernel trick is an apparent benefit to handling the non-linearity for most machine learning problems by projecting each view into higher dimensional space. To take advantage of the previous task, and mitigate the kernelization dimensionality curse, an anchor-based representation is performed based on the RBF kernel, in which the similarity measure relies on only a few data samples being selected.

Assigning weights is another outstanding strategy to facilitate the clustering and fusing of multi-view information properly, to distinguish between different views and alleviate or delete the noisy kernel (view), and provide a boundary error guarantee [6].

Finally, to boost data scalability, a crucial extension is introduced for embedding and learning in a feature space named binary hashing, where the real dense feature dimensions are converted into discrete compact similarity-preserving codes. This promotes significant gains in storage efficiency and computational speed.

More insightful, the first objective is to learn a unified discrete representation using the kernelized multi-view data, hence refining the fusion process by utilizing an automatically view-weighted as well as an automatically sample-weighted strategy to differentiate each view/sample based on their degree of contribution, which is self-estimated by the learning model. Any popular signal-view algorithms could serve the purpose of returning to the primary subject, which is clustering the reached consensus representation. Like K-means, that factorizes the data matrix into two low-rank matrices, center matrix, and assignment

vectors.

As such, a one-step learning algorithm is adopted to combine the discrete representation and binary matrix factorization simultaneously. Here we mention that the pivotal factor lies in the optimal solution, which is highly dependent on the initial setup. A novel technique is carried out by deep features extracted from Vgg16 and compressed into 128 bits using FFT domain transformation. Finally, the results obtained are very interesting. Indeed, we achieved an average accuracy improvement over four real-world datasets of 16%, compared to the second-best baseline scores, which translates to our system's scalability, robustness, and efficiency.

1.1 Context and Motivation

In the era of big data, it is natural to have data with different modalities or come from multiple sources, known as a "multi-view dataset". Unlike single-view clustering, multi-view clustering comes out to study alternative solutions in order to achieve consistent partitioning from such data. In light of the Multi-view Clustering (MVC) literature, intermediate-level fusion is the most expressive approach for representing the shared and individual view features; on the other hand, Non-negative Matrix Factorization (NMF) represents one typical learning model that has attracted sustaining attention in MVC and brought several fascinating advantages: (1) The ability to handle high-dimensional data and create a low-rank approximation, (2) Interpretability of the mathematical factors that reflect conceptual properties within the data, and clustering capability [8][9].

Assume we have a matrix of data $\mathbf{X} \in \mathbb{R}^{M \times N}$, The aim of NMF is to find two low-rank matrices $\mathbf{V} \in \mathbb{R}^{M \times k}$, and $\mathbf{U} \in \mathbb{R}^{N \times k}$, whose multiplication provides a good approximation, i.e., $\mathbf{X} \approx \mathbf{V}\mathbf{U}^T$. i.e., each observation can be explained as an additive linear combination of nonnegative basis vectors. The nonnegativity constraints on the factor matrices are usually enforced to promote the interpretability of the NMF models. Please refer to [10], for an overall conceptualization of NMF-based approaches and domains. For MVC, we admit the hypothesis that different views should share the same underlying structure where the learned matrices from the distinct views should be as consistent as possible. For this purpose, a regularization term was presented to superimpose the distinct views

coefficient matrices toward a unified consensus [11].

Assume we have a multi-view dataset with V views, $\mathbf{X} = \{\mathbf{X}^1, ..., \mathbf{X}^V\}$, The partition of $\mathbf{X}$ into k clusters can be conducted through the following objective function:

$$\begin{aligned} &\min_{\mathbf{V}^v, \mathbf{U}^v} \sum_{v=1}^{V} ||\mathbf{X}^v - \mathbf{V}^v \mathbf{U}^{v^T}||_F^2 + \sum_{v=1}^{V} \lambda_{\mathbf{v}} ||\mathbf{U}^v - \mathbf{U}^*||_F^2 \\ &s.t.\ \mathbf{V}^v \geq 0, \mathbf{U}^v \geq 0, \mathbf{U}^* \geq 0 \end{aligned} \tag{1.1}$$

where $\mathbf{U}^*$ is a learned consensus matrix that captures the latent individual clustering structures within all views, and λ_v, is a hyperparameter used to distinguish the relative view significance and balance between the reconstruction error term and the disagreement term. λ_v is set to be small enough to make the model in Eq.1.1, less sensitive to low-quality views, and thus not require that all views share a common $\mathbf{U}^*$; correspondingly, He et al. premised a similarity constraint on each pairwise view Instead of a rigid consensus constraint [12], it is expected that this scenario of pairwise co-regularization results in complementary action of the two coefficient matrices acquired from each view during the factorization learning. The target function of this method can be defined as:

$$\begin{aligned} &\min_{\mathbf{V}^v, \mathbf{U}^v} \sum_{v=1}^{V} \lambda_v ||\mathbf{X}^v - \mathbf{V}^v \mathbf{U}^{v^T}||_F^2 + \sum_{p,q=1}^{V} \lambda_{\mathbf{pq}} ||\mathbf{U}^p - \mathbf{U}^q||_F^2 \\ &s.t.\ \mathbf{V}^v \geq 0, \mathbf{U}^v \geq 0 \end{aligned} \tag{1.2}$$

where λ_v is an additional parameter associated with the factorization operation to draw the importance of each view and λ_{pq} is assigned to weight the pairwise similarity constraint on $\mathbf{U}^p$ and $\mathbf{U}^q$.

It is noted that when adopting the vector-based l2-norm, the column vector in the assignment matrix $\mathbf{U}$ indicates the membership information; consequently, each element of $\mathbf{U}^T\mathbf{U}$ provides a measure of the cosine similarity between every two discrete partitions. For multiple-view data, the clustering consistency (similarity of clusters) between different views should be emphasized, which is expressed by the cluster-wise CoNMF framework [12]. The pairwise regularization term in Eq. 1.2, is replaced by the Cluster-wise CoNMF as follows:

$$\sum_{p,q=1}^{V} \lambda_{\mathbf{pq}} ||\mathbf{U}^{p^T} \mathbf{U}^p - \mathbf{U}^{q^T} \mathbf{U}^q||_F^2, \tag{1.3}$$

The above NMF models work with real raw datasets and may not improve performance, and this occurrence is experimentally observed when views differ in quality.

In fact, NMF-based methods have recently been developed by adding new criteria or revising old constraints specifically to reach the desired solution and improve the performance of clusters for multi-view clustering. To mention a few, Feng et al. [13], proposed a Graph-based NMF algorithm using Graph regularization terms to preserve the latent data structure during the factorization. Furthermore, Huang et al. [14] introduced a Deep Matrix Factorization (DMF) framework to achieve factorization via multi-hierarchical layers. The concept of "deep" gives the ability to explore non-linearity structure among different views and acquire a consensus low-rank representation that exposes a final partitioning. In many cases, the advances made by multi-view clustering technologies do not treat the problem of scaling time complexity and increasing memory needs. Under the hypothesis that real-world multi-view representation, which usually holds high-dimensional features, may reveal the unavoidable spread of noises during the fusion process. Few studies have been conducted on the clustering of large binary data sets. Gong et al. [15] have developed a method for binary clustering in a view that consists of two separate steps: binary code generation and binary k-means clustering. The main drawback is that the binary code is generated using a data-independent method, Iterative Quantization (ITQ). The work described in [16] adopted a two-level clustering that breaks the link between binary representation and data partitioning. Shen et al. [17] combined binary structural Support Vector Machine (SVM) and conventional k-means in an optimization algorithm to accelerate the large-scale clustering of single views. However, neither method can be applied to large-scale MVC, and the characteristics of multi-view data have yet to be thoroughly investigated. In the meantime, the binary codes generated by. [17] obtained unsatisfactory results due to the lack of a complete joint representation. Zhang et al. [18] have developed an interesting approach called "Binary Multi-View Clustering" (BMVC) to overcome a major problem that requires less computation time and storage cost. BMVC has uncovered two essential elements: collaborative discrete representation learning and binary clustering structure learning in a standard model. By considering only the complementary features, this framework has considered encoding multi-view features into a

common compact binary code. This model provides a non-negative normalized vector to weight the views, with an additional adjustable parameter to balance the importance of the different views. This method suffered from the proper distinction between shared and individual information, which can lead to the loss of local structure preservation in binary code learning. As an alternative working solution to the above problem, the HSIC method has jointly learned a common binary representation and robust discrete cluster structures [19]. The former decomposes each projection into a combination of shareable and individual projections across multiple views to capture the underlying correlations; the latter can significantly improve the computational efficiency and robustness of clustering. However, the above work is very sensitive to the initialization of the binary clustering process, and even the performance degrades when trying to get rid of the extra parameter and learn the weighting factor of each view automatically. The contribution elements will be discussed in the next section to fill the gaps in previous works.

1.2 Contributions

The main goal of our proposal is to design a fast and high-performance model in order to address the problem of large-scale binary multi-view clustering. In this sense, the thesis work is essentially built on four specific inquiries:

(1) Learning in feature space: How could anchor-based representation improve the understandability and the separability ?

(2) The integration(fusion) quality: What kind of knowledge can be transferred between related data views during the combination within the learning model?

(3) The joint learning strategy: Could the unified learning strategy contributes to the overall effectiveness of the model?

(4) The initialization procedure: Why is it necessary to care about the initialization procedure in NMF optimization?

1. Our first contribution concerns the study of anchor selection (sampling) to extract a few representative samples that can explore the entire dataset's structure. Sampling comes as a universal tackler for the computational cost. A smaller random (in terms of ideally manageable) sample from the whole dataset is selected to get

a small, representative subset, and the mining task is performed on that. Thus, these anchors are exploited by a powerful and flexible mapping referred to as a non-linear RBF Kernel. We settled on the K-medoids as the encouraging sampling method compared to random sampling and K-means. Finally, the crucial kernelization step is to generate a new data matrix (similarity matrix) by projecting the original data into a higher dimensional feature space.

2. Our second contribution was about the different types of information that contribute to the enrichment of the fusion system. To achieve this purpose, we have introduced the objective of having views automatically weighted and samples automatically weighted as two independent variables to distinguish the individual characteristics between different views and samples, respectively, while enhancing the consensus representation. This mode is characterized by a self-evaluation based on learning loss. As a result, the noise and outlier effects are perfectly minimized; moreover, adding extra hyperparameters to the model is avoided.

3. In our third scenario, we offered a one-step objective learning strategy to combine two criteria, the discrete representation and the binary clustering in Hamming space, and get the final clustering results without any post-processing. This mode would maintain the coherence and safe handover of the loss rate throughout the learning operation. The overall objective is optimized using an alternating strategy through three regularization parameters; the first is associated with the mapping function to treat the ill-posed problem, the second is conceived for the need of binary code learning (maximum entropy principle), and the third is necessary to sidestep domination between the two aforementioned criteria.

4. Finally, we focus on non-negative matrix factorization as the ultimate multi-view clustering stage. Whatever the constraints imposed, all the NMF approaches are popular iterative algorithms that are very sensitive to the initialization procedure. Therefore, we have developed a highly efficient solution to initialize our binary clustering algorithm by projecting new deep features from Vgg16 into a low-dimensional Hamming space using a "Bidirectional FFT technique" (BD-FFT).

1.3 Organization of the Manuscript

The rest of the thesis is organized as follows:

Chapter 2: represents some background about multi-view domain and applications, specifically unsupervised machine learning challenges in terms of different stages of data integration and mining processes.

Chapter 3: This chapter reviews the literature on multi-view based clustering, whereby a clear taxonomy is schematized, and different popular frameworks are grouped into various categories, including Multi-View Spectral Clustering; Multi-View Subspace Clustering; Multi-View NMF Clustering.

Chapter 4: This chapter presents the proposed work founded upon the mentioned four elements of contribution; Anchor selection technique, View auto-weighted, Sample auto-weighted, and Binary clustering initialization. A flowchart depicts the combined components and the optimization variables; furthermore, we conduct a series of experiments on well-known datasets that are elaborated and discussed to evaluate the model's robustness and efficiency in clustering benchmarks.

Chapter 6: This chapter concludes the thesis work, discusses the implications and limitations of AW-BMVC and promotes future work.

Chapter 2

Multi-view data and multi-view learning

2.1 Introduction

The gist of this work revolves around multi-view data, which is a dataset containing different sub-divisions of the data (it could be of different types), each one describing the same set of entities [20][21]. Besides, the observed entity can be described in all of the views, with possibly missing data in at least one of the views. Since each view describes the same phenomena, multi-view data has, in some respects, inherently correlated features. An example of multi-view data is a news article, which often has text and images about the same event. Another example is an image of something passed through multiple filters; each output version of the raw image data can represent a view. Multi-view learning, which employs multiple distinct representations, forms an important section of machine learning models [22]. These "representations" can be original features of the data or features acquired through specific measurements.

2.2 Data Integration Stages

When building a data analysis workflow, the developer can choose to conduct the integration step in different stages; we can then discern between early (direct combination), middle (sharing a similar structure combination), and late integration (combination after

projection) [23]. The preference for one method over another depends on the specific aspects of the problem, such as the heterogeneity of the raw data and the statistical problem to be treated. Generally, the characteristics of each stage are illustrated in (Fig 2.1). Early integration is the ordinary learning methodology that was resorted to by simply concatenating all the raw variables from the multiple views into a single representation before fitting the unsupervised model. The main drawback of this strategy is the increase in the dimensionality of the feature space against the small dataset, hence generating over-fitting and redundancy of information and wasting the statistical properties of each view. The intermediate strategy involves applying some transformation to each raw view data separately before integrating them into an expressive unified feature space, thus alleviating the problem of increasing data dimensionality. Finally, in late integration, discriminant partitions are first learned independently without understanding the specificity of every single view, and then the outputs are further integrated to make the optimal determination.

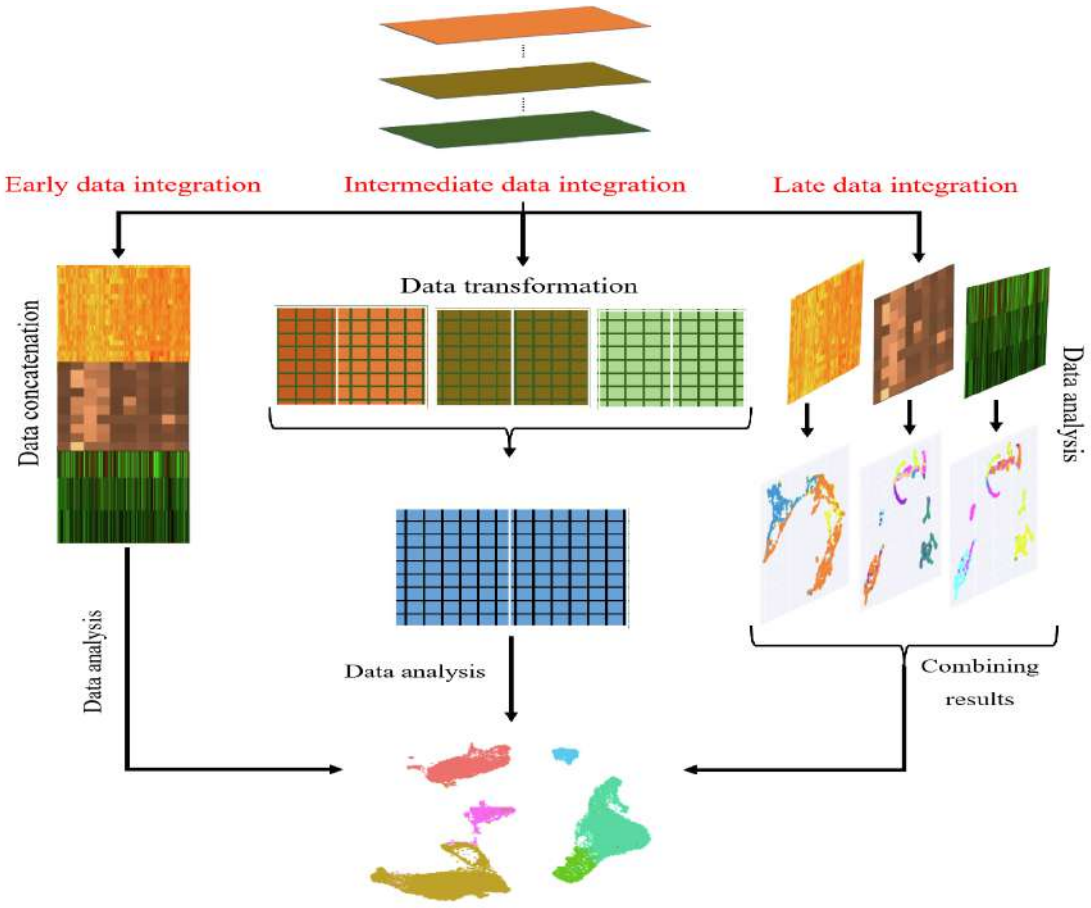


Figure 2.1 – Data Integration Stages

2.3 Multi-view application domains

MVC has been employed in many scientific fields, including computer vision, Natural Language Processing, Textural documents, Bioinformatics, Health informatics, and Social-based data. Figure 2.2 illustrates some multi-view learning application domains.

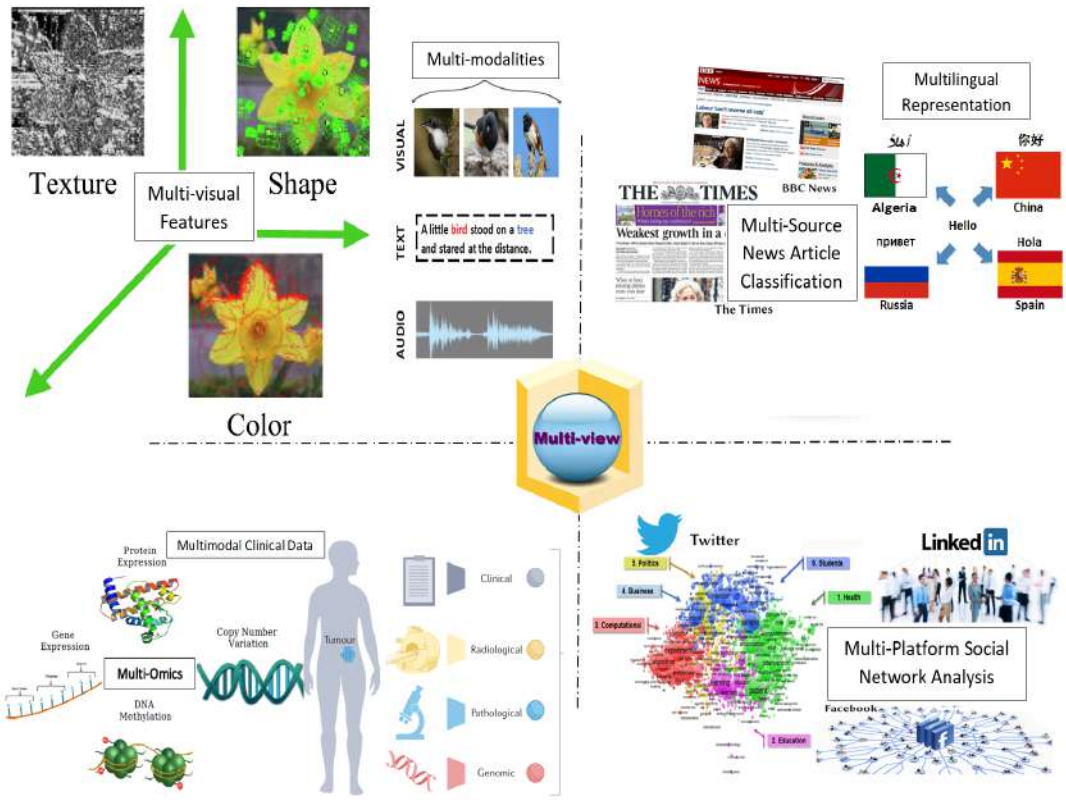


Figure 2.2 – Multi-view learning application domains.

A) **Computer Vision:** MVC has been vastly utilized in image categorization [24], [25],[26], and motion segmentation tasks [27], [28]. Typically, different image feature species, e.g., HOG [29],CENTRIST [30], Color Moment [31], SIFT [32], and LBP [33], could be extracted and collected for cluster analysis.

B) **Natural Language Processing and Textural Document:** In natural language processing, we usually deal with text documents in multiple languages. To classify documents, it is natural to apply MVC [34],[11],[35] by taking each language as one view. Another perspective regarding textual clustering, we could recog-

nize text documents from multiple representations such as syntactic features, i.e., term occurrences [36], topical [37], and semantic links between words [38]. MVC is a powerful way to conduct document categorization [4], [39], with each view corresponding to one feature and capturing an aspect of the text.

C) **Health Informatics and Bioinformatics:** Clustering has many applications in the biomedical and healthcare fields, such as discovering modules of co-regulated genes and finding subtypes of diseases in the context of precision medicine [40]. In medical care, patient information may be distributed in numerous forms, such as nursing notes, pathology tests, genomic data, radiology images, etc. Extracting features from these heterogeneous data presents different and multiple views for the same subject. [41],[42]. These pieces of information could be effectively utilized to develop intelligent models. Garg et al. have introduced "Collective Matrix Factorization" (CMF) to integrate the extracted features from multiple views of clinical records towards locating both subjective and objective guesstimates of patient's conditions and improving healthcare systems. Moreover, the biological applications domain helped deliver a new common set of samples named multi-omics datasets for multi-view analysis. These measurements were obtained from different platforms (e.g., DNA methylation and gene expression). Therefore, clustering multi-omics data has the potential to develop methods [43], [44],[45], for fusing different types of omics and capturing a comprehensive statement of latent disease, particularly as a biological process that is necessary to improve disease detection, treatment, and prevention.

D) **Social Multimedia:** Social media networks often generate human interactions with many types of data within social-based data. For example, different kinds of interactions could be raised on Twitter, such as following or retweeting (network data), as well as non-network data content, such as images or text. Multi-view clustering is of particular importance to combine heterogeneous correspondent data into a meaningful analysis. Twitter topic/event detection is one hot thematics embraced to characterize communities. A multi-view clustering model may integrate

multi-relations among tweets, such as semantic relations, social tag relations, and temporal relations, into a group of keywords, where each cluster of keywords is finally represented as an event [46], [47], [48]. In addition, multi-view clustering has been dedicated to group news stories [49], multimedia collections [50], and social web videos [51].

2.4 Challenges in Multi-View Analysis

Conventional machine learning algorithms, such as discriminant analysis, spectral clustering, artificial neural networks, support vector machines and kernel machines, are designed to operate on single-view data. In contrast, the nature of multi-view data must be carefully considered and highlights the following open problems.

1. **Data Heterogeneity:** The most straightforward approach to driving multi-view datasets using traditional machine learning algorithms is concatenating all the features across different views and constructing one single view. Nonetheless, this concatenation is neither compatible nor expressive, as each view has distinct statistical properties and is usually measured in terms of scale, unit, and variance. Unbiased fusion of multi-view data implies dealing with a transformed feature space that preserves the intrinsic properties of each view. Clustering may be conducted individually on each view, but a late fusion of clustering assignments may fail to capture cross-view correlations.

2. **High-Dimension Low-Sample Size Nature:** In real-life data analysis, we usually deal with many observed variables, such as 10^6 pixels in images, thousands of words in documents, nearly 20K genes in DNA microarrays, etc. The number of samples in these datasets is typically smaller than the dimensions. The small training samples cause system overfitting and poor performance generalization. However, a multi-collinearity issue has occurred when dealing with high-dimensional data like images, in which features are highly correlated. This degrades the consistency properties of the eigenvalues and eigenvectors of the sample covariance matrix [52]. Moreover, in the context of clustering algorithms, features in high-dimensional space are geometrically sparse, resulting in an expensive computation.

3. **Noisy and Redundant Views:** In real-world data, the observations in different views are often influenced by noise due to measurement errors. This noise can be propagated into different views or amplified during the data integration process if it is not well taken care of. Under the assumption that each view provides individual and consensus information about the data's underlying patterns, all the available views are crucial for multi-view learning. However, within each view, one can encounter redundant, disparate, or even worse information, which reduces the quality of multi-view fusion and destroys the cluster structures and decision boundaries learned from the data.

4. **Incomplete Views:** Due to measurement and pre-processing errors, some multi-view approaches institute the study of incompleteness, assuming that part of the samples was not observed in one view (missing sample) or the sample was partially observed (missing variables). Therefore, incomplete views require the employment of a connection between views and restoring the missing samples with the help of interconnected samples in the complete views [53].

5. **Multiple Solutions:** Most of the existing MVC approaches, including single-view clustering, yield a single clustering solution. However, data may reveal different possible groupings in real-world applications where the ground truth is unknown. Only a small number of these are likely to be feasible and meaningful, according to the exploratory analysis problem. To simplify the perception, let us consider partitioning the fruits, such as grapes, apples, and bananas, depending on their color or fruit kind. To date, few works have examined this trend [54], which has discovered multiple clustering structures of multi-view data embedded in different subspaces that are orthogonal to each other; hence, non-redundant clustering solutions were obtained. In another work, Donglin et al. [55] adopted the Hilbert-Schmidt independence criterion to measure non-linear dependencies across different views. Each view was then subjected to a spectral clustering solution. Finally, Chang et al. [56] addressed the problem of how to automatically learn multiple expert views and the clustering structure corresponding to each view based on a novel Bayesian

probabilistic model.

Some of the aforementioned challenges are inherent to multi-view data, like data heterogeneity and incomplete views. In contrast, others, like high-dimension low-sample size nature and low-rank non-linear geometry, exist either in single-view data. Hence, the presence of multiple representations increases the problem's complexity. Therefore, new evolved frameworks must be conceived to mine meaningful patterns embedded in multi-view datasets and efficiently conduct the outlined challenges.

2.5 Learning in Feature Space

Because of the complexity of real-world applications, the target task to be learned from data cannot be treated as a simple linear combination of the given abstract features. Instead, it is strongly related to the power of its representation. Accordingly, the difficulty of the learning function can vary. Therefore, changing the data representation to match the typical learning problem is one of the common pre-processing strategies in machine learning. To raise the computational learning power of the linear algorithm, one solution is to map our features into a high-dimensional space where there exist linear relationships between data points in that space, then perform a linear algorithm. However, it is computationally challenging to represent data in (possibly high-dimensional) space as a workaround to the original problem without explicitly considering the transformed coordinates of the data. We conduct pairwise similarity comparisons (dot product) in the lower-dimensional original space, and then we carry out algorithms that only need the values of that metric. The function that takes the vectors in the original space as its inputs and returns the dot product of the vectors in the feature space is called a kernel function or kernel trick. Other theorems guarantee the existence of such kernel functions under certain conditions. This step describes the mapping of the input space $\mathbf{X}$ into a new space, $\mathbf{F} = \{\phi(\mathbf{x}) | \mathbf{x} \in \mathbf{X}\}$. The function called a kernel "$\mathbf{K}$" if there exists a Hilbert space $\mathbf{H}$ and a map ϕ: $\mathbf{X} \rightarrow \mathbf{F}$ such that

$\mathbf{K}(\mathbf{x}, \mathbf{x}') := \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle \mathbf{F}, \forall \mathbf{x}, \mathbf{x}' \in \mathbf{F}$. Where ϕ: Feature map and $\mathbf{F}$: Feature space.

Figure 2.3 illustrates a simple example of two-dimensional data that is non-linearly separable from the original space mapped into a two-dimensional linearly separable feature

space using a kernel function.

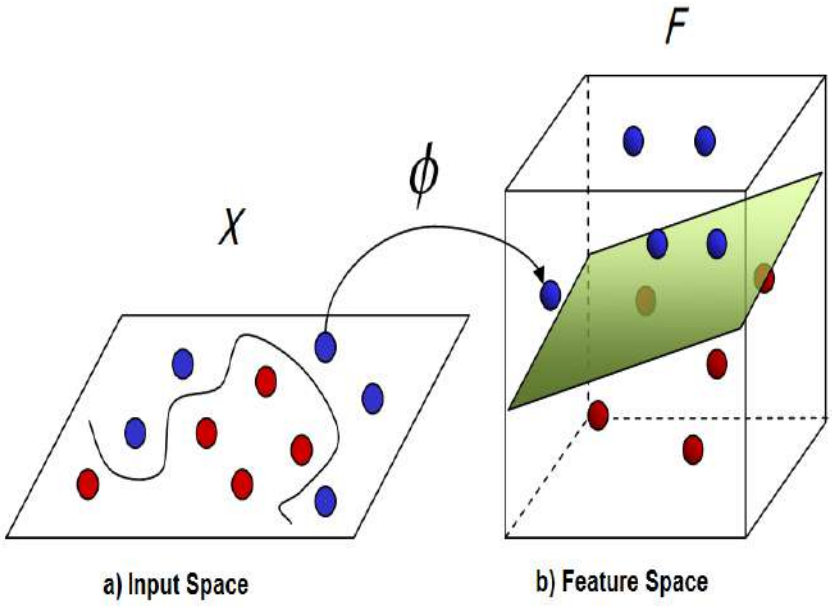


Figure 2.3 – Transformation from input space into feature space using a kernel function.

Before we can complete this path, we should first define the properties of the function $\mathbf{X}(\mathbf{x},\mathbf{y})$ necessary to ensure that it is a kernel for some feature distance. Before traversing this path, to ensure that the function is a kernel for some feature distance, we must first define the necessary properties of the function $\mathbf{X}(\mathbf{x},\mathbf{y})$. The obvious condition is symmetry:

$\mathbf{K}(\mathbf{x},\mathbf{y}) = \langle\phi(\mathbf{x})\cdot\phi(\mathbf{y})\rangle = \langle\phi(\mathbf{y})\cdot\phi(\mathbf{x})\rangle = \mathbf{K}(\mathbf{y},\mathbf{x})$

and it also should satisfy the inequalities that follow from the Cauchy-Schwarz inequality,

$\mathbf{K}(\mathbf{x},\mathbf{y})^2 = \langle\phi(\mathbf{x})\cdot\phi(\mathbf{y})\rangle^2 \leq ||\phi(\mathbf{x})||^2||\phi(\mathbf{y})||^2 = \langle\phi(\mathbf{x})\cdot\phi(\mathbf{x})\rangle\langle\phi(\mathbf{y})\cdot\phi(\mathbf{y})\rangle$

$= \mathbf{K}(\mathbf{x},\mathbf{x})\mathbf{K}(\mathbf{y},\mathbf{y})$.

However, these conditions are insufficient to guarantee a feature space's existence; other approaches, such as Mercer's Theorem, have introduced a series of Mercer kernel functions described in this book [57].

2.5.1 Kernel Function

Kernel methods can be mainly classified based on the effect of translation into two functions of interest: (1) shift-invariant and (2) dot product kernels [58]. The first class

of kernels denoted translation invariance, while this property is not maintained in the second category of kernels.

2.5.1.1 Shift invariant kernels

A shift-invariant kernel comprises every kernel function $\mathbf{K}$ whose values rely exclusively on the differences between the input vectors.

$\mathbf{K}(\mathbf{x},\mathbf{y}) = \mathbf{K}(\mathbf{x}-\mathbf{y})$

The most widely used function that falls under this section is the Gaussian kernel.

$\mathbf{K}(\mathbf{x},\mathbf{y}) = exp - \frac{||\mathbf{x}-\mathbf{y}||^2}{2\sigma^2}$

Gaussian kernel(RBF): It is a complete form of the Gaussian function to estimate the distance between two vectors given the width σ^2 (variance). This function offers excellent scale for a large number of input features.

Laplacian kernel : It is a modified form of the radial basis function kernel (Manhattan distance metric) given the width σ (standard deviation). It has some similar properties to the previous exponential kernel and is less affected by changes in the data.

$\mathbf{K}(\mathbf{x},\mathbf{y}) = exp - \frac{||\mathbf{x}-\mathbf{y}||_1}{\sigma}$

For the two previous functions, estimating the correct value of "σ" is important to measure how close two points are to each other.

2.5.1.2 Dot product kernels

As the name suggests, the result of this type of kernel can be expressed as the dot product across the input vectors.

Linear Kernel: It is the basic type of kernel with a one-dimensional structure that represents a linear scale of one vector with another.

$\mathbf{K}(\mathbf{x},\mathbf{y}) = \mathbf{x}^T\mathbf{y} + \mathbf{c}$

where $\mathbf{c}$, denoted the bias or shift which sometimes added to the result.

Polynomial kernel: It computes a degree q polynomial of the dot product of two vectors $\mathbf{x}$ and $\mathbf{y}$. The polynomial kernel deals with the degree/order of the features, thus allowing the learning of non-linear models by computing a degree q polynomial of the original variables.

$\mathbf{K}(\mathbf{x},\mathbf{y}) = (\mathbf{x}^T\mathbf{y} + \mathbf{c})^q$

The constant $\mathbf{c} \geq 0$ is a trading-off parameter between higher-order and lower-order terms in the polynomial. When $\mathbf{c} = 0$, the kernel prescribed as homogeneous.

Sigmoid Kernel: This type of kernel employs the *tanh* function. It is mainly used as an activation function for neural networks.

$\mathbf{K}(\mathbf{x}, \mathbf{y}) = tanh(\mathbf{m}\mathbf{x}^T\mathbf{y} + \mathbf{c})$

where $\mathbf{m}$ is the slope, and $\mathbf{c}$ refers to the intercept term.

By elaborating an RBF kernel, we can Intuitively restrained the number of implicit comparisons over a few selected points(anchors). One approach is to think about anchors as being statistically representative of each dataset by focusing on 3 different selection methods: Random Sampling; K-medoids; SMRS (Sparse Modeling Representative Selection); moreover, we suppose that the number of the selected anchors for each view is limited to 1000 ($m = 1000$).

2.5.2 Anchor selection (Sampling)

Sampling comes as a universal tackler for the computational cost, where a smaller (ideally controllable) random samples is selected from the entire data set in order to get a few representative subset and the mining task is performed on that [59]. There are 2 main types of sampling: (A) Probability (random) sampling (B) Non-probability sampling.

A) **Probability sampling method**

A probability sampling method is considered a selection procedure that serves as a form of random selection, knowing that each instance of the dataset has an opportunity of being chosen. It is generally held to be the most precise type of sampling. Furthermore, we will discuss two techniques, Discrete uniform distribution and K-medoids.

(a) Discrete uniform distribution

Suppose a given finite population that we need a sample So that each subject has an equal chance of selection with no replacement (Within the selected samples, all its subjects must be different). The discrete uniform distribution [60], represents the theoretical concept of the random sampling model. In MATLAB random sampling is achieved by the function “randsample” (Statistics toolbox).

the original dataset is coded as a vector population by the indices and a random sample of size k are founded.

(b) Sampling with K-medoids (Cluster sampling)

Partitioning Around Medoids (PAM) or the K-medoids [61], is a well-known clustering technique that is slightly modified from the K-means algorithm. They both attempt to minimize the squared-error but the K-medoids algorithm is more robust to noise than the K-means algorithm [62]. A medoid can be defined as that object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal. It starts from an initial select of k representative medoid data items arbitrarily, and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resultant clustering.

For each pair of non-medoid data item $\mathbf{X}_i$ and selected medoid $\mathbf{C}_i$, the total swapping cost J is calculated. If $J<0$, $\mathbf{C}_i$ is replaced by $\mathbf{X}_i$. Thereafter each remaining data item is assigned to cluster based on the most similar representative medoid. This process is repeated until there is no change in medoids. The dissimilarity of the medoid ($\mathbf{C}_i$) and object ($\mathbf{X}_i$) is calculated by using

$$\mathbf{J} = \sum_{\mathbf{C}_i} \sum_{\mathbf{X}_i \in \mathbf{C}_i} ||\mathbf{X}_i - \mathbf{C}_i||^2 \tag{2.1}$$

B) **Non-probability sampling methods**

A core characteristic of non-probability sampling technique is that samples are selected based on subjective judgment and utilizes a convenient selection of units from the population rather than random selection.

(a) Sampling using Sparse Modeling Representative Selection (SMRS)

Draw inspiration from sparse coding, as a universal method for data modeling; we introduce the problem of finding data representatives using dimensionality reduction in the object space [63]. Specifically, collecting N data points of a dataset in $\mathbb{R}^m$ as columns of a data matrix $\mathbf{Y} \in \mathbb{R}^{m \times n}$; we wish to find at most $k \ll N$ representatives that best reconstruct the data collection.

Learning Compact Dictionaries:

Consider a set of points in $\mathbb{R}^m$ Organized as a column matrix of a single view data $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_N]$. The dictionary learning approach is interpreted as linear combination that is able to efficiently approximate a compact dictionary $\mathbf{D} = [\mathbf{d}_1 \dots \mathbf{d}_l] \in \mathbb{R}^{m \times l}$ and coefficients $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_N] \in \mathbb{R}^{l \times n}$ simultaneously, from a collection of given data points. We may typically convey the best representation by only optimizing an objective criterion subject to verified constraints. In the sparse dictionary learning model, we must satisfy the sparsity condition over the coefficient matrix $\mathbf{X}$ by solving the following objective:

$$\min_{\mathbf{D},\mathbf{X}} = ||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2 \quad s.t ||\mathbf{X}_i||_0 \leq s, ||\mathbf{d}_j||_2 \leq 1, \forall i, j, \tag{2.2}$$

where $||\mathbf{X}_i||_0$ denotes the number of nonzero components among $\mathbf{X}_i$. In this case, we are still far apart from the efficient representation because of incomplete learned atoms of the dictionary. Therefore, we enforce selecting representatives from the extant data points; otherwise, approximate each data point based on the reconstruction error as a linear mapping of all the data.

To achieve that, we impose the matrix of data points as a dictionary $\mathbf{Y}$ and optimize the formulation with standard L_1 relaxation (A minimal $L-norm$ solution is also the sparsest solution); given by a Lagrangian form:

$$\min \lambda ||\mathbf{C}||_1, q + \frac{1}{2}||\mathbf{Y} - \mathbf{Y}\mathbf{C}||_F^2 \quad s.t \ 1^T\mathbf{C} = 1^T \tag{2.3}$$

where λ is a parameter that balances the tradeoff between sparsity and reconstruction error and $\mathbf{C}$ is the coefficient matrix. This convex optimization problem is solved based on Alternating Direction Method of Multipliers (ADMM) space [64]. Practically the sampling is achieved by ranking the examples based on the $L2$ norm of the associated rows in a coding matrix $\mathbf{C}$.

2.5.3 Hashing for dimensionality reduction

Hashing is a family of popular approaches that rely on approximate nearest neighbor search towards an efficient dimensionality reduction and is employed as a primary machine learning aspect in many large-scale search and retrieval including feature compression [65], re-ranking [66], and computing environments [67]. The need for hashing techniques is becoming more crucial due to the huge growth in data sizes. In the context of clustering, the actual size of the raw feature dataset is often very large in terms of memory requirement, and it is often time-consuming to compare any two images. Therefore, to speed up the target task, hashing came up with a beneficial technology of compression referred to as binary-code representation in the Hamming space by mapping nearby points in the original feature space into low-dimensional (compact codes) closer binary codes in the hash code space. The distance computed on a compact representation is very efficient, as the codes are much smaller than the original data items. As a result, we can achieve a sub-linear or constant search time complexity. Moreover, such encoding results in great memory cost reduction and scalable ability to large-scale datasets. For example, with only 16 MB of memory, we can store a dataset of 1 million points with 128 bits of encoding for each data point.

2.5.3.1 Hashing function

Assume a dataset with n data points $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_n] \in \mathbb{R}^d$.

The hashing function $\mathbf{h}(\cdot)$ can be used to map each sample within the data into hash values (hash codes) as $\mathbf{h}(\mathbf{X}) = [\mathbf{h}(\mathbf{x}_1), \mathbf{h}(\mathbf{x}_2) \dots \mathbf{h}(\mathbf{x}_n)] \in [0, 1]$.

To accomplish the hashing task, three key terminologies associated with the hash learning techniques must be considered:

1. **Nature of hash function** According to the hash function nature, different algorithms can be counted based on kernels [68], linear projections [69], neural networks [70], etc. A common model is the linear hash approach, where each bit sequence output is associated with a projection vector w and a quantization function (usually the signum). For example, $\mathbf{y} = sign(\mathbf{w}^T\mathbf{x}) \in \{0, 1\}$,
   where $sign(\mathbf{w}^T\mathbf{x}) = 1$ if $\mathbf{w}^T\mathbf{x} \geq 0$ and 0 otherwise.

Assigning a hash function is a pivotal decision that affects the flexibility of hash codes and the computational feature space partitioning.

2. **Similarity Measure**

   A set of binary codes of length N: To measure the distance between two binary equal-length vectors, the Hamming distance stipulates the number of positions where the bits differ.

   $\mathbf{H}_{dis}\ (0011000, 0111011) = 3$

   $\mathbf{H}_{dis}\ (1011001, 1011001) = 0$

   $\mathbf{H}_{dis}$ can be computed using a bitwise XOR operation.

   Due to its computation efficiency and low memory footprint, it should be pointed out that an exhaustive search in the Hamming space is quite faster than in the original feature space.

3. **Approximate Nearest Neighbour (ANN)**

   The nearest neighbor is defined as the process of finding a data point among n data points $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_n] \in \mathbb{R}^d$, previously preprocessed into a data structure, which is closest to the query point $\mathbf{y} \in \mathbf{X}$.

   $NN(\mathbf{y}) = \arg\min_{\mathbf{x} \in \mathbf{X}} dist(\mathbf{x}, \mathbf{y})$

   where dist$(\mathbf{x}, \mathbf{y})$ is the distance between $\mathbf{y}$ and $\mathbf{x}$.

   Typically, if $\mathbf{X}$ lies in d-dimensional space $\mathbb{R}^d$, p-norm (usually p = 2) distance is induced:

   $||\mathbf{x} - \mathbf{y}||_p = \left(\sum_{i=1}^{d} |\mathbf{x}_i - \mathbf{y}_i|^s\right)^{\frac{1}{s}}$

   Nearest Neighbour Search (NNS) is the optimization algorithm or similarity search, proximity search, which addresses the process of finding the most similar data points. Given any positive real ϵ; ANN returns a $(1+\epsilon)$ point or multiple points of the distance between a query point y and the true nearest neighbor $\mathbf{x}^*$,

   $d(\mathbf{x}, \mathbf{y}) \leq (1+\epsilon) dist(\mathbf{x}^*, \mathbf{y})$.

Curse of dimensionality: As the number of features characterizing each sample in the data set increases, the measurement of Euclidean distances between data points becomes similar, making achieving efficient clustering challenging.

2.5.3.2 Hashing function learning

Similarity-preserving hashing models go through two phases: (1) projection learning and (2) quantizing the projected data into binary codes. The first stage focuses on achieving low-dimensional feature embedding with linear or non-linear projections to encourage similar points to be closer. In the second stage, a quantizer is introduced to quantify each real-valued projection dimension into binary code using a thresholding operation. Based on the projection, the traditional hashing methods can be further divided into two big classes [16], [71]: Data-independent methods and Data-dependent methods. Please see Hashing methods Categorizing chart 2.4.

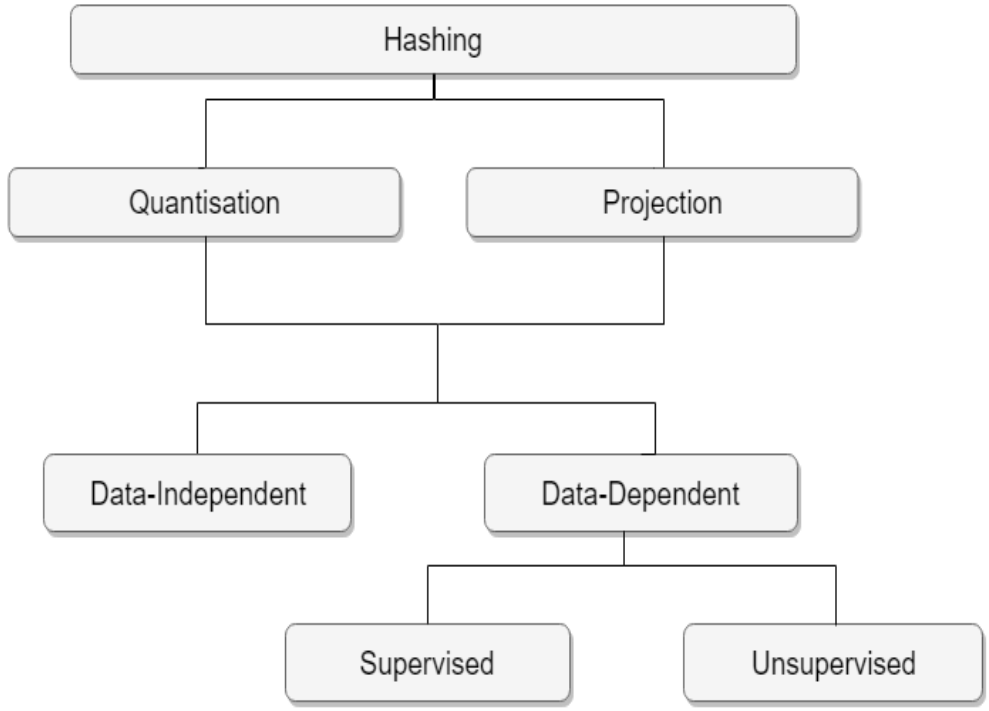


Figure 2.4 – Hashing methods Categorization.

Most works focus on the projection stage, which seeks to find good projection functions. For example, Locality Sensitive Hashing (LSH) [72] is one of the representative data-independent approaches. It employed a projection function derived from a random Gaussian matrix. Like LSH, shift-invariant kernel hashing (SIKH) [73] adopted a random projection and a shifted cosine function to produce hash values. Both LSH and SIKH are built on the hypothesis that points with the greatest similarity have a high probability

of being exposed to the same hash codes. All data-independent methods fall into the trade-off between satisfactory performance and the need for short codes, which makes them less efficient due to the higher storage and computational cost. From the literature, it has been shown that the ability to capture a data structure positively affects the performance of several tasks, such as retrieval and clustering. Therefore, most recent works take up data-driven approaches whose hash functions are learned from the training data. Torralba et al. [74], proposed spectral hashing for similarity graph partitioning. Binary reconstruction embedding (BRE) [75] is another hash function learning that optimizes the reconstruction loss between the original and Hamming distances. For feature presentation learning and hashing, numerous end-to-end (i.e., feature extraction followed by binarization) based deep learning approaches have been proposed, Xia et al. introduced the first end-to-end work, CNNH [76], to learn the hash functions based on the deep convolutional network and pairwise similarity matrix. Zhu et al. [70] proposed a DHN approach based on two learning criteria similarity-preserving by improving the pairwise entropy loss while controlling the hash quality by improving the pairwise quantization error. Finally, DMDH [77] addresses the original discrete problem to transform it into a differentiable optimization by minimizing the objective discrepancy through the Taylor series expansion. The above works summarize hashing techniques driven by single-view shallow learning or algorithms based on deep hashing. With the progress attained by multi-view learning, it is substantiated that integrating multiple descriptors enables learning more informative hash functions [78], [79]. Several multi-view hashing works have emerged by pursuing binary code compactness and exploiting the complementary information across hash functions. For example, the work in [80] learns computationally feasible multiple discriminative hash tables fusion based on the exemplar-based approximation techniques to leverage multi-view information and table correlations. For Unsupervised methods, [81] produced hash codes by exploring the global and local structures. In [82], Shen et al. suggested a matrix factorization technique to generate hash codes with adaptive kernel space learning. In contrast, supervised methods utilize information from the semantic labels to reinforce similarities between views. In [83], Zhang et al. learned hash codes by employing multi-view graphs, in accordance with a view weighting to value different contributions. Finally,

Kim et al. presented an anchor graph multi-view hashing [84], which induced a low-rank form of the averaged similarity matrix.

2.6 Evaluation criteria

Broadly, the evaluation of the clustering algorithm is assessed as a measurement of goodness. On the other hand, clustering stability is known as the sensitivity of the algorithm to the different tunable parameters, for example, the number of clusters. In contrast, clustering tendency evaluates the cluster-ability, that is, whether the dataset comprises meaningful clusters. It exists several statistical functions and validity metrics for each of the aforesaid schemes, which can be categorized into three basic classes:

External: As its name indicates, it represents a validation procedure where the criteria is independent of the dataset, so usually, additional information is needed or prior knowledge about the groups is determined by experts for example, the true label for each datapoint.

Internal: It defines a validation procedure and utilizes criteria that are related to the dataset itself, for example, cluster compactness can measure the intra-cluster and inter-cluster distances to acquire respectively how similar or far apart the datapoints.

Relative: Relative validation metrics are used to directly evaluate the clustering structures obtained from the same version of the algorithm which is generally triggered by different parameters; for instance, we can try a different number of clusters, in order to select its optimal value.

1. **External measures**

Following the above definition of external validation, we assume that the true class labels (ground truth) are provided. This external information plays a major role in gauging the extent to which partition labels match the supplied ground truth. Considering $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{n}$ which represents a dataset comprises $\mathbf{n}$ points in a d-dimensional feature space, grouped into $\mathbf{k}$ partitions. Given $\mathbf{y}_i \in \{1, 2, \ldots, \mathbf{k}\}$ which indicates for each point, the membership information (true class label). $\mathbf{T} = \{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_k\}$, represents a given ground-truth vector, where the partition $\mathbf{T}_j$ denotes all datapoints with j label , i.e., $\mathbf{T}_j\{\mathbf{x}_i \in \mathbf{X} | \mathbf{y}_i = j\}$. Let also designate $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_r\}$ over the same dataset, that

identify the obtained partitioning into r clusters of a specific clustering algorithm. For clarification, the ground-truth clustering will refer to $Tvect$ and each partition to $\mathbf{T}_i$. moreover, the clustering will be referred to $\mathbf{C}$ where will call each $\mathbf{C}_i$, as a cluster. Typically, the clustering algorithm will be driven with the true number of classes, i.e $\mathbf{r} = \mathbf{k}$. Nonetheless, for the generalization purpose, we hold the distinction notation between $\mathbf{r}$ and $\mathbf{k}$. Note that we will report the experimental results using the three most widely used external evaluation metrics, including, Accuracy (ACC), Normalized mutual information (NMI) and Purity; thereby, a theoretical summary of these techniques is reviewed in the following section.

1. **Unsupervised clustering accuracy**

   Unsupervised clustering accuracy (ACC)[85] is roughly the same as classification accuracy; It differs from its predecessor in that it uses a function to find the best mapping between the output indicator vector $\mathbf{c}$ and the ground truth $\mathbf{y}$. This designation is necessary since an unsupervised algorithm may use a distinct label than the real ground truth to describe the same data collection. This metric for assessing the quality of clustering is described as follows:

   $\text{Acc}=\frac{\sum_{i=1}^{n}\delta(\mathbf{y}_i,map(\mathbf{c}_i))}{n}$,

   Where n is the number of objects, $\mathbf{y}_i$ and $\mathbf{c}_i$ denote the true category label and the obtained cluster label of the image respectively. $\delta(\mathbf{y},\mathbf{c})$ is a function that equals 1 if $\mathbf{y} = \mathbf{c}$ and equals 0 otherwise. $map(\cdot)$ is a permutation function that maps each cluster label to a category label, and the optimal matching can be found by the Hungarian algorithm [86].

2. **Entropy-based Measures**

   • **Conditional Entropy** The entropy of a clustering $\mathbf{C}$ is defined as

   $\mathbf{H}(\mathbf{C}) = -\sum_{i=1}^{r}\mathbf{pC}_i \log \mathbf{pC}_i$

   where $\mathbf{pC}_i = \frac{\mathbf{n}_i}{\mathbf{n}}$ is the probability of cluster $\mathbf{C}_i$. Likewise, the entropy of the partitioning $\mathbf{T}$ is defined as

   $\mathbf{H}(\mathbf{T}) = -\sum_{j=1}^{k}\mathbf{pT}_j \log \mathbf{pT}_j$

   where $\mathbf{pT}_j = \frac{\mathbf{m}_j}{\mathbf{n}}$ is the probability of partition $\mathbf{T}_j$. The cluster-specific entropy of,

that is, the conditional entropy of $\mathbf{T}$ with respect to cluster $\mathbf{C}_i$ is defined as [87],

$\mathbf{H}(\mathbf{T}|\mathbf{C}_i) = -\sum_{j=1}^{k} \frac{\mathbf{n}_{ij}}{\mathbf{n}_i} \log \frac{\mathbf{n}_{ij}}{\mathbf{n}_i}$

The conditional entropy of $\mathbf{T}$ given clustering $\mathbf{C}$ is then defined as the weighted sum:

$\mathbf{H}(\mathbf{T}|\mathbf{C}) = -\sum_{i=1}^{r} \frac{\mathbf{n}_i}{\mathbf{n}} \mathbf{H}(\mathbf{T}|\mathbf{C}_i) = -\sum_{i=1}^{r} \sum_{j=1}^{k} \frac{\mathbf{n}_{ij}}{\mathbf{n}} \log \frac{\mathbf{n}_{ij}}{\mathbf{n}_i}$

$= -\sum_{i=1}^{r} \sum_{j=1}^{k} \mathbf{p}_{ij} \log \frac{\mathbf{p}_{ij}}{\mathbf{pC}_i}$

where $\mathbf{p}_{ij} = \frac{\mathbf{n}_{ij}}{\mathbf{n}}$ is the probability that a point in cluster i also belongs to partition j.

The greater the conditional entropy, the more a cluster's members are divided into various partitions. The conditional entropy value for flawless clustering is zero, while the worst potential conditional entropy value is $\log \mathbf{k}$. Extending on the preceding statement, we can see that,

$\mathbf{H}(\mathbf{T}|\mathbf{C}) = -\sum_{i=1}^{r} \sum_{j=1}^{k} \mathbf{p}_{ij} (\log \mathbf{p}_{ij} - \log \mathbf{pC}_i)$

$= -\left(\sum_{i=1}^{r} \sum_{j=1}^{k} \mathbf{p}_{ij} \log \mathbf{p}_{ij}\right) + \sum_{i=1}^{r} \left(\log \mathbf{pC}_i \sum_{j=1}^{k} \mathbf{p}_{ij}\right)$

$= -\sum_{i=1}^{r} \sum_{j=1}^{k} \mathbf{p}_{ij} \log \mathbf{p}_{ij} + \sum_{i=1}^{r} \mathbf{pC}_i \log \mathbf{pC}_i$

$= \mathbf{H}(\mathbf{C}, \mathbf{T}) - \mathbf{H}(\mathbf{C})$

Where $\mathbf{H}(\mathbf{C}, \mathbf{T}) = -\sum_{i=1}^{r} \sum_{j=1}^{k} \mathbf{p}_{ij} \log \mathbf{p}_{ij}$

is the joint entropy of $\mathbf{C}$ and $\mathbf{T}$.

The conditional entropy $\mathbf{H}(\mathbf{T}|\mathbf{C})$ thus gauges the residual entropy of $\mathbf{T}$ when the clustering $\mathbf{C}$ is taken into account. $\mathbf{H}(\mathbf{T}|\mathbf{C}) = 0$ if and only if $\mathbf{T}$ is totally specified by $\mathbf{C}$, which corresponds to the ideal clustering. If $\mathbf{C}$ and $\mathbf{T}$ are uncorrelated, then $\mathbf{H}(\mathbf{T}|\mathbf{C}) = \mathbf{H}(\mathbf{T})$, implying that $\mathbf{C}$ has no information about $\mathbf{T}$.

- **Normalized Mutual Information**

The mutual information [88],is defined as the amount of common information between clustering $\mathbf{C}$ and partitioning $\mathbf{T}$, as follows

$\mathbf{I}(\mathbf{C}, \mathbf{T}) = \sum_{i=1}^{r} \sum_{j=1}^{k} \mathbf{p}_{ij} \log\left(\frac{\mathbf{p}_{ij}}{\mathbf{pC}_i \cdot \mathbf{pT}_j}\right)$

It quantifies the relationship between the observed joint probability $\mathbf{p}_{ij}$ of $\mathbf{C}$ and $\mathbf{T}$, and the predicted joint probability, under the independence assumption, $\mathbf{pC}_i \cdot \mathbf{pT}_j$. When $\mathbf{C}$ and $\mathbf{T}$ are unrelated, $\mathbf{p}_{ij} = \mathbf{pC}_i \cdot \mathbf{pT}_j$, and so $\mathbf{I}(\mathbf{C},\mathbf{T}) = 0$. There is, however, no upper constraint on mutual information. Extending on the previous expression, we see that $\mathbf{I}(\mathbf{C},\mathbf{T}) = \mathbf{H}(\mathbf{C}) - \mathbf{H}(\mathbf{C},\mathbf{T})$

Using the conditional entropy, we obtain the two equivalent expressions:

$\mathbf{I}(\mathbf{C},\mathbf{T}) = \mathbf{H}(\mathbf{T}) - \mathbf{H}(\mathbf{T}|\mathbf{C})$

and $\mathbf{I}(\mathbf{C},\mathbf{T}) = \mathbf{H}(\mathbf{C}) - \mathbf{H}(\mathbf{C},\mathbf{T})$

Finally, because $\mathbf{H}(\mathbf{C},\mathbf{T}) \geq 0$ and $\mathbf{H}(\mathbf{T}|\mathbf{C},) \geq 0$

Given the inequality formulations $\mathbf{I}(\mathbf{C},\mathbf{T}) \leq \mathbf{H}(\mathbf{C})$ and $\mathbf{I}(\mathbf{C},\mathbf{T}) \leq \mathbf{H}(\mathbf{T})$.

By taking into account the ratios, we may derive a normalized version of mutual information.

$\mathbf{I}(\mathbf{C},\mathbf{T})|\mathbf{H}(\mathbf{C})$ and $\mathbf{I}(\mathbf{C},\mathbf{T})|\mathbf{H}(\mathbf{T})$, The geometric mean of these two ratios is known as the normalized mutual information (NMI):

$\mathrm{NMI}(\mathbf{C},\mathbf{T}) = \sqrt{\frac{\mathbf{I}(\mathbf{C},\mathbf{T})}{\mathbf{H}(\mathbf{C})} \cdot \frac{\mathbf{I}(\mathbf{C},\mathbf{T})}{\mathbf{H}(\mathbf{T})}} = \frac{\mathbf{I}(\mathbf{C},\mathbf{T})}{\sqrt{\mathbf{H}(\mathbf{C}) \cdot \mathbf{H}(\mathbf{T})}}$

The NMI value is between 0 and 1, with a higher value indicating a better clustering outcome.

3. **Matching Based Measures**

   - **Purity**

   Purity measures how much of a cluster $\mathbf{C}_i$ includes items from just one partition. In other words, it assesses the "purity" of each cluster. Cluster purity $\mathbf{C}_i$ is defined as [89], $\mathbf{P}_i = \frac{1}{n_i} \max_{j=1}^{k} \{n_{ij}\}$.

   Where $\mathbf{P}_i$ denotes the number of items in i that have the class label j. In other words, $\mathbf{P}_i$ indicates a proportion of the total cluster size represented by the greatest class of objects assigned to that cluster. The clustering solution's total purity is calculated as the weighted sum of the various cluster purities and is reported as:

   $\mathrm{Purity} = \sum_{i=1}^{r} \frac{n_i}{n} \mathbf{P}_i = \frac{1}{n} \sum_{i=1}^{r} \max_{j=1}^{k} \{n_{ij}\}$.

   Where n_i signifies the size of cluster i; r is the number of clusters, and n is the total number of items; The proportion of points in cluster $\mathbf{C}_i$ is denoted by the

ratio $\frac{n_i}{n}$. The alignment with the ground truth is better when the purity of $\mathbf{C}$ is higher. The greatest value of purity is 1 can be achieved when each cluster consists of points from just one partition. A purity score of 1 indicates flawless clustering with a one-to-one connection between clusters and partitions for $r = k$. The purity can be 1 when each cluster is a subset of a ground-truth partition. Because at least one cluster must comprise points from more than one partition, purity can be 1 even when $r > k$. When $r < k$, purity can never be 1 since at least one cluster must include points from more than one partition.

2.7 Conclusion

For various machine learning paradigms, especially the clustering task, exploiting multi-view data may provide extra information and significantly improve performance. Several multi-view clustering studies have focused on image, audio, and text. Recently, genetic (medical) and social-based data have attracted many researchers. Different multi-view fusion phases are present for these challenges. However, none of these stages is considered a prior or superior because the clustering goal is also related to the mathematical design and large-scale efficiency. In contrast to the terminology of multi-view analysis, we got onto an exciting theoretical concept about learning in feature space, including Kernelization and hashing techniques. The main advantage of operating in a kernel-defined feature space is to address the non-linearity problem, which is inherently present in large-scale datasets, by mapping our data points onto a higher-dimensional space without explicitly representing them (the kernel trick). The second blessing concerns the super dimensionality reduction strategy known as hashing function learning or binary code learning. It is recognized as an advanced indexing technique that can significantly increase performance and memory savings. The real-value features are compiled into a compressed hash code with successful similarity preservation. We finalize this chapter by taking a holistic view of the common clustering performance evaluation known explicitly as the external measures; Accuracy, Normalized mutual information, and Purity.

Chapter 3

Literature review and Related Works

3.1 Introduction

Similar to Multi-view clustering (MVC) direction, multi-view representation, multi-view supervised, and multi-view semi-supervised learning methods are worth mentioning paradigms. An apparent parallel between them is that they all learn with multi-view data; however, they target different learning schemes. Multi-view representation aims to learn a unified compact representation for subjects from all the views, whereas MVC aims to achieve sample partitioning. Multi-view representational learning evokes three vital factors:

— The correlation aims to be maximized across multiple views.

— The consensus seeks to maximize the agreement between different learned representations on multiple views.

— The complementarity attempts to employ individual knowledge in each view to represent the data comprehensively.

Multi-view supervised and semi-supervised learning methods employ full or partial sample label information, whereas; label information is not included in MVC. In one way or another, MVC methods are multifaceted and may exploit and/or adapt some integration strategies related to previous modes to accomplish their mission.

3.2 Taxonomy of MVC models

Based on the taxonomy of MVC, it can be categorized into two big classes: generative and discriminative. In this work, the fact that we do not touch on the generative category does not mean that it is less important, only because the context of our study is limited to the discriminative approaches. Generative methods assume that each group comes from a specific distribution in each view and group them for an MVC procedure, the details of which can be found in the earlier survey [90]. Discriminative methods directly optimize an objective function that minimizes the average intra-cluster similarity and maximizes the average inter-cluster similarity. Depending on the common property of different similar structures shared, a plethora of discriminative clustering methods can be divided into three main classes: (1) a common eigenvector matrix (mainly multi-view spectral clustering), (2) a common coefficient matrix (mainly multi-view subspace clustering), and (3) a common indicator matrix (mainly multi-view non-negative matrix factorization clustering). Figure 3.1 shows a taxonomy diagram of MVC models.

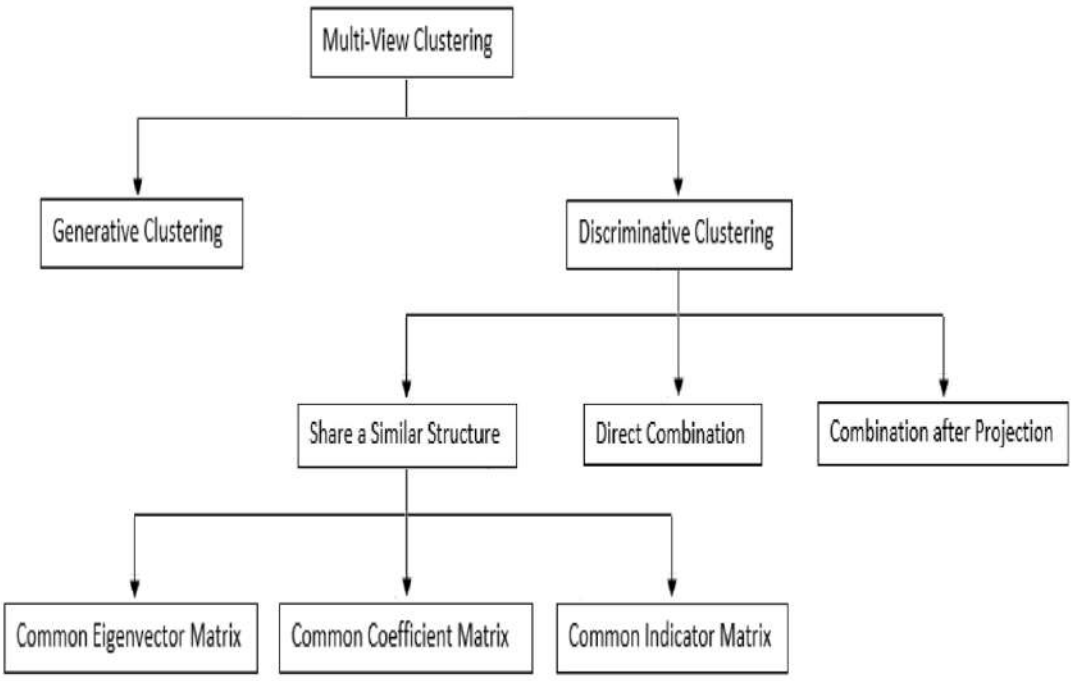


Figure 3.1 – The diagram of multi-view clustering models.

A) **Common Eigenvector Matrix (Mainly Multi-View Spectral Clustering):** Multi-view spectral clustering methods, also known as graph-based algorithms,

learn the data similarities from multiple views and achieve a common clustering result by assuming that all the views share the same or a similar eigenvector matrix. [91]. Before addressing different methods, we will define spectral clustering first. 1) Spectral clustering is a technique of clustering that operates on graph Laplacian properties. First, the graph is constructed, and its edges denote the similarities between data points, where a relaxed normalized min-cut problem is solved [92]. Different from the other popular methods, such as the k-means algorithm, which only fits spherical-shaped clusters, spectral clustering can be applied to arbitrary-shaped clusters, and its good performance has been demonstrated. Most multi-view spectral clustering methods are subject to two principal stages: 1) similarity graphs constructed from multi-view data; 2) spectral clustering applied to the similarity graphs to acquire the final partition result. The first stage conveys an approximation of the relationship and correlation between data samples. The features of multi-view data are described in terms of diversity, redundancy, and correlation due to diverse sources of data [93]. Hence, leveraging multi-view information to construct the similarity graph and promote the clustering performance in multi-view spectral clustering becomes a crucial problem. The second stage concentrates on obtaining more accurate discrete assignment results from the pre-constructed similarity graphs. Although several spectral clustering methods suffer from the drawbacks of two-phase clustering, many strategies were followed to meet the clustering results from the pre-constructed multi-view similarity graphs; particularly, spectral embedding pursued by k-means. However, such a two-step procedure generates loss and diminishes the clustering quality. Assuming the inner product of an embedded matrix is a low-rank approximation of a similarity graph, this represents a partial clustering structure with less noise in the similarity graph. Therefore, combining the inner product of the embedded matrix and the similarity graph can produce better results. Moreover, the noise and outliers are inherent in the originally constructed graph; one of the multi-view clustering solutions is to perform feature selection. In [94], they have utilized l21-norm regularization to mitigate the effect of noise and outliers. Following state-of-the-art

methods, co-training and co-regularization are two basic implementations of multi-view spectral clustering. Their common goal is to make multiple-view embeddings consistent. Co-training mutually modified the Laplacian matrix, assisted by the partitioning results of the other views [95]. Co-regularized utilizes a linear kernel to minimize the disagreement between different spectral embeddings [96]. Another approach to valuing the contribution of each view is by adding weight parameters; however, these extra parameters are manually tunable [97], [53]. Other methods in [98],[99],[100] adopt an auto-weighted learning strategy to get rid of the additional parameters drawback. The authors in [101] proposed a method dubbed 'Adaptively Weighted Procrustes' (AWP), which uses a spectral projection matrix with K-connected components from overall graphs to learn the cluster indicator matrix instantly. Two recent frameworks have been developed. The first approach in [102] integrates three learning components into a combined learning scheme: the similarity graph matrix, the unified graph matrix, and the clustering assignment by automatically assigning a weight for each graph matrix to attain the joint graph matrix. In addition, this method dictated a rank constraint on the Laplacian matrix to assemble exactly K clusters. The second approach in [103] assumes that the Laplacian matrix of each view is a perturbation of the consensus Laplacian matrix. So, first, it constructs a similarity graph for each view. The second step refers to consensus matrix learning, where weight is assigned to the Laplacian matrix of each view. Then, under the spectral perturbation theory, the clustering ability between each selected view and the consensus-clustering matrix is minimized to obtain the final predicted labels. In [104], the authors designed two auto-weighted multi-view clustering techniques that managed kernelized graph learning. The first approach studied the mapping of data into a feature space where they are linearly separable using a single kernel. The second is characterized by its capacity to handle multiple kernel matrices, capture complementary information from different views, and force the similarity matrix S to be unified. In contrast, the performance of the last technique is often sensitive to the type and parameters of each input kernel. Furthermore, a novel one-step graph-based multi-view clustering framework

has been implemented recently without any additional parameter [105] to explore non-negative embedding (cluster label space) under the assumption that "if two samples that have low similarity in the data space may have high similarity in the cluster label space." Therefore, they introduced an extra graph using cluster label correlation technique to the graphs associated with the data space; Therefore, they introduced an extra graph using cluster label correlation technique to the graphs associated with the data space; the smoothness of the cluster labels over all graphs has been imposed, and the clustering result has directly provided without a post-processing step which can be used as a cluster indicator matrix to perform the final cluster assignment without further post-processing steps such as spectral rotation or k-means.

B) **Common Coefficient Matrix (Mainly Multi-View Subspace Clustering):** Even if the given data for many practical applications is high-dimensional, the latent dimension of the problem's scope could be higher. For example, a given image holds many pixels, whereas only a few units are highlighted to describe a scene's geometry, appearance, and dynamics. This drives us to explore the latent low-dimensional subspace, considering that the data could be sampled from multiple subspaces. Multiview subspace clustering [106] is the class of methods that take care of learning a unified self-representation from multiple high-dimensional subspaces while assuming that different views share the same underlying representation. This joint information will be introduced as input to the clustering model to derive the grouping results. Over the past few years, subspace-based multiview clustering approaches have gained great attention due to their mathematical interpretability. The key challenges for subspace-based methods are in the robust multiview learning ability; hence, multiple perspectives have been set forth to solve the problem efficiently, such as LMSC [107], ECMSC [108], and CSMSC [109]. Despite proving its effectiveness in handling several computer vision and pattern recognition concerns, the aforenamed methods show three critical points: (i) learning the subspace structure in the original space, which unfortunately misses

out on the valuable nonlinearity structure of multiview data. (ii) disregard the linkage between subspace clustering and affinity matrix learning and treat them independently, which may not perfectly figure out the latent representations within data samples. Finally, (iii) most models react equally to different views, which are badly affected by the low-quality views in multiview data. Aiming to relieve the above three issues, Maria et al. [110] considered the low rank and sparse representation needed to fulfill multiview subspace clustering. Wang et al. [111] presented an angular-based similarity to estimate the consensus correlation in multiple views. Wang et al., in [25], adopted a similar concept to combine multiview information while associating a multigraph regularization with each graph Laplacian to characterize the view-dependent nonlinear data similarity. Finally, Zhang et al. [112], combine each view representation with a neural network and linear correlation. Different from the above approaches, the three works [113], [114], [115] treated the problem of incomplete information and adopted general NMF formulation to achieve unified representation for the samples while preserving the specificity of each view. Diversity-induced Multiview Subspace Clustering (DiMSC) [116] analyzes complementary information within the self-representation scheme based on the Hilbert-Schmidt Independence Criterion. In the same direction as diversity induction, Tao et al. [117] proposed extra loss terms with regularization subject to the contribution of consensus learning among different views. The method in [118] merges different views to attain directly a common indicator matrix rather than a joint subspace representation. Beyond these works, Chen et al. [119] addressed the problem of subspace clustering in a one-step optimization strategy and directly learned the non-negative transition probability matrix to ensure its optimality.

C) **Common Indicator Matrix (Mainly Multi-View Non-negative Matrix Factorization Clustering):** NMF is customarily used in clustering. It is conceived to split a given matrix into the bases matrix and the indicator matrix whose nonzero entry points whether the sample belongs to which cluster [8]. For multiple views, forcing the indicator matrix to be the same or similar is the optimal

way to accomplish the MVC task. Since the NMF strategy does not care about preserving the geometrical structures of the data from complementary view spaces during factorization, a big challenge was raised on how to split the data and earn a meaningful clustering solution. Akata et al. [27] extracted a consensus indicator matrix from two views to conduct MVC-based NMF. However, the view-specific and consensus indicator matrices might not be comparable at the same scale. Liu et al. [11] carried out the disagreement measurement between the coefficient matrix of each view and the consensus coefficient matrix toward a common indicator matrix. A view-dependent coefficient matrix normalization has been constrained, inspired by the connection between the NMF and the probabilistic latent semantic analysis. Cai et al. [120] proclaimed a k-means-based multi-view clustering method by optimizing the L2,1 norm objective to alleviate the effect of outliers in original data, together with weighting assignment to draw the importance of information from different views. Xu et al. [121], introduced an individual mapping matrix for each view and controlled the clustering model by promoting the common indicator matrix. Furthermore, the Frobenius norm has been replaced with an L2 norm, and each view-importance has been approximated. Recently, Liu and Fu [122] presented a categorical utility function to measure the similarity between the two partitions, the indicator matrix from each view, and the unified indicator matrix. Generating basic partitions from each view and the fusion of consensus clustering have been designed interactively in a one-step framework, and high-quality basic partitions positively contribute to consensus clustering. However, NMF remains unqualified to hold onto the intrinsic structure of data space. This prompted most researchers to incorporate other techniques to satisfy the manifold learning scheme efficiently. Cai et al. [123], [124] investigate the geometrical structure of data using a graph-based NMF method; therefore, a similarity matrix has been constructed with the parts-based data representation concept. Zhang et al. [125] combined the consensus manifold and the joint coefficient matrix learning to maintain the local geometrical structure of data space. Wang et al. [126] integrated a manifold regularization term into a concept factorization to preserve the geometrical information.

Further, Pu et al. [127] embraced the L2,1 norm with manifold regularization to regularize the matrix factorization and mitigate the impact of outliers; this makes the algorithm largely discriminative for multi-view data clustering. Motivated by recent progress in discrete hashing techniques, Zhang et al. [18] deployed the first meaningful work that addresses large-scale binary multi-view clustering problems. By leveraging the view-specific feature information, the unified learning model has founded upon the conversion of the kernelized features into a common compact binary code powered by the weighting vector built-in additional adjustable parameter to draw the importance of each view. Meanwhile, the binary clustering structure learning has been accomplished. Finally, as an alternative to previous work, Zhang et al. [19] have jointly learned a common binary representation by decomposing each projection into a combination of shareable and individual projections across multiple views to capture the underlying correlations; the latter can greatly improve the computational efficiency and robustness of clustering.

D) **Other MVC Methods:**

CCA-Based MVC (View Combination after projection) Canonical correlation analysis (CCA) was a commonly used technique to solve the problem of clustering analysis in high dimensions and data integration from different data types by introducing combinations after projection. For example, for two views of the same data, the target was to find two projections to maximize the correlation between them. Further, the CCA problem has transformed into a distance minimization problem, which is widely used in many works [113], [115], [128]. Chaudhuri et al. [129] stated the assumption that multiple views are uncorrelated conditioned on which mixture component generated the views; thus, the low-dimensional subspace spanned using the component distributions has been approximated via CCA projection and partitioning was conducted using single linkage clustering. Blaschko et al. [130] offered a correlational spectral clustering based on kernel canonical correlation analysis. First, the data were projected onto the top directions obtained by the KCCA across different views, and then a conventional k-means method was

applied to the projected space. Liu et al. [131] introduced tensor decompositions to analyze the latent pattern (cluster structure) hidden in multi-view data. Based on the assumption that the exemplar of a cluster in one view is always an exemplar of that cluster in the other views. They formed a similarity tensor from the similarity graph for each view, then performed a spectral analysis based on the obtained joint dominant subspace. Afterward, k-means is run to acquire the cluster indices. Following a similar previous assumption, Zhang et al. [132] proposed a multi-view and multi-exemplar fuzzy clustering method demonstrating performance improvement against a single-view clustering.

Multi-kernel-based MVC (Direct view combination) Canonical correlation analysis (CCA) was a commonly used technique to solve the problem of clustering analysis in high dimensions and data integration from different data types by introducing combinations after projection. For example, for two views of the same data, the target was to find two projections to maximize the correlation between them. Further, the CCA problem has transformed into a distance minimization problem, which is widely used in many works [113], [115], [128]. Chaudhuri et al. [129] stated the assumption that multiple views are uncorrelated conditioned on which mixture component generated the views; thus, the low-dimensional subspace spanned using the component distributions has been approximated via CCA projection, and partitioning has been conducted using single linkage clustering. Blaschko et al. [130] offered a correlational spectral clustering based on kernel canonical correlation analysis. First, the data were projected onto the top directions obtained by the KCCA across different views, and then a conventional k-means method was applied to the projected space. Liu et al. [131] introduced tensor decompositions to analyze the latent pattern (cluster structure) hidden in multi-view data. Based on the assumption that the exemplar of a cluster in one view is always an exemplar of that cluster in the other views. They formed a similarity tensor from the similarity graph for each view, then performed a spectral analysis based on the obtained joint dominant subspace. Afterward, k-means is run to acquire the cluster indices. Following a similar previous assumption, Zhang et al. [132] proposed a multi-view and

multi-exemplar fuzzy clustering method demonstrating performance improvement against a single-view clustering.

Deep-Based MVC In an early survey article [133], inspired by embedding-based architectures, multi-view representation techniques have been classified broadly into two categories: shallow methods and deep methods. The shallow methods mainly focus on the traditional feature extraction schemes in the context of multi-view learning, while the deep methods exploit deep visual and linguistic feature embedding. Several works affiliated with MVC are based on multi-view representation learning. For example, Huang et al. [14] performed an MVC implementation by using multiple-layer matrix factorization and sharing the same representation matrix across distinct views.

proposed using multiple-layer matrix factorization and shared the same representation matrix across different views to conduct MVC. This framework outperformed the state-of-the-art shallow clustering methods like co-training multi-view clustering, co-regularized multi-view clustering, and multi-view k-means clustering. Zhu et al. [134] designed a convolutional auto-encoders architecture to learn a multi-view self-representation matrix in an end-to-end manner with two sub-networks: a diversity network (Dnet) to learn view-specific self-representation matrices, and a universality network (Unet) to learn a common self-representation matrix for all views. Li et al. [135] proposed a deep MVC method inspired by a generative adversarial network (GAN). Based on adversarial learning and attention mechanism, Zhou et al. [136] presented an MVC method by combining GAN and attention mechanisms to align the latent feature distributions. It consists of three modules, modality-specific feature learning, modality fusion, and cluster assignment to guide the network training. Experiments support its effectiveness. Compared with traditional multi-view shallow clustering methods, the above multi-view deep clustering methods showed better performance, which may be attributed to several reasons summarized in two parts. First, deep networks adopted in multi-view deep clustering methods have better expressiveness and can capture the most realistic structure

of multi-view data. Second, most of them adopt an end-to-end multi-view deep clustering scheme. Thus, it can comprehensively reflect the representation reached, which serves the ultimate goal of efficient clustering.

3.3 Conclusion

This section formally described shallow methodologies and presented a diverse bird's-eye view of multi-view clustering approaches. Most of models failed to perform satisfactorily in the presence of noisy, redundant, and misleading views, while some achieved results that deserve attention and development. Furthermore, we looked at recent alternatives from deep approaches that have been considered in the literature. We would like to re-emphasize here that the developed strategies for various MVC algorithms have not been covered in depth, in order to facilitate access to information in a smooth form, and to give an opportunity for those who are not familiar with the mathematical concepts of machine learning to comprehend the basic building blocks and benefit from typical scenarios such as choosing a specific MVC method to process certain task.

Part II

Contributions

Chapter 4

Automatically Weighted Binary Multi-View Clustering via deep initialization (AW-BMVC)

4.1 Introduction

CLUSTERING , as subject of this study is one of the unsupervised learning approaches. The vast majority of partitioning techniques are only appropriate for single-view data [137] [138] [139]. Furthermore, combining all views into a single data matrix and then applying the most sophisticated clustering techniques to that matrix may not enhance clustering performance owing to information redundancy, which results in overfitting.

In contrast to conventional single-view clustering methods, multi-view clustering (MVC) approaches [90, 140], may be roughly categorised into three distinct groups: (1) Shared feature subspace (combination by projection), such as utilising Canonical Correlation Analysis (CCA) [129] to minimise cross-correlation error and then grouping the data with one of the clustering algorithms (e.g. k-means); (2) Multi-view spectral clustering (common eigenvector matrix [141] and/or common graph similarity matrix [142]), which generates numerous graphs to characterise the geometric structure, followed by data partitioning with one of the existing clustering algorithms; (3) Multi-view NMF clustering

(common indicator matrix) [18] based on matrix factorization by decomposing the feature matrix into a centroid matrix and a cluster indicator matrix.

Hashing techniques have become increasingly important for large-scale data analysis, resulting in quick computation and reduced memory needs [143, 16, 144]. Multi-view hash approaches have also evolved to encode high-dimensional feature vectors into binary low-dimensional codes, preserving the original space and allowing for multi-view information to be exchanged [143],[19]. Despite tremendous improvement in terms of quick computing and somewhat satisfying outcomes, the majority of extant multi-view learning algorithms suffer from the following three limitations:

1. The majority of current techniques use static or equal weights in conjunction with extra factors to evaluate the contribution of each view, resulting in sub-optimal representation learning.
2. In the process of clustering, most of the available models treat all samples identically.
3. The suggested procedures lack a practical and instructive initialization of the binary codes throughout the task of clustering, leading to a sub-optimum point.

In this research, a unique multiview clustering approach is devised to address the aforementioned issues: Auto-Weighted Binary Multiview Clustering via deep initialization (AW-BMVC). The main components of our contribution are as follows:

1. To take advantage of the heterogeneity of data with numerous views, we present an automatic weighting technique adapts pairwise relevance of samples and views, with view weight obtained from criterion term and sample weight explicitly determined.
2. We suggest an objective function the optimization of which enables the unified estimate of the following entities: joint binary code, view mapping, view/sample weights, binary centroids and cluster indicator matrix.
3. In order to acquire a solid initialization for the proposed optimization, Vgg16 network is mined for deep features, which are then transfer the features into the Hamming space and used to initialize the iterative clustering algorithm.

4. On the basis of the offered objective function and alternating optimization methodology, several state-of-the-art multiview clustering approaches, including those using real values, can be outperformed by the suggested method.

The following is the remainder of the chapter: **section 4.2** gives a thorough grasp of the intended task. Performance evaluation using numerous experiments is covered in **section 4.3**. In **section 4.4**, we conclude the chapter with a future study.

Table 4.1 – Summary of the main notations.

Notation	Description
n	Number of samples
c	Number of clusters
V	Number of views
m	Number of anchors
d_v	Data dimensionality for view v
$\mathbf{X}^1, ..., \mathbf{X}^M$ where $\mathbf{X}^v \in \mathbb{R}^{d_v \times n}$	A set of V data matrices
$\mathbf{x}_s^v$	s-th sample from the v-th view
$\mathbf{a}_1^v, \mathbf{a}_2^v, ..., \mathbf{a}_m^v$	A set of selected anchors from the v-th view
$\mathbf{\Phi}^v \in \mathbb{R}^{m \times n}$	Nonlinear Radial Basis Function mapping for view v
σ	Kernel width
$sgn(\cdot)$	Signum operator
$\|\|\cdot\|\|_F$	Frobenius norm
$Tr(\cdot)$	Trace of a matrix
$(\cdot)^T$	Transpose operator
$\mathbf{I}$	Identity matrix
$h(\cdot)$	Discrete hash function
l	Binary code length
$\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_n] \in \{-1, +1\}^{l \times n}$	The common binary codes of the n samples
$\mathbf{U}^v \in \mathbb{R}^{l \times m}$	The mapping matrix for the v-th view
α	View-weighting vector
$\mathbf{W} \in \mathbb{R}^{n \times n}$	Sample-weighting matrix (a diagonal matrix)
$\beta, \gamma, \lambda, \rho$	Regularization parameters
$\mathbf{C} \in \{-1, +1\}^{l \times c}$	Clustering binary centroids
$\mathbf{G} \in \{0, 1\}^{c \times n}$	Clustering assignment
$\mathbf{1}$	Column vector of ones

4.2 The Proposed approach

This section describes the novel multi-view clustering approach known as Auto-Weighted Binary Multi-View Clustering via deep initialization in detail. (AW-BMVC). It comprises two shared learning objectives: a common discrete representation powered by the auto-weighted sampling method and the auto-weighted view strategy. The global objective function is also simultaneously initialised using an excellent binary matrix representation. In brief, Figure 4.1 depicts the suggested framework's diagram.

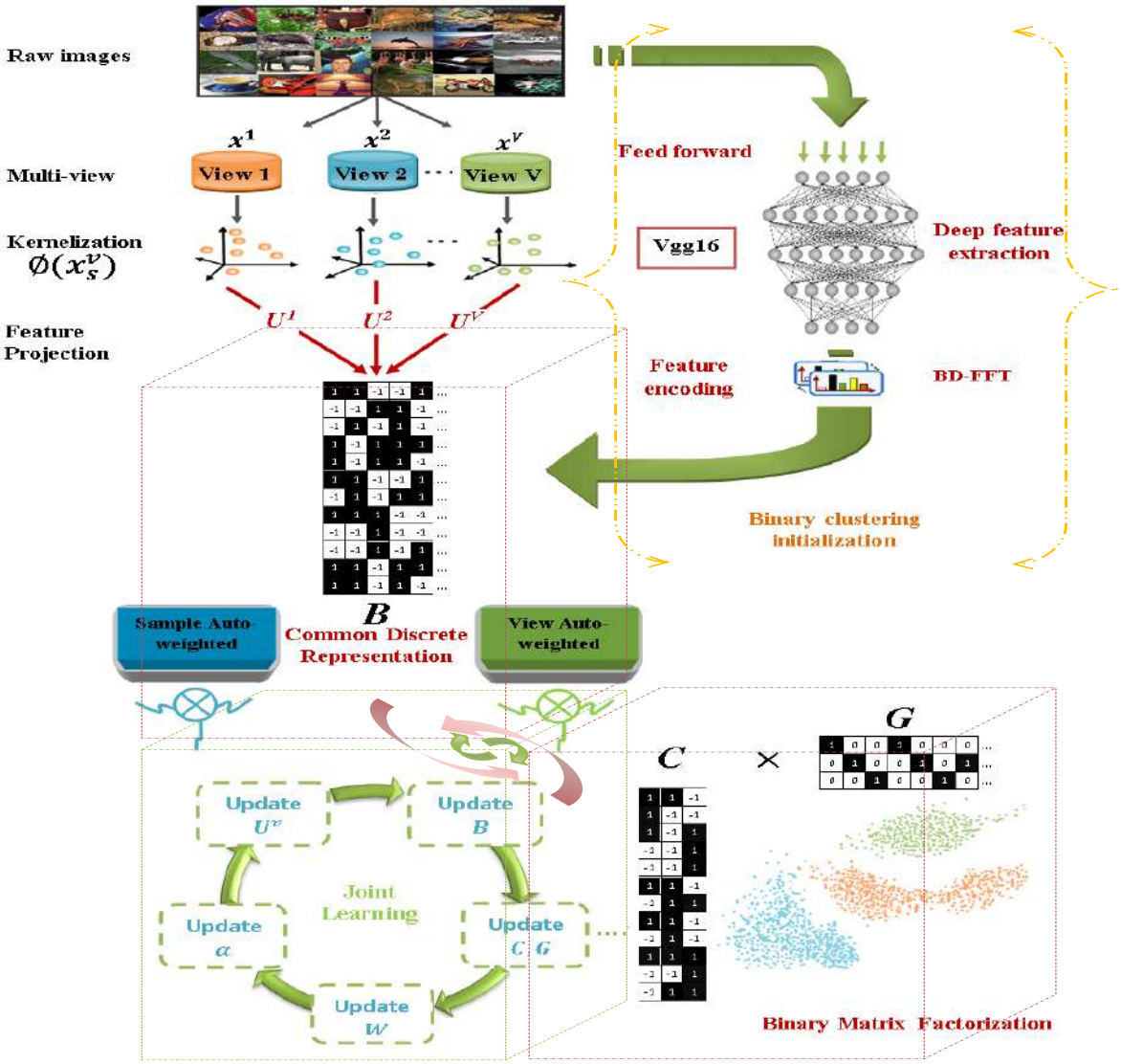


Figure 4.1 – The flowchart of the proposed method. Common discrete representation, Binary clustering initialization, Sample & view auto-weighting, and binary matrix factorization are integrated into a unified learning framework.

4.2.1 Anchor-based representation

Consider an RBF (Radial Basis Function) that can clearly organise several views into a single tensor of fixed dimensions and investigate the high-order latent structure inside each view by projecting them into a higher dimensional space.

Assume a multi-view dataset consists of V representations (i.e. V views) for n instances, that are identified by a collection of matrices $\{\mathbf{X}^1, ..., \mathbf{X}^V\}$; where $\mathbf{X}^v \in \mathbb{R}^{d_v \times n}$, is the data matrix of the v-th view, and d_v is the dimensionality of data features from the v-th view. The data samples in each view are considered to be zero-centered. i.e. $\sum_s \mathbf{x}_s^v = 0$, to keep the data balanced.

The first phase involves encoding data using non-linear RBF mapping. This encoding is determined by the mapping formulation below:

$$\mathbf{\Phi}(\mathbf{x}_s^v) = \left[exp\left(-\frac{||\mathbf{x}_s^v - \mathbf{a}_1^v||^2}{\sigma^v} \right), ..., exp\left(-\frac{||\mathbf{x}_s^v - \mathbf{a}_m^v||^2}{\sigma^v} \right) \right]^T \tag{4.1}$$

where σ^v is the kernel width for the $v\text{-}th$ view, $\mathbf{\Phi}(\mathbf{x}_s^v) \in \mathbb{R}^m$ denotes the m-dimensional nonlinear embedding of $s\text{-}th$ sample from the $v\text{-}th$ view, $\{\mathbf{a}_1^v, \mathbf{a}_2^v, ..., \mathbf{a}_m^v\}$ is a set of m selected anchors from $v\text{-}th$ view. Consider anchors to be statistically representative of the whole dataset. Instead of random sampling or K-means, these anchors are obtained using the K-medoids approach [62], which is more resilient to noise.

Remark:

Based on the reported experiments in [18], we set the number of chosen anchors for each view to $m = 1000$. It is also important to note that the variance of the kernel width parameter σ^v is critical as it defines the degree of smoothing. [145] and frequently manual inquiry is required. The empirical establishment of a universal adaptive scaling through each view is adjusted to the average of the Euclidean distances between the samples and their respective anchors.

4.2.2 Common discrete representation

The primary objective of our unsupervised technique is to accomplish direct clustering in a much lower-dimensional Hamming space by embracing common binary codes. In particular, multiple-view compression is accomplished. As a solution, hashing is a standard technique that has been frequently employed, for retaining similarity in a computationally efficient manner.

For each $\mathbf{\Phi}(\mathbf{x}_s^v)$ to be quantized into a discrete representation, we investigate the process of discovering a discriminative hashing function for each view as follows:

$$\min_{\mathbf{U}^v, \mathbf{b}_s} \sum_{v=1}^{V} \sum_{s=1}^{n} ||\mathbf{b}_s - \mathbf{U}^v \mathbf{\Phi}(\mathbf{x}_s^v)||^2 = \min_{\mathbf{U}^v, \mathbf{B}} \sum_{v=1}^{V} ||\mathbf{B} - \mathbf{U}^v \mathbf{\Phi}(\mathbf{X}^v)||_F^2 \tag{4.2}$$

$$\mathbf{b}_s = h_s^v(\mathbf{\Phi}(\mathbf{x}_s^v); \mathbf{U}^v) = sgn(\mathbf{U}^v \mathbf{\Phi}(\mathbf{x}_s^v)) \tag{4.3}$$

where $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is the common binary codes from different views (i.e., $\mathbf{x}_s^v, \forall v = 1, \ldots, V$), $\mathbf{\Phi}(\mathbf{X}^v)$ is the matrix nonlinear representation of all samples in view v, $\mathbf{\Phi}(\mathbf{X}^v) = [\mathbf{\Phi}(\mathbf{x}_1^v), \ldots, \mathbf{\Phi}(\mathbf{x}_n^v)]$, $\mathbf{U}^v$ is the mapping matrix, $sgn(\cdot)$ is the element-wise sign operator. Despite the fact that the model in Eq. (4.2) is linear, the entire projection from data space to common binary code space is nonlinear because of the nonlinear mapping $\mathbf{\Phi}(\mathbf{X}^v)$.

4.2.3 Sample-view auto-weighting

Recognizing that multiple views show the same topic from distinct measurements, the projection $\{\mathbf{U}^v\}_{v=1}^V$ should incorporate consensus information that optimises the similarity between diverse views as well as the dissimilarity that distinguishes particular features. In light of this, implicit automated view weighting will be implemented to define the link between views. Global optimization, on the other hand, will estimate explicit sample weighting factors. This technique enables the interchangeable highlighting of essential samples and the promotion of complementary information across various views, resulting in a comprehensive, consistent, uniform representation.

To advance, it is necessary, from an information-theoretic standpoint, to maximize the amount of information conveyed by each bit of binary code [146]. Using the maximum entropy principle [16], an extra regularizer is implemented for the binary codes $\mathbf{B}$ based on this notion. So, our aim is to maximize the variance of the matrix $\mathbf{B}$ provided by:

$$\begin{aligned} var[\mathbf{B}] &= \frac{1}{n}\sum_{v=1}^{V} var[\mathbf{U}^v \mathbf{\Phi}(\mathbf{X}^v)] = \frac{1}{n}\sum_{v=1}^{V} \|\mathbf{U}^v \mathbf{\Phi}(\mathbf{X}^v)\|^2 \\ &= \frac{1}{n}\sum_{v=1}^{V} tr((\mathbf{U}^v \mathbf{\Phi}(\mathbf{X}^v))(\mathbf{U}^v \mathbf{\Phi}(\mathbf{X}^v))^T) \end{aligned} \tag{4.4}$$

This extra regularization on $\mathbf{B}$ may provide a balanced partition and decrease the

duplication of the binary codes [19]. A relaxed regularization is constructed as a common discrete representation learning problem:

$$\min F(\mathbf{U}^v,\mathbf{B},\mathbf{W}) = \sum_{v=1}^{V}\left(||(\mathbf{B}-\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}_s^v))\,\mathbf{W}||_F^2+\beta||\mathbf{U}^v||_F^2-\frac{\gamma}{n}tr((\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))(\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))^T)\right) \\ s.t.\ \ \mathbf{B}\in\{-1,1\}^{l\times n},\sum_s w_s=1,w_s>0, \tag{4.5}$$

where β and γ are two regularization parameters.

The second term is a regularizer that determines the scales of the parameters (contribute to the stable solution). The diagonal sample-weighting matrix is denoted by $\mathbf{W}=diag(w_1,w_2,...,w_n)$. By determining the weights of the samples, the more significant ones will be given a large weight.

Inspired by previously suggested auto-weighted approaches [147], we offer a unique formulation in which no explicit view weight components are provided. Here, the preceding goal function is replaced with a new one which is the square root of the term to be minimized. Thus, the issue may be phrased as follows:

$$\min_{\mathbf{U}^v,\mathbf{B},\mathbf{W}} = \sum_{v=1}^{V}\sqrt{||(\mathbf{B}-\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v)\mathbf{W}||_F^2+\beta||\mathbf{U}^v||_F^2-\frac{\gamma}{n}tr(\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))(\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))^T)} \\ s.t.\ \ \mathbf{B}\in\{-1,1\}^{l\times n},\sum_s w_s=1,w_s>0 \tag{4.6}$$

Following the other multi-view algorithms, this kind of criteria implicitly assigns a weight to each view. Hence, minimizing Eq. (4.6) is the same as minimizing the following:

$$\min_{\mathbf{U}^v,\mathbf{B},\mathbf{W}} = \sum_{v=1}^{V}\alpha^v\left(||(\mathbf{B}-\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))\mathbf{W}||_F^2+\beta||\mathbf{U}^v||_F^2-\frac{\gamma}{n}tr(\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))(\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))^T\right) \\ s.t.\ \ \mathbf{B}\in\{-1,1\}^{l\times n},\sum_s w_s=1,w_s>0, \tag{4.7}$$

where the auto-weight α^v is given by the following expression:

$$\alpha^v = \frac{1}{2\sqrt{||(\mathbf{B}-\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))\,\mathbf{W}||_F^2+\beta||\mathbf{U}^v||_F^2-\frac{\gamma}{n}tr(\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))(\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))^T}} \tag{4.8}$$

4.2.4 Binary matrix factorization and overall objective function

AW-BMVC takes into account the decomposition of the acquired discrete representation $\mathbf{B}$ into two matrices, the binary clustering centroids $\mathbf{C}$ and the discrete clustering indications $\mathbf{G}$, in accordance to the relevant constraints:

$$\begin{aligned} &\min_{\mathbf{C},\mathbf{g}_s} ||\mathbf{b}_s - \mathbf{C}\mathbf{g}_s||_F^2 \\ s.t.\ \ &\mathbf{C}^T\mathbf{1} = 0, \mathbf{C} \in \{-1,1\}^{l\times c}, \mathbf{g}_s \in \{0,1\}^c, \sum_i^c g_{is} = 1 \end{aligned} \tag{4.9}$$

where $\mathbf{C}$ and $\mathbf{g}_s$ are the clustering centroids and the assignment vector for the sample s, respectively.

The balancing criterion is granted by the clustering centres restriction ($\mathbf{C}^T\mathbf{1} = 0$) in order to optimize the information submitted by each bit. Referencing Equation (4.9), we define the following factorization problem for all samples:

$$\begin{aligned} &\min_{\mathbf{C},\mathbf{G}} ||(\mathbf{B} - \mathbf{C}\mathbf{G})\mathbf{W}||_F^2 \\ s.t.\ \ &\mathbf{C}^T\mathbf{1} = 0, \mathbf{C} \in \{-1,1\}^{l\times c}, \mathbf{G} \in \{0,1\}^{c\times n}, \sum_{i=1}^c G_{is} = 1 \end{aligned} \tag{4.10}$$

Hence, the overall joint AW-BMVC is stated as follows:

$$\begin{aligned} \min F(\mathbf{U}^v, \mathbf{B}, \mathbf{C}, \mathbf{G}, \mathbf{W}, \alpha) = \sum_{v=1}^V \alpha^v (||(\mathbf{B} - \mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))\mathbf{W}||_F^2 + \\ \beta||\mathbf{U}^v||_F^2 - \frac{\gamma}{n} tr((\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}_s^v))(\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))^T) + \lambda||(\mathbf{B} - \mathbf{C}\mathbf{G})\mathbf{W}||_F^2 \\ s.t.\ \ \mathbf{C}^T\mathbf{1} = 0, \sum_s w_s = 1, w_s > 0, \\ \mathbf{B} \in \{-1,1\}^{l\times n}, \mathbf{C} \in \{-1,1\}^{l\times c}, \mathbf{G} \in \{0,1\}^{c\times n}, \sum_{i=1}^c G_{is} = 1, \end{aligned} \tag{4.11}$$

where λ is the regularization parameter.

Attention should be drawn to the existence of the sample auto-weighted matrix $\mathbf{W}$ for the binary clustering learning phase. This would make sense in terms of information

conservation and maintaining interrelation balance between discrete representation and binary clustering structure.

4.2.5 Optimization

Due to the discrete constraints and nonlinearity of the objective function, the solution to the problem (4.11), is essentially a challenging combinatorial optimization problem. Therefore, To deconstruct the subject into smaller subproblems and update it with rwspect to one variable while fixing the other variables, an alternating optimization strategy is performed. Consequently, we specify each step to iteratively update the mapping matrix $\mathbf{U}^v$, the discrete representation $\mathbf{B}$, the binary cluster centroids $\mathbf{C}$ and the indicator $\mathbf{G}$, the sample auto-weighting $\mathbf{W}$ and the view auto-weighting α^v, respectively.

- **Step 1: Update $\mathbf{U}^v, v = 1, ..., V$.**

By fixing other variables, the optimization formula for $\mathbf{U}^v$ is

$$\begin{aligned} \min F(\mathbf{U}^v) = ||(\mathbf{B} - \mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))\,\mathbf{W}||_F^2 + \beta||\mathbf{U}^v||_F^2 \\ - \frac{\gamma}{n} tr((\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))(\mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))^T \end{aligned} \tag{4.12}$$

We may derive the following solution by calculating the derivative of the objective function with respect to $\mathbf{U}^v$, and setting it to 0:

$$\mathbf{U}^v = \mathbf{B}\,\mathbf{W}\,\mathbf{W}\,\mathbf{\Phi}(\mathbf{X}^v)^T \cdot \mathbf{Q} \tag{4.13}$$

where $\mathbf{Q} = \left[\mathbf{\Phi}(\mathbf{X}^v)\mathbf{W}\mathbf{W}\mathbf{\Phi}(\mathbf{X}^v)^T - \frac{\gamma}{n}\mathbf{\Phi}(\mathbf{X}^v)\mathbf{\Phi}(\mathbf{X}^v)^T + \beta\mathbf{I}\right]^{-1}$.

- **Step 2: Update B.**

The optimization formula for $\mathbf{B}$ is

$$\begin{aligned}
\min_{\mathbf{B}} &= \sum_{v=1}^{V} \alpha^v \left(||(\mathbf{B} - \mathbf{U}^v \mathbf{\Phi}(\mathbf{X}^v))\mathbf{W}||_F^2 \right) + \lambda ||(\mathbf{B} - \mathbf{C}\mathbf{G})\mathbf{W}||_F^2 \\
&= \sum_{v=1}^{V} \alpha^v tr\left((\mathbf{B}\mathbf{W} - \mathbf{U}^v \mathbf{\Phi}(\mathbf{X}^v)\,\mathbf{W})^T (\mathbf{B}\,\mathbf{W} - \mathbf{U}^v \mathbf{\Phi}(\mathbf{X}^v)\,\mathbf{W}) \right) + \\
&\qquad \lambda tr\left((\mathbf{B}\,\mathbf{W} - \mathbf{C}\,\mathbf{G}\,\mathbf{W})^T (\mathbf{B}\,\mathbf{W} - \mathbf{C}\,\mathbf{G}\,\mathbf{W}) \right) \\
&\qquad = tr\left[\mathbf{B}^T \left(\sum_{v=1}^{V} \alpha^v \mathbf{W}\mathbf{W}^T + \lambda\, \mathbf{W}\mathbf{W}^T \right) \mathbf{B} \right] - 2 \\
&\quad tr\left[\mathbf{B}^T \left(\sum_{v=1}^{V} \alpha^v \mathbf{U}^v \mathbf{\Phi}(\mathbf{X}^v)\mathbf{W}\mathbf{W} + \lambda\, \mathbf{C}\,\mathbf{G}\,\mathbf{W}\,\mathbf{W} \right) \right] + \text{cons} \\
&\quad s.t.\ \ \mathbf{B} \in \{-1, 1\},
\end{aligned} \tag{4.14}$$

where *cons* indicates a constant value w.r.t. $\mathbf{B}$.

The solution for $\mathbf{B}$ is given by:

$$\mathbf{B} = sgn\left(\sum_{v=1}^{V} \alpha^v \mathbf{U}^v \mathbf{\Phi}(\mathbf{X}^v)\mathbf{W}\,\mathbf{W} + \lambda\, \mathbf{C}\,\mathbf{G}\,\mathbf{W}\,\mathbf{W} \right) \tag{4.15}$$

• **Step 3: Update C and G.**

Taking into consideration the discrete constraints, the regularised optimization formula for $\mathbf{C}$ and $\mathbf{G}$ is as follows:

$$\begin{aligned}
&\min F(\mathbf{C}, \mathbf{G}) = ||(\mathbf{B} - \mathbf{C}\mathbf{G})\mathbf{W}||_F^2 + \rho ||\mathbf{C}^T \mathbf{1}||^2 \\
&\quad s.t.\ \ \mathbf{C} \in \{-1, 1\}^{l \times c}, \mathbf{G} \in \{0, 1\}^{c \times n}, \sum_i g_{is} = 1,
\end{aligned} \tag{4.16}$$

Maintaining the discrete constraints throughout the optimization phase, we repeatedly optimize the cluster centroids using the Adaptive Discrete Proximal Linearized Minimization (ADPLM) method [144].

Update C.

With $\mathbf{G}$ fixed we have the following minimization problem:

$$\min F(\mathbf{C}) = -2tr\left[(\mathbf{B}\mathbf{W})^T (\mathbf{C}\mathbf{G}\mathbf{W}) \right] + \rho ||\mathbf{C}^T \mathbf{1}||^2 + \text{cons} \tag{4.17}$$

The following expression is the derivative of the obtained functional with regard to $\mathbf{C}$:

$$\begin{aligned} \nabla F(\mathbf{C}) &= -2\mathbf{B}\,\mathbf{W}\,(\mathbf{G}\,\mathbf{W})^T + 2\rho\,\mathbf{E}\,\mathbf{C} \\ s.t.\ \ \mathbf{C} &\in \{-1,1\}^{l\times c}, \end{aligned} \tag{4.18}$$

where $\nabla F(\mathbf{C})$ is the gradient of $F(\mathbf{C})$ and $\mathbf{E}$ is $l \times l$ square matrix of ones.

Based on the rule of ADPLM, we update $\mathbf{C}$ in the $p+1$-th iteration by

$$\mathbf{C}^{p+1} = sgn\left(\mathbf{C}^p - \frac{1}{\mu}\nabla F(\mathbf{C}^p)\right) \tag{4.19}$$

where $\frac{1}{\mu}$ is a step size. We set $\mu^p \in (L, 2L)$, where L is the Lipschitz constant.

Update G.

$$\min F(\mathbf{G}) = ||(\mathbf{B} - \mathbf{C}\,\mathbf{G})\mathbf{W}||_F^2 \tag{4.20}$$

Each column in $\mathbf{G} \in \{0,1\}^{c\times n}$ reflects the hard cluster assignment (i.e., the vector $\mathbf{g}_s$), for sample s which is provided by:

$$g_{is}^{p+1} = \begin{cases} 1 & i = \arg\min_k H(\mathbf{b}_s, \mathbf{c}_k^{p+1}) \\ 0 & otherwise \end{cases} \tag{4.21}$$

where $H(\mathbf{b}_s, \mathbf{c}_k)$ is the Hamming distance between the s-th binary code $\mathbf{b}_s$ and the k-th cluster centroid $\mathbf{c}_k$.

- **Step 4: Update the Sample weighting matrix W**.

$\mathbf{W}$ is the sample weight diagonal matrix. It is initialized by $w_1 = ... = w_s = ... = w_n = \frac{1}{n}$. It is updated using the following:

$$\begin{aligned} \min F(\mathbf{W}) &= \sum_{v=1}^{V} \alpha^v(||(\mathbf{B} - \mathbf{U}^v\mathbf{\Phi}(\mathbf{X}^v))\mathbf{W}||_F^2) + \lambda||(\mathbf{B} - \mathbf{C}\mathbf{G})\mathbf{W}||_F^2 \\ s.t. \sum_{s=1}^{n} w_s &= 1, w_s > 0, \end{aligned} \tag{4.22}$$

By adopting the following intermediate matrices, the loss function (4.22) is simplified:

$\mathbf{P}^v = [\mathbf{p}_1^v, ..., \mathbf{p}_n^v] = \mathbf{B} - \mathbf{U}^v \mathbf{\Phi}(\mathbf{X}^v)$

$\mathbf{M} = [\mathbf{m}_1, ..., \mathbf{m}_n] = \mathbf{B} - \mathbf{C}\,\mathbf{G}$

$$F(\mathbf{W}) = \sum_{v=1}^{V} \alpha^v \left(\sum_{s=1}^{n} w_s^2 ||\mathbf{p}_s^v||^2 \right) + \lambda \sum_{s=1}^{n} w_s^2 ||\mathbf{m}_s||^2 - \varepsilon \left(\sum_{s=1}^{n} w_s - 1 \right) \tag{4.23}$$

$$\frac{\partial F(\mathbf{W})}{\partial w_s} = 0 \implies \sum_{v=1}^{V} \alpha^v 2 w_s ||\mathbf{p}_s^v||^2 + 2\lambda w_s ||\mathbf{m}_s||^2 - \varepsilon = 0 \tag{4.24}$$

$$\implies 2\, w_s \left[\sum_{v=1}^{V} \alpha^v \, ||\mathbf{p}_s||^2 + \lambda \, ||\mathbf{m}_s^v||^2 \right] = \varepsilon \tag{4.25}$$

$$\implies 2\, w_s \, A_s = \varepsilon \tag{4.26}$$

where $A_s = \sum_{v=1}^{V} \alpha^v \, ||\mathbf{p}_s^v||^2 + \lambda \, ||\mathbf{m}_s||^2$

$$\implies w_s = \frac{\varepsilon}{2\, A_s} \tag{4.27}$$

$$\sum_{s=1}^{n} w_s = 1 \implies \varepsilon = \frac{1}{\sum_{s=1}^{n} \frac{1}{2\, A_s}} \tag{4.28}$$

$$\implies w_s = \frac{\frac{1}{\sum_{s=1}^{n} (\frac{1}{2 \cdot A_s})}}{2\, A_s} \tag{4.29}$$

• **Step 5: Update the View weight** $\alpha^v, v = 1, ..., n.$

These are initialized by $\alpha^v = \frac{1}{v}$, $\forall v = 1, \ldots, V$.

With fixed $\mathbf{U}^v$, $\mathbf{B}$, $\mathbf{W}$; α^v can be optimized using Eq. (4.8). Algorithm 1 summarizes the proposed framework.

Algorithm 1 : Auto-Weighted Binary Multi-View Clustering via deep initialization (AW-BMVC)

Input: Multi-view features $\mathbf{X}^v \in \mathbb{R}^{d_v \times n}$, and Selected anchors $\mathbf{A}^v \in \mathbb{R}^{d_v \times m}$, $v = 1, ..., V$, Parameters β, γ, λ, Number of clusters c, Number of iterations r & t, Length of binary codes l.

Output: Binary representation $\mathbf{B}$, Cluster centroid $\mathbf{C}$, Cluster indicator $\mathbf{G}$.

Initialization: Initialize view weights $\alpha^v = \frac{1}{V}$, Initialize sample weights $w_s = \frac{1}{n}$, Initialize binary representation $\mathbf{B}$ (see section (4.2.6)).
Compute anchor-based representation $\mathbf{\Phi}(\mathbf{X}^v), v = 1, ..., V$ using (4.1).
repeat
  Update $\mathbf{U}^v$ using (4.13).
  Update $\mathbf{B}$ using (4.15).
  repeat
    Update $\mathbf{C}$ using (4.19).
    Update $\mathbf{G}$ using (4.21).
  until *converge or reach "r" iterations*;
  Update $\mathbf{W}$ using (4.29).
  Update α using (4.8).
until *converge or reach "t" iterations*;

4.2.6 Binary clustering initialization

Our approach to the issue of iterative clustering is very dependent on the initial configuration of the factorable binary matrix. The consequence of poor initialization is perceived as the clustering algorithm being trapped in an undesirable local minimum.

Diverse Convolutional Neural Network (CNN) designs have better-recognized object characteristics than novel hand-crafted feature detectors. [148]. exploiting this concept, we introduce Bidirectional-Fast Fourier Transform (BD-FFT) method, which leverages Fourier decomposition to build effective representative codes. [149].

Using the dense feature representation from the pre-trained Visual Geometry Group model (VGG16), the initial job is to feed forward our image data collection and obtain features from the second FC layer (4096 neurons). Each of these neurons is sensitive to a certain feature, as shown by [150]. We regard each deep feature vector in this transformation as a one-dimensional signal. Using a bidirectional FFT, the second goal is to build a frequency domain representation as a series of sorted frequencies. On the basis of this concept, the coefficients corresponding to low frequencies were chosen and converted to binary codes. [151, 149], Using the mean of the frequency coefficients as the threshold (See figure 4.2).

The deep features are not employed as an extra view in the suggested criteria (4.11),

but rather to establish a decent initialization of the matrix **B**.

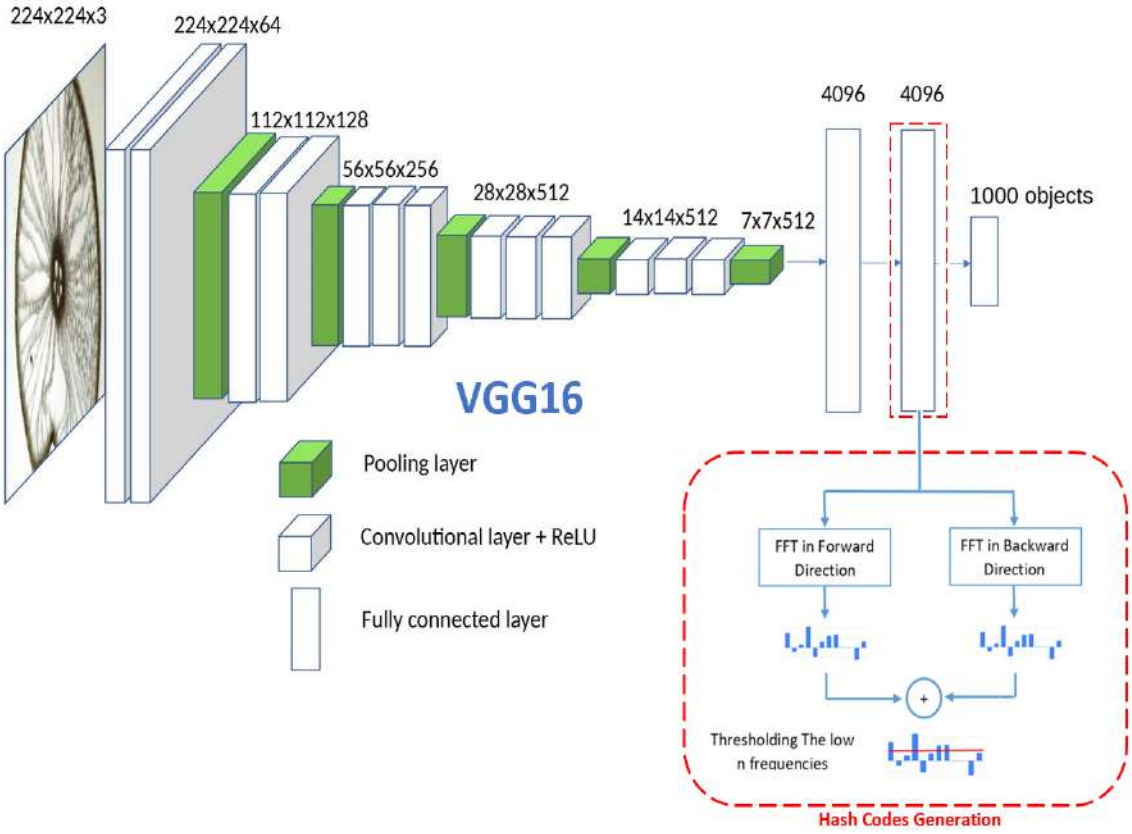


Figure 4.2 – Binary matrix generation for clustering initialization using BD-FFT.

4.3 Performance analysis

4.3.1 Experimental setup

4.3.1.1 Datasets

We conduct experiments on four publicly available multiview image datasets that are often used to evaluate clustering techniques, including **Caltech101-7**, **Caltech101-20** [1][152], **NUSWIDE-Obj** [2] [153], and **Scene-15** [3] [154]. To characterise each image, multiview features are extracted. Table 4.2, provides a comprehensive exposition of these datasets. The Caltech101 database comprises 9,144 images organised into 101 classes. Figure 4.3 shows a number of images from various classes.

We chose the extensively used seven-category object recognition dataset by tracing past work in [147]. i.e., motorcycle, face, Windsor, Garfield, chair, stop sign, Dolla-Bill,

1. *https://data.caltech.edu/records/20086*
2. *https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html*
3. *https://figshare.com/articles/dataset/15-Scene_Image_Dataset/7007177*

and Snoopy. 1474 images are compiled from the data to create the so-called Caltech 101-7. In addition, 2386 images belonging to 20 classifications were chosen. i.e., motorcycle, Garfield, face, Windsor chair, stop sign, Dolla-Bill, Snoopy, brain, leopard, camera, binoculars, ferry, camera, hedgehog, pagoda, car side, pagoda, yin-yang, water lily, rhino, wrench, and stapler. This dataset is referred to as Caltech101-20. Caltech101-7 and Caltech101-20 each include six distinctive features; namely, the 40-dim wavelet moments (WM), 48-dim Gabor feature, 254-dim CENTRIST feature, 40-dim wavelet moments (WM), 1984-dim HOG feature, 928-dim LBP feature and 512-dim GIST feature.

There are 30,000 images in NUSWIDE-Obj, spread among 31 classes. 4.4, shows an assortment of images.

This dataset utilises five common descriptors: a histogram (CH), 65-bin colour, 145-dim colour correlation (CORR), 226-dim colour moments (CM), wavelet texture (WT), and 74-dim edge distribution (ED).

Scene-15 consists of 4485 images organised into 15 categories of interior and outdoor scenes, including building, bedroom, living room, shop, industrial, seaside, kitchen, highway, office, inside city, mountain, woodland, suburb, open country, and street. Figure 4.5, shows a selection of images. From each image, features are taken to create three views.

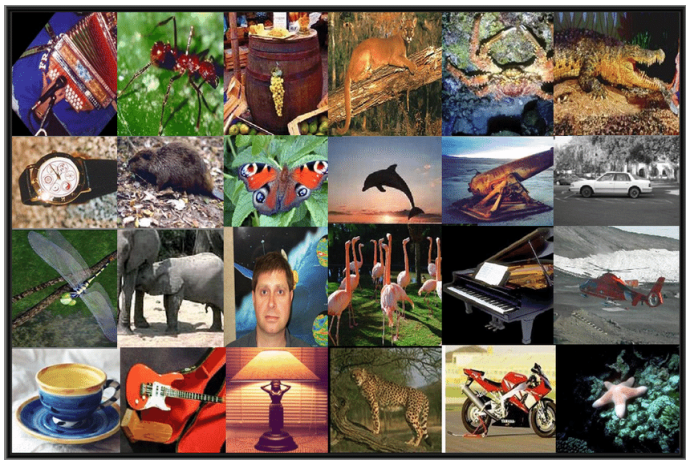


Figure 4.3 – Sample images(300×200 resolution) from Caltech-101 classes

4.3.1.2 Evaluation metrics and competitors

we verified the suggested methodology using the three most prevalent external assessment criteria: Accuracy (ACC), Normalized Mutual Information (NMI), and purity,

Figure 4.4 – Sample images(240x160 resolution) from NUSWIDE-Obj classes

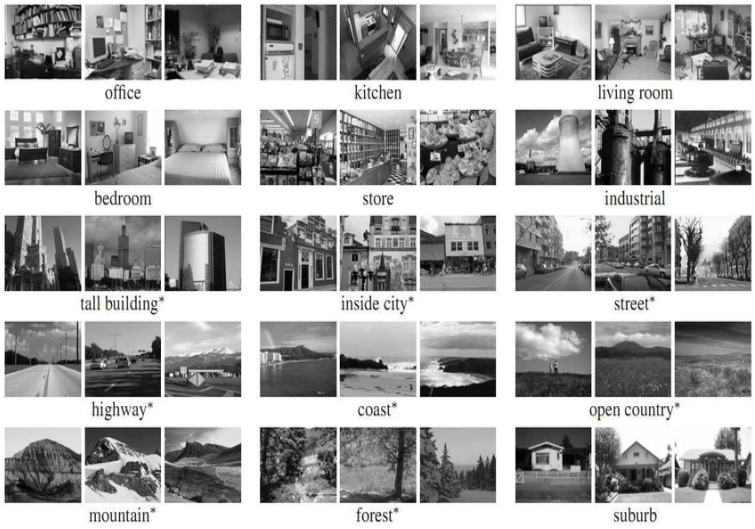


Figure 4.5 – Sample images(300×250 resolution) from 15scene classes

[155]. Moreover, Our suggestion is supported by a comparison of eleven state of the art algorithms: **(RMSC)**[156], **(DiMSC)**[116], **(AWP)**[101], **(WMSC)**[103], **(BMVC)**[18], **(OMSC)**[157], **(LMVSC)**[158], **(NESE)**[159], **(GMC)**[102], **(SMVSC)** [160], **(Co-FW-MVFCM)**[161].

We execute the comparison algorithms using the optimum parameter settings specified for each work.

Table 4.2 – Datasets used in our experiments. "dim" refers to the feature dimension.

Dataset	#Samples	#Views	Feature descriptors	#Classes
Caltech101-7/20	1474/2386	6	48-dim Gabor features 40-dim Wavelet moments 254-dim Centrist features 1984-dim HOG 512-dim GIST 928-dim LBP	7/20
NUSWIDE-Obj	30,000	5	65-dim Color Histogram 226-dim Color moments 145-dim Color correlation 74-dim Edge distribution 129-dim Wavelet texture	31
Scene-15	4485	3	20-dim GIST 59-dim PHOG 40-dim LBP	15

4.3.2 Parameter sensitivity

The proposed approach is parameterized such that its behaviour may be customized by adjusting three hyperparameters: β, γ, and λ. It is anticipated that these regularisation parameters would lead to a stable solution. By adjusting these factors, we investigated their impacts, λ to 1e-9 and empirically varying the values of β and γ from the grid {1e-5, 1e-4, 1e-3, 1e-2 1e-1 2, 4, 6, 10}.

Figure 4.6 depicts the variance in clustering accuracy for the four datasets and various *beta* and *gamma* settings.

Moreover, the sensitivity across the Caltech101-7/20 datasets is similarly dependent on the number of chosen anchors. which is advised to be fewer than one thousand anchors; this is due to the numerical perturbation that will be handled in the convergence study (section(4.3.6)).

Working with low *beta* values ($\beta = 1e-5$) and relatively large γ values ($\gamma = 10$), yields outstanding clustering results. The performance of clustering is reasonably steady while $1e-5 < \beta < 1e-2$; $2 < \gamma < 10$; beyond this range, we risk losing efficacy.

Table 4.3 – Best parameter tuning.

Datasets	β	γ	λ
Caltech101-7(20)	1e-05	10	1e-09
NUSWIDE-Obj	1e-05	2	1e-09
Scene-15	1e-05	10	1e-09

The optimal parameter settings for the three parameters are summarised in Table 4.3, which demonstrates that despite the sensitivity noted before, we got good clustering results for all datasets evaluated with a single tuning of γ within a short search range.

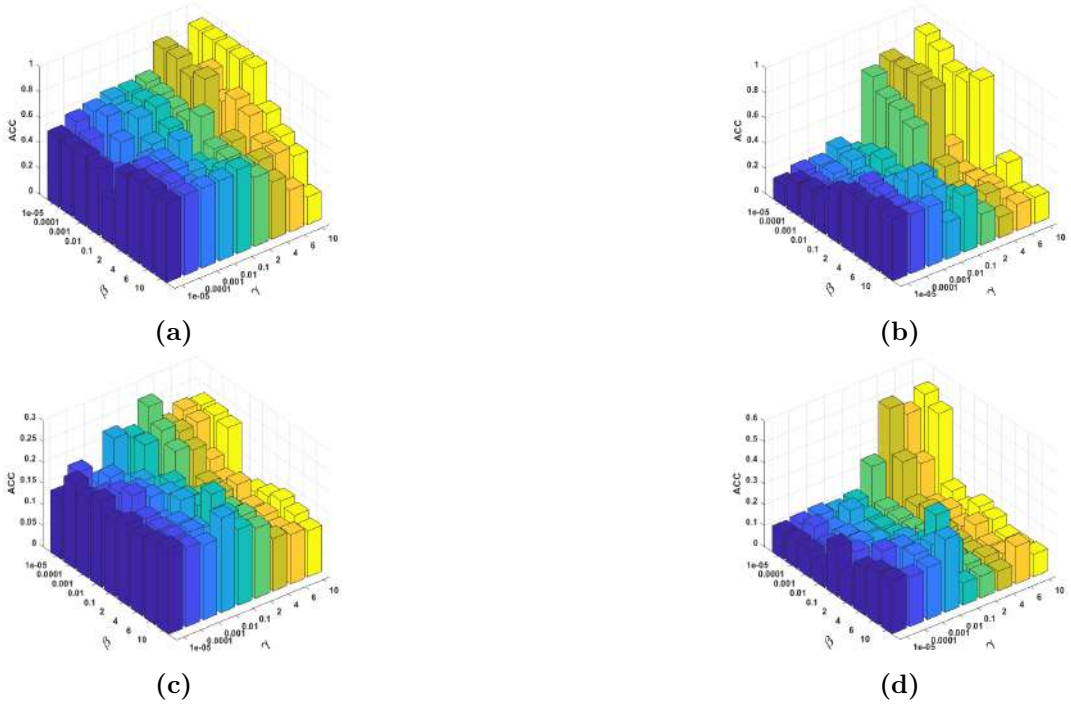


Figure 4.6 – Variability of accuracy with respect to β and γ parameters on: (a) Caltech101-7, (b) Caltech101-20, (c) NUSWIDE-Obj, (d) Scene-15

4.3.3 Computational complexity

In the suggested study, the subject of binary code learning was examined as an intriguing research strategy for overcoming the challenge of large-scale clustering of various views. The total complexity of AW-BMVC is $O(nlm^2V)t$ when six optimization operations are considered: $\mathbf{U}^v, \mathbf{B}, \mathbf{C}, \mathbf{G}, \mathbf{W}$, and α,. This computation exceeds the complexity of the preprocessing stage of deep feature extraction.

It may be noticed that $l << n$ and $m << n$ hold true. where n is the number of data samples and "t" is the number of iterations, and due to the quick convergence of the proposed model, its time complexity may be summed up as $O(n)$, which is proportional to n.

Our trials on running time are conducted on a computer with a 2.39 GHz Intel i5-2430M processor and 6 GB of Memory. In Table 4.4, we provide the execution timings of several multi-view techniques over Caltech101-7.

It is evident that DiMSC (355.77 *sec*), OMSC (107.55 *sec*), LMVSC (135.79 *sec*), SMVSC (236.32 *sec*) and Co-FW-MVFCM (1864.57 *sec*), compared to other techniques, are considerably time-consuming (greater than 100 seconds). As shown in the table, our method obtains clustering results for the Caltech101-7 dataset within 15.23 *sec* in just t= 3 iterations due to the self-weighted term of the samples, which somewhat increases the time cost. The AWP, WMSC, and BMVC approaches are more time-efficient, but our method produces the best clustering outcomes.

Table 4.4 – The running time (seconds) of different clustering approaches on the Caltech101-7 dataset.

Method	Time(Sec)	Method	Time(Sec)
RMSC-2014	92.08	**LMVSC-2020**	135.79
DiMSC-2015	355.77	**NESE-2020**	63.18
AWP-2018	7.76	**GMC-2020**	92.81
WMSC-2018	7.73	**SMVSC-2021**	236.32
BMVC-2018	**6.18**	**Co-FW-MVFCM-2021**	1864.57
OMSC-2019	107.55	**AW-BMVC(Ours)**	15.23

4.3.4 Ablation study

We shorten our presented approach in three fundamental modules: automated sample weighting, automatic view weighting, and initialization of binary clustering. In Table 4.5, we detail the performance of two datasets when each module is deleted or added.

Observe that the combination of these three important elements results in the all-out proposal, the supremacy of which is shown by the boldface outcomes. In comparison, removing these components returns us to the BMVC architecture [18] (first row in Table 4.5).

We also observe that the BCI module plays a crucial role in enhancing clustering performance since combining the two forms of sample and view auto-weighted without considering the binary clustering initialization component cause a huge performance loss. Furthermore, separating either one of the auto-weighted elements from the binary clustering initialization diminishes the capacity of the model.

Table 4.5 – Ablation experimental results. SAW: Sample Auto-Weighted. VAW: View Auto-Weighted. BCI: Binary Clustering Initialization.

Removing or adding a component				Dataset			
	SAW	VAW	BCI	Caltech101-7	Caltech101-20	NUSWIDE-Obj	Scene-15
ACC				0.2856	0.2355	0.1680	0.2312
NMI	✗	✗	✗	0.1079	0.1864	0.1621	0.1466
Purity				0.5916	0.4392	0.2872	0.2580
ACC				0.2904	0.2921	0.1875	0.1739
NMI	✔	✗	✗	0.1645	0.2149	0.1082	0.1062
Purity				0.6832	0.4715	0.2383	0.1835
ACC				0.3209	0.3814	0.1336	0.2446
NMI	✗	✔	✗	0.2065	0.4932	0.1457	0.2062
Purity				0.7123	0.7318	0.2721	0.2999
ACC				0.5122	0.5159	0.1874	0.5032
NMI	✗	✗	✔	0.4935	0.6830	0.1935	0.4322
Purity				0.8718	0.8688	0.3081	0.5574
ACC				0.3141	0.2200	0.1695	0.2881
NMI	✔	✔	✗	0.1583	0.1898	0.1676	0.2080
Purity				0.6784	0.4484	0.2714	0.3097
ACC				0.4274	0.6144	0.1598	0.4330
NMI	✔	✗	✔	0.2746	0.4900	0.1552	0.3986
Purity				0.7822	0.6174	0.2811	0.4384
ACC				0.5102	0.4736	0.1853	0.4932
NMI	✗	✔	✔	0.4931	0.6659	0.1957	0.3564
Purity				0.8718	0.8395	0.3137	0.5462
ACC				**0.9022**	**0.8734**	**0.2190**	**0.5634**
NMI	✔	✔	✔	**0.8733**	**0.8180**	**0.2156**	**0.5089**
Purity				**0.9022**	**0.8873**	**0.3220**	**0.5884**

4.3.5 Clustering initialization analysis

In our suggested optimization technique, we ran a series of experiments on four datasets to evaluate the effect of various binary code initialization. Hence, in the algorithm 1, the matrix of binary codes *Bvect* was initialized using three distinct algorithms: a random binary matrix, non-linear PCA, and Deep-FFT. In the first plan, we construct a random binary matrix;The second scenario demonstrates a non-linear PCA methodology employed by the BMVC approach [18]; and the third approach is Deep-FFT, which is the initialization scenario we suggest. As demonstrated in Table 4.6, the binary random matrix, which is fully independent of the data, produces the poorest results. The non-linear PCA approach yields homogenous but suboptimal scoring measures. We infer that this is because the eigenvalue decomposition was performed on an embedded view. Specifically, compared to the previous two instances, we can infer that our starting strategy yields a considerable increase in clustering outcomes as a consequence of a new deep feature extraction that directly enhances the optimization procedure.

Table 4.6 – Clustering initialization study. RI: Random Initialization. PCA: One-view PCA Initialization. Deep: Deep-FFT Initialization.

Initialization scenario			
Dataset	**RI**	**PCA**	**Deep**
ACC			
Caltech-7	0.2863	0.2924	**0.9022**
Caltech-20	0.2393	0.2200	**0.8734**
NUSWIDE-Obj	0.1508	0.1956	**0.2190**
Scene-15	0.1445	0.2453	**0.5634**
NMI			
Caltech-7	0.0622	0.2103	**0.8733**
Caltech-20	0.1471	0.1898	**0.8180**
NUSWIDE-Obj	0.0785	0.1500	**0.2156**
Scene-15	0.0476	0.1536	**0.5089**
Purity			
Caltech101-7	0.5733	0.6934	**0.9022**
Caltech101-20	0.4510	0.4484	**0.8873**
NUSWIDE-Obj	0.2041	0.2634	**0.3220**
Scene-15	0.1521	0.2660	**0.5884**

4.3.6 Convergence analysis

Figure 4.7 illustrates the objective function value for every iteration across four datasets. Alternating iterative optimization is used to iteratively update each variable: the mapping matrix $\mathbf{U}^v$, the discrete representation $\mathbf{B}$, the binary cluster centroids $\mathbf{C}$, the cluster indicator matrix $\mathbf{G}$, the auto-weighted sample $\mathbf{W}$, and the auto-weighted view α. The subproblems $\mathbf{U}^v$ and $\mathbf{B}$ arising from Eq. (4.12) and Eq. (4.14) guarantee a closed-form optimal solutions given by Eq. (4.13) and Eq. (4.15), respectively. The subproblem $\mathbf{C}$ in Eq. (4.17) has an analytical solution using ADPLM [144], Eq. (4.19) effectively shows its optimal solution, followed by the obvious solution for $\mathbf{G}$ in Eq. (4.21), which is comparable to the K-means learning algorithm. The solution for automated sample weighting in Eq. (4.29) as well as the automatic weighting of views in Eq. (4.8) is the exact minimum points. Hence, the loss values of the globally adopted objective function have decreased $F(\mathbf{U};\mathbf{B};\mathbf{C};\mathbf{G};\mathbf{W};\alpha)$ in Eq. (4.11), rapidly drop and achieve the lowest point after around $t = 5$ iterations, as well as verify the monotonic bound, which is adequate for convergence.

Numerical perturbation on Caltech101-7/20: Particularly experimenting with the Caltech101-7/20 datasets revealed a second phenomena that may provide light on the stability challenge. A numerical disturbance caused a quick and transitory decline in the

objective function's value to its minimum. As the calculation of the mapping matrices $\mathbf{U}^v$ becomes ill-conditioned, a succeeding sharp increase and/or halt was noticed. With a small dataset such as Caltech-7 ($n = 1474$), the number of anchors may be constrained to a maximum. Hence, an experimental extension is accomplished. We resolve the prior perturbation problem by restricting the number of selected anchors to fewer than one thousand, where m=700 anchors are tested and verified (see Figure 4.8).

Figure 4.9 displays the clustering performance according to the number of anchors over the Scene-15 dataset. Clearly, the performance of clustering may be altered by this quantity. The accuracy ranges from 0.4939 (m=400) to 0.5666 (m=700). There are no exact criteria for finding the appropriate number of anchors. However, the number of samples plays a significant role. In Scene-15, for instance, clustering efficiency degrades if the number of anchors is fewer than 500. But, as m exceeds 500, performance peaks and becomes steady.

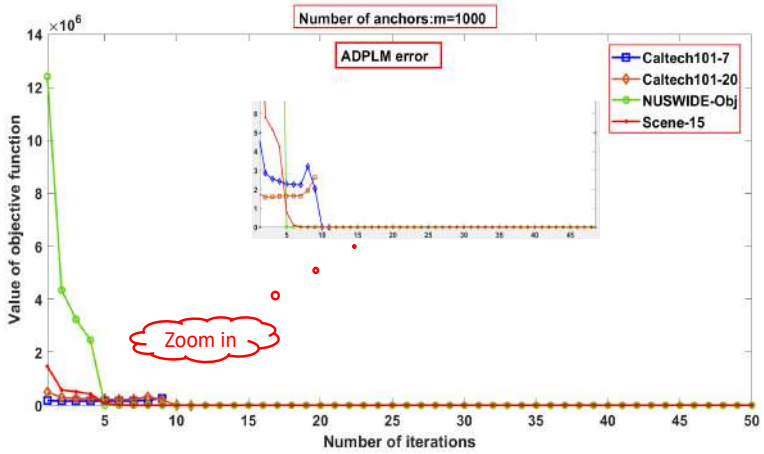


Figure 4.7 – Objective function as a function of iteration number on all datasets. The number of anchors m is set to 1000.

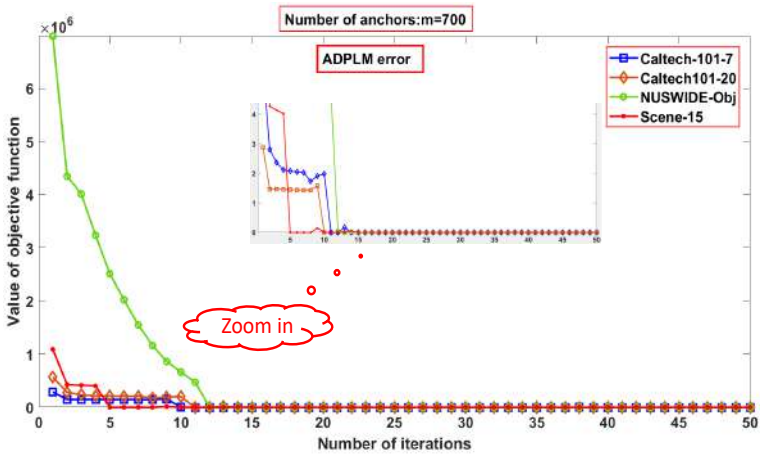


Figure 4.8 – Objective function as a function of iteration number on all datasets. The number of anchors m is set to 700.

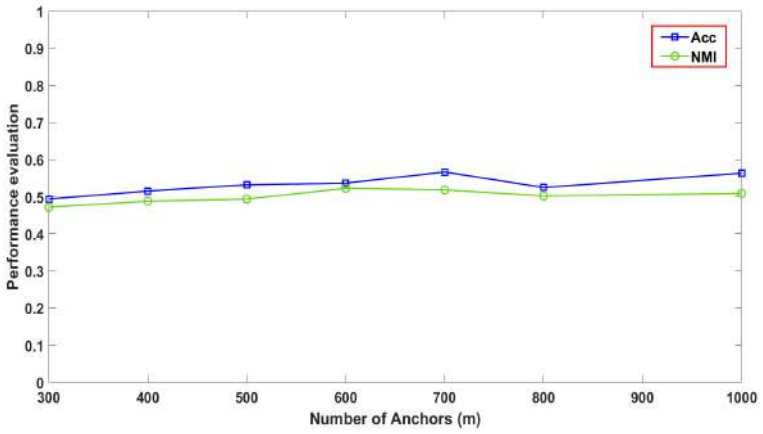


Figure 4.9 – ACC and NMI variation versus the number of anchors on the Scene-15 dataset.

4.3.7 Comparison with state-of-the-art multi-view methods

To demonstrate the superiority of the proposed algorithm, we conducted thorough trials using 11 state-of-the-art comparison approaches. Table 4.7, demonstrates the performance of all competing clustering algorithms for each of the four datasets. The best clustering performance in this table is shown in boldface.

According to Table 4.7, the OMSC technique has inconsistent performance, observable at very low ACC, in contrast to the better NMI and Purity for all datasets; additional adjustments may be required.

Based on the data shown in Tables4.4 and 4.7, the following conclusions may be drawn: Analytically, the SMVSC and LMVSC algorithms demand a considerable amount of processing time yet provide mediocre results. This is owing to the fact that graph

filtering in SMVSC and the anchor graph approach in LMVSC accomplish the smooth representation requirement. The three approaches WMSC, AWP, and DiMSC techniques yield the third-best outcomes, correspondingly. The first strategy has a shorter duration. It exposes a technique for decreasing the clustering ability of two sets of eigenvectors, the Laplacian for each view and the Laplacian of the consensus matrix. The second method employs Procrustes analysis, to assign clusters and skip eigenvalue decomposition at each iteration step, making it more efficient. The third technique is the second most time-consuming since it allows for the individual representation of each view through the original raw space. In addition, it may need extending the fusibility investigation towards a comprehensive diversity estimate. Even for small data sets, the NESE approach is time-efficient and yields accurate findings. It makes use of consistent non-negative embedding but need to handle the issue of heterogeneity across various views by describing the degree of contribution made by each view. BMVC is computationally highly efficient due to its simultaneous binary representation and clustering. Nevertheless, our suggested solution has successfully addressed three fundamental flaws with this paradigm (automatic view weighting, automatic sample weighting, and binary clustering initialization).

Due to the two independent phases of learning consensus graphs and clustering structures, RMSC performs poorly. The approach with the lowest performance is Co-FW-MVFCM, which has a relatively lengthy runtime owing to the a priori clustering of the individual views and the information exchanged between individual members. On top of that, the technical method of feature reduction by thresholding each view and the empirical exponent parameter used to manage the distribution of each view is insufficient.

In general, the majority of baselines perform better in clustering metrics for a particular dataset. BMVC, for instance, is superior for large datasets such as NUSWIDE-Obj. In contrast, WMSC is superior for the Scene-15 dataset, GMC for the Caltech101-7 dataset, and NESE for the Caltech101-20 dataset. In three assessment indices, AW-BMVC performs excellently on four benchmark image datasets: Caltech101-7, Caltech101-20, NUS-WIDE-obj, and Scene-15, and beats the results of the other approaches.

Table 4.7 – The clustering performance comparisons on challenging datasets. "-" indicates unavailable results due to out of memory.

	Caltech101-7			Caltech101-20		
Methods	**ACC**	**NMI**	**Purity**	**ACC**	**NMI**	**Purity**
RMSC-2014[156]	0.4037	0.3544	0.8026	0.4035	0.5073	0.7360
DiMSC-2015[116]	0.5611	0.4221	0.8318	0.4728	0.4935	0.7347
AWP-2018[101]	0.5685	0.4710	0.8554	0.4953	0.5590	0.7594
WMSC-2018[103]	0.5943	0.4960	0.8588	0.5310	0.5893	0.7682
BMVC-2018[18]	0.2856	0.1079	0.5916	0.2355	0.1864	0.4392
OMSC-2019[157]	0.0257	0.1770	**0.9545**	0.0255	0.3108	**0.9241**
LMVSC-2020[158]	0.7266	0.5193	0.7517	0.5306	0.5271	0.5847
NESE-2020[159]	0.4857	0.4614	0.8548	0.6085	0.6045	0.7556
GMC-2020[102]	0.6919	0.6056	0.8846	0.4564	0.3845	0.5549
SMVSC-2021[160]	0.7354	0.5204	0.8487	0.5692	0.5190	0.6442
Co-FW-MVFCM-2021[161]	0.4016	0.2819	0.7944	0.3051	0.3887	0.5746
Ours	**0.9022**	**0.8733**	0.9022	**0.8734**	**0.8180**	0.8873
	NUSWIDE-obj			**Scene-15**		
RMSC-2014[156]	0.1473	0.1421	0.2624	0.3482	0.3483	0.3797
DiMSC-2015[116]	0.1330	0.1363	0.2165	0.2555	0.2083	0.2758
AWP-2018[101]	0.1440	0.1123	0.2446	0.3429	0.3366	0.4035
WMSC-2018[103]	0.1382	0.1344	0.2475	0.4370	0.4341	0.4807
BMVC-2018[18]	0.1680	0.1621	0.2872	0.2312	0.1466	0.2580
OMSC-2019[157]	0.0678	**0.2530**	**0.4465**	0.0084	0.3133	**0.8403**
LMVSC-2020[158]	0.1181	0.1063	0.1363	0.3134	0.3297	0.3551
NESE-2020[159]	-	-	-	0.4312	0.4042	0.4822
GMC-2020[102]	0.1192	0.1128	0.1205	0.1400	0.1105	0.1464
SMVSC-2021[160]	0.1254	0.1123	0.1587	0.3583	0.3433	0.3861
Co-FW-MVFCM-2021[161]	0.1673	0.0913	0.2209	0.2856	0.2822	0.3257
Ours	**0.2190**	0.2156	0.3220	**0.5634**	**0.5089**	0.5884

4.4 Conclusion

We presented a large-scale approach called Auto-Weighted Binary Multi-View Clustering via deep initialization (AW-BMVC) to discover a common discrete representation of multi-view data while optimizing binary clustering based on matrix factorization. Thanks to the benefits of self-weighted samples and views as the initial component in this system, which demonstrate its capacity to differentiate between views based on significant samples and generate a comprehensive joint discrete representation. Regarding the clustering initialization problem, we have additionally emphasized a novel deep representation strategy. Consequently, our binary clustering initialization technique yielded final clustering with exceptional performance. Therefore, within a few iterations, rapid convergence was reached. In addition, empirical findings on a number of well-known datasets have validated our approach's superiority over other multi-view clustering techniques of the present day.

Nevertheless, for the sake of the scientific integrity of the suggested method, three associated shortcomings must be mentioned: (1) It is quite evident so long as the preferred number of anchors is fixed at 1000 samples; which makes the model selective as it cannot handle smaller datasets. (2) Despite efforts to have the model autonomously learn view and sample weights, the clustering performance is heavily dependent on manually configurable regularisation parameters (β,γ,λ). (3) The use of pre-trained deep VGG16 as part of the binary matrix initialization approach confines the evaluation's scope to just image datasets. Extending our findings to text datasets using versions of feasible binary clustering initialization techniques is seen as promising, however. It provides a remedy for one of the most critical identified flaws.

Chapter 5

General conclusion and Perspectives

5.1 General Conclusion

This thesis aimed to develop an efficient large-scale framework that can cluster multi-view data. Clustering is regarded as an important task in mining and data discovery. Before formalizing the problem, we started with the definitions and theoretical concepts related to data type domains and applications, including different data fusion and learning stages. With the knowledge that data is inherently nonlinear and may be large in terms of instances/features, and possibly carry some noise and missing values. Because of these factors, the mining process has become much more difficult, and many methods have been intensively proposed recently. We reviewed several state-of-the-art multi-view-based clustering techniques, broadly categorized into three main classes: Multi-View Spectral Clustering, Multi-View Subspace Clustering, and Multi-View NMF Clustering. We have seen that the primary building block in our approach was data kernelization, where we found a good representation of each raw-view data in terms of better structure understanding and linear separability. In this area, anchor-based representation is one major technique achieved by considering RBF similarity measures through a preselected subset. Following this line of thought, we targeted a particular and interesting case of learning in a feature space called hashing. By opting for the intermediate data fusion, a unified

discrete representation learning is accomplished using the projected kernelization data matrices and governed by two configurations, an automatically implicit view weighted and an automatically explicit sample weighted. We have realized that the decisive step in our clustering task is not the partitioning of the learned discrete representation based on the non-negative matrix factorization technique but a good initialization of the binary matrix that drove the joint learning algorithm at least to the best local optimum.

5.2 Perspectives

Through several experiments and encouraging results that have been reached in this thesis and continuing in the same research direction of multi-view learning and clustering, we can identify further interesting and challenging perspectives, including:

1. It is mentioned that our developed model could handle only image datasets since it employed a CNN for the binary matrix initialization. As a solution, we could adopt several scenarios of fusing by diffusion in a pre-constructed similarity graph to extend our work to the text, genes, or any other types of datasets, etc.

2. In contrast to the shallow approaches, it is very promising to exploit multi-auto-encoder architectures to handle an end-to-end multi-view clustering problem.

3. For multi-view learning, another interesting application topic is "outlier detection," which represents abnormal behavior, fraud detection, system health monitoring, contaminants, etc. Multi-view clustering could be helpful to complete the missing information and analyze hidden patterns.

Chapter 6

Appendix

6.1 Cluster Analysis

Clustering is the process of partitioning objects into different groups (clusters) according to their similarities and properties. Hence, more similar objects belong to the same group than others in other clusters. Typically, the similarity measure is an estimation process that influences the outputs based on the selected class of algorithms, including: Centroid-based models, connectivity-based models, density-based models, etc. In this project, we utilize the centroid-based algorithm. Below is a synopsis of each model and its similarity concept. For each algorithm, the clustering results will be described based on the original data represented in Figure 6.1a. The ground truth is illustrated in Figure 6.1b. The yellow data points are noise.

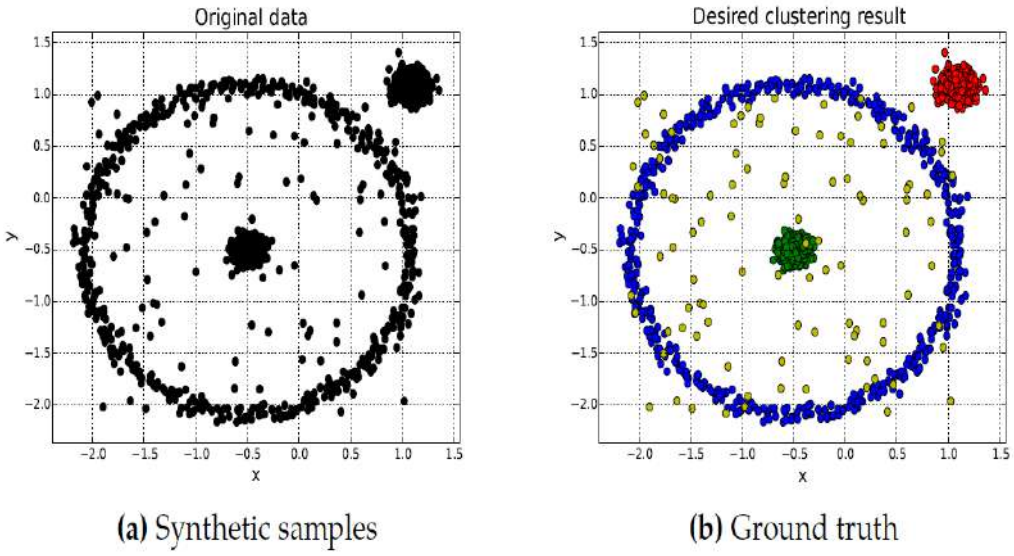


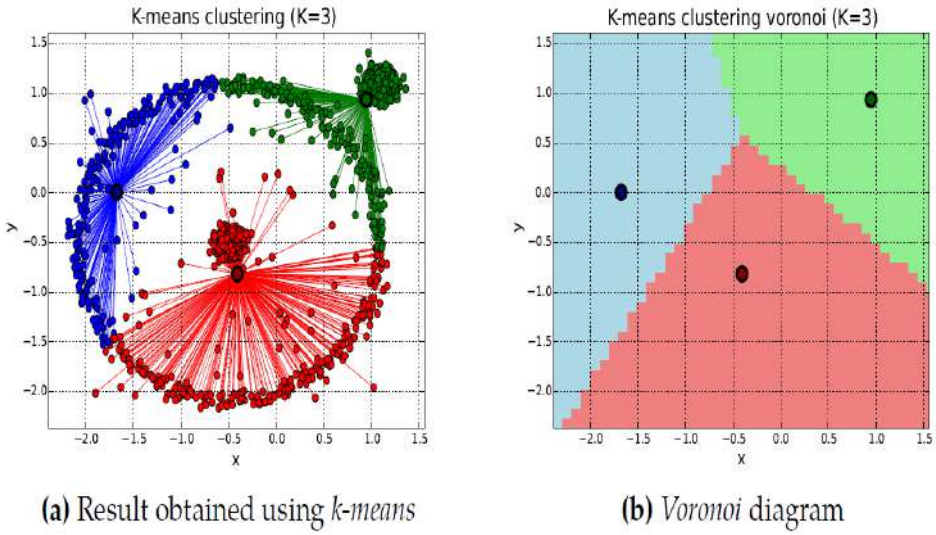
Figure 6.1 – Example of clustering: (a) Synthetic data; (b) Ground truth. Yellow datapoints are considered noise.

6.1.1 Centroid based clustering

The core idea behind this model is to find k sets of samples based on the proximity of representative points called centroids. Several algorithms have been developed based on the centroid selection/creation strategy. K-means initially defines the arbitrary centers, then assigns samples to the nearest center, and subsequently moves the coordinates of the centers according to the average of all belonging samples. Note that the centroid is not necessarily a sample of the dataset. Therefore, k-medoids constrain the centroid to be one sample of the dataset.

In contrast to the k-means, another algorithm called k-median is interested in the median instead of the average of the belonging set. The common drawback among all these algorithms is that the number of clusters k has to be predefined. The outlier influence and their cluster tendency are other hot topics to be considered.

For better illustration, we select the k-means algorithm (see Figure 6.2a). Therefore, the so-called Voronoi diagram represents a space partition in cells (see Figure 6.2b).



(a) Result obtained using *k-means* (b) *Voronoi* diagram

Figure 6.2 – k-means clustering example: (a) result with k = 3 clusters; (b) Voronoi diagram. The centroids are denoted by a larger font size.

Given a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x} \in \mathbb{R}^d$ and $S = \{S_1, S_2, \ldots, S_k\}$, is a set of partitions. The objective of k-means is to minimize the intra-cluster square error, as shown in Equation (6.1). Note that μ_i is the centroid average of all samples assigned to the cluster S_i.

$$\arg\min_{S} \sum_{i=1} \sum_{\mathbf{X}_j \in S_i} ||\mathbf{X}_j - \mu_i||^2 \tag{6.1}$$

The problem is very hard to solve, and approximate solutions are sought for optimization efficiency. Lloyd's algorithm [162] is a commonly used approximation that finds a local minimum iteratively through three differentiated steps: initialization, assignment, and update.

1. **Initialization step:** This step selects the initial random positions (seeds) as centroids. Note that this step is critical and significantly affects the solution obtained. We run the algorithm multiple times with different initializations, and the best score is picked. Some commonly used initializations are:

   (a) Forgy's method: A randomly selected centroids that are well separated.

   (b) Randomly partition: K initial centroids are randomly selected, and points are assigned to different classes based on the distance. The average distance is

calculated. Afterward, we update the centroids and recompute the distance.

(c) K-means++: This algorithm updates the centroids based on two factors: the squared distance, and the probability proportion derived from the point which is close to existing cluster centroids. [163].

2. **Assignment step:** Each sample is assigned to the closest centroid that contributes to a minimum intra-cluster quadratic error.

3. **Update step:** The new centroids μ_i are recalculated according to the average of all samples within the cluster S_i as expressed in Equation 6.2,

$$\mu_i^{t+1} = \frac{1}{|S_i^t|} \sum_{x_j \in S_i^t} \mathbf{x}_j \tag{6.2}$$

6.1.2 Connectivity-based clustering

it is formally known as hierarchical clustering and is based on the premise that local objects are more closely related to one another than distant ones. Therefore, the similarity is measured by the distance between the two samples. This clustering model does not locate a single data point, but rather presents a hierarchy of data clusters that have been combined at specific distances. This hierarchy is depicted by dendrograms (see Figure 6.3b), in which the y-axis reflects the distance at which two clusters join and the x-axis is arranged so that clusters do not mix. In general, connectivity-based clustering algorithms are categorized according to the following strategies:

1. Agglomerative strategies: It produces a cluster tree; the top is a list of all samples, and these are then joined to form subclusters as one moves down the tree until all cases are merged into a single large cluster. Consequently, there is the same number of clusters as samples.

2. Divisive strategies: Are a top-down clustering approach. Initially, all the samples belong to the same unique cluster, and partitioning is performed recursively as one moves down the hierarchy.

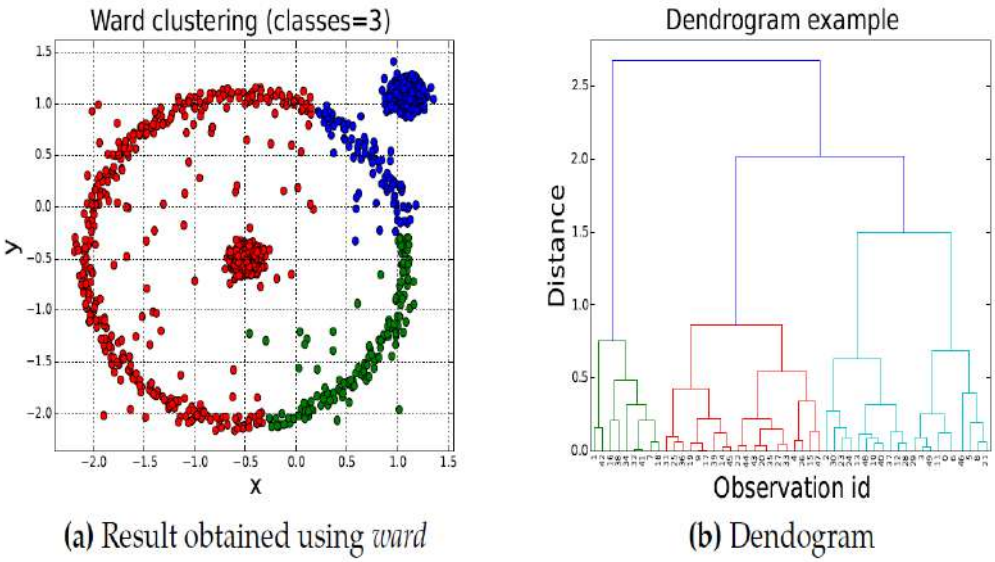


Figure 6.3 – Example of clustering using ward: (a) Result with a number of clusters k=3; (b) A dendogram with 50 samples.

Ward is a connectivity-based clustering algorithm, and its obtained result is illustrated in Figure 6.3a. Ward follows an agglomerative strategy, starting with one cluster for each sample at each step. The process makes a new cluster that minimizes total variance within the cluster. The initial distances between clusters are the Euclidean distances between samples; $d(x_i, x_j) = ||x_i - x_j||^2$. In each recursion, the pair of clusters that leads to a minimum total variance increment within clusters are merged. This process is repeated until the total number of clusters indicated by the user is reached.

6.1.3 Density based clustering

The key idea is to consider a cluster/group in a data space as a contiguous region of high point density, separated from other clusters by sparse or empty regions. The samples placed between two densely populated regions are considered noise. The popular density-based clustering algorithm is DBSCAN. This method connects samples at a given radius under the assumption that it must contain at least a minimum number of points and must also fulfill density criteria. DBSCAN has a clear weakness in being unable to group samples into clusters with very different densities. To solve this problem, Mihael Ankerst et al. [164] provided a generalization of the Ordering Points to Identify Cluster Structure method (OPTICS). First, the samples are ordered so that adjacent samples

are close together in the sorted vector. Additionally, the acceptable distance between two samples to retain them in the same cluster is stored. This length is known as the reachability distance. They can then be represented as a unique sort of dendrogram. The x-axis depicts the sorted samples, and the y-axis indicates their reachability distance. Typically, clusters have short reachability distances to their nearest neighbors; therefore, clusters are depicted as valleys in this depiction (see Figure 6.4b). The denser the cluster, the deeper the valley. Figure 6.4a displays the outcome obtained for the initial synthetic example.

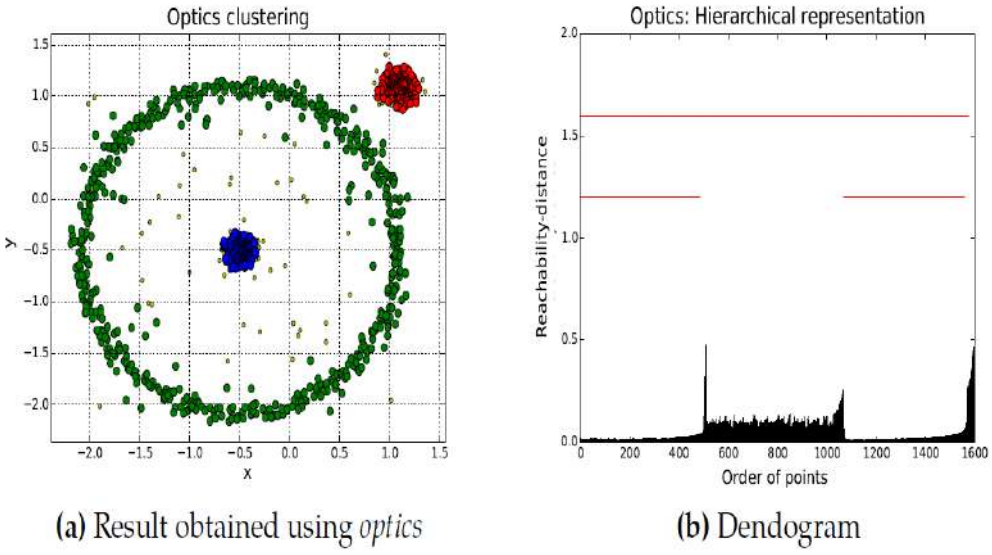


Figure 6.4 – Example of clustering utilising optics: (a) Clustering result without defining the number of clusters; (b) Samples ordered by their reachability distance, where valleys represent clusters. Notate the presence of yellowed outliers with lesser sizes.

Personal Contribution

Publication

— **Khamis, Houfar** & Djamel, Samai & Fadi, Dornaika & Azeddine, Benlamoudi & Khaled, Bensid & Abdelmalik, Taleb-Ahmed. (2023). **Automatically Weighted Binary Multi-View Clustering via Deep Initialization (AW-BMVC).** Pattern Recognition. https://doi.org/10.1016/j.patcog.2022.109281.

Bibliography

[1] W.Ian, F.Eibe *et al.*, "Practical machine learning tools and techniques," in *Data Mining*, no. 4th Edition, 2016.

[2] H.David, "Principles of data mining," *Drug safety*, vol. 30, pp. 621–622, 2007.

[3] J.K.Anil, M.Narasimha *et al.*, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[4] Y.Yan and W.Hao, "Multi-view clustering: A survey," *Big Data Mining and Analytics*, vol. 1, no. 2, pp. 83–107, 2018.

[5] B.Steffen and S.Tobias, "Multi-view clustering." in *ICDM*, vol. 4, no. 2004. Citeseer, 2004, pp. 19–26.

[6] N.W.Stafford and B.Asa, "Integrating information for protein function prediction," *Bioinformatics-From Genomes to Therapies*, pp. 1297–1314, 2007.

[7] H.Chenping, Z.Changshui *et al.*, "Multiple view semi-supervised dimensionality reduction," *Pattern Recognition*, vol. 43, no. 3, pp. 720–730, 2010.

[8] L.D.Daniel and S. Sebastian, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[9] Y.Zuyuan, Z.Yu *et al.*, "Non-negative matrix factorization with dual constraints for image clustering," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 7, pp. 2524–2533, 2018.

[10] W.Yu-Xiong and Z.Yu-Jin, "Nonnegative matrix factorization: A comprehensive review," *IEEE Transactions on knowledge and data engineering*, vol. 25, no. 6, pp. 1336–1353, 2012.

[11] L.Jialu, W.Chi *et al.*, "Multi-view clustering via joint nonnegative matrix factorization," in *Proceedings of the 2013 SIAM international conference on data mining*. SIAM, 2013, pp. 252–260.

[12] H.Xiangnan, K.Min-Yen *et al.*, "Comment-based multi-view clustering of web 2.0 items," in *Proceedings of the 23rd international conference on World wide web*, 2014, pp. 771–782.

[13] F.Lin, L.Wenzhe *et al.*, "Re-weighted multi-view clustering via triplex regularized non-negative matrix factorization," *Neurocomputing*, vol. 464, pp. 352–363, 2021.

[14] H.Shudong, K.Zhao *et al.*, "Auto-weighted multi-view clustering via deep matrix decomposition," *Pattern Recognition*, vol. 97, p. 107015, 2020.

[15] G.Yunchao, P.Marcin *et al.*, "Web scale photo hash clustering on a single machine," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2015, pp. 19–27.

[16] Y.Gong, S.Lazebnik *et al.*, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval, pattern analysis and machine intelligence," *IEEE Transactions on, PP (99)*, vol. 1, 2012.

[17] S.Xiaobo, L.Weiwei *et al.*, "Compressed k-means for large-scale clustering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[18] Z.Zheng, L.Li *et al.*, "Binary multi-view clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1774–1782, 2018.

[19] Z.Zheng and L.Li, "Highly-economized multi-view binary compression for scalable image clustering," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 717–732.

[20] L.Xin, L.Qiao *et al.*, "A multi-view model for visual tracking via correlation filters," *Knowledge-Based Systems*, vol. 113, pp. 88–99, 2016.

[21] X.Chang, T.Dacheng *et al.*, "A survey on multi-view learning," *arXiv preprint arXiv:1304.5634*, 2013.

[22] S.Shiliang, "A survey of multi-view machine learning," *Neural computing and applications*, vol. 23, pp. 2031–2038, 2013.

[23] S.Angela, G.Paola *et al.*, "Multiview learning in biomedical applications," in *Artificial Intelligence in the Age of Neural Networks and Brain Computing.* Elsevier, 2019, pp. 265–280.

[24] Y.Qiyue, W.Shu *et al.*, "Multi-view clustering via pairwise sparse subspace representation," *Neurocomputing*, vol. 156, pp. 12–21, 2015.

[25] W.Yang, Z.Wenjie *et al.*, "Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering," *arXiv preprint arXiv:1608.05560*, 2016.

[26] O.Mete, V.Fatos *et al.*, "Fusion of image segmentation algorithms using consensus clustering," in *2013 IEEE International Conference on Image Processing.* IEEE, 2013, pp. 4049–4053.

[27] A.Zeynep, T.Christian *et al.*, "Non-negative matrix factorization in multimodality data for segmentation and label prediction," in *16th Computer vision winter workshop*, 2011.

[28] D.Abdelaziz, F.Jean-Sebastien *et al.*, "Multi-view object segmentation in space and time," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2640–2647.

[29] D.Navneet and T.Bill, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, pp. 886–893.

[30] W.Jianxin and R.M.Jim, "Centrist: A visual descriptor for scene categorization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1489–1501, 2010.

[31] Y.Hui, L.Mingjing *et al.*, "Color texture moments for content-based image retrieval," in *Proceedings. International Conference on Image Processing*, vol. 3. IEEE, 2002, pp. 929–932.

[32] L.G.David, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[33] O.Timo, P.Matti *et al.*, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[34] K.Young-Min, A.Massih-Reza *et al.*, "Multi-view clustering of multilingual documents," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 2010, pp. 821–822.

[35] Z.Peng, J.Yuan *et al.*, "Multi-view matrix completion for clustering with side information," in *Advances in Knowledge Discovery and Data Mining: 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea, May 23-26, 2017, Proceedings, Part II 21*. Springer, 2017, pp. 403–415.

[36] H. Fawad, M.Muhammad *et al.*, "Multi-view document clustering via ensemble method," *Journal of Intelligent Information Systems*, vol. 43, no. 1, pp. 81–99, 2014.

[37] B. M, N.Y.Andrew *et al.*, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[38] M.Tomas, C.Kai *et al.*, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[39] F.Maha, B.H.Mohamed.Aymen *et al.*, "On the use of ensemble method for multi view textual data," *Journal of Information and Telecommunication*, vol. 4, no. 4, pp. 461–481, 2020.

[40] P.Vinay, F.Tito *et al.*, "Precision oncology: origins, optimism, and potential," *The Lancet Oncology*, vol. 17, no. 2, pp. e81–e86, 2016.

[41] C.J.Mitchell, G.D.Adam *et al.*, "Identification of complex metabolic states in critically injured patients using bioinformatic cluster analysis," *Critical Care*, vol. 14, no. 1, pp. 1–11, 2010.

[42] W.Xiang, S.David *et al.*, "Unsupervised learning of disease progression models," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 85–94.

[43] L.Wenyuan, L.Chun-Chi *et al.*, "Integrative analysis of many weighted co-expression networks using tensor computation," *PLoS computational biology*, vol. 7, no. 6, p. e1001106, 2011.

[44] S.K.Nora and P.Nico, "Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery," *Bioinformatics*, vol. 31, no. 12, pp. i268–i275, 2015.

[45] D.Daisy, L.Shuangning *et al.*, "Cooperative learning for multiview analysis," *Proceedings of the National Academy of Sciences*, vol. 119, no. 38, p. e2202113119, 2022.

[46] F.Yixiang, Z.Haijun *et al.*, "Detecting hot topics from twitter: A multiview approach," *Journal of Information Science*, vol. 40, no. 5, pp. 578–593, 2014.

[47] D.Shang, D.Xin-Yu *et al.*, "A multi-view clustering model for event detection in twitter," in *Computational Linguistics and Intelligent Text Processing: 18th International Conference, CICLing 2017, Budapest, Hungary, April 17–23, 2017, Revised Selected Papers, Part II 18*. Springer, 2018, pp. 366–378.

[48] C.Chengyao, G.Dehong *et al.*, "A constrained multi-view clustering approach to influence role detection," in *Social Media Content Analysis: Natural Language Processing and Beyond*. World Scientific, 2018, pp. 237–252.

[49] W.Xiao, N.Chong-Wah *et al.*, "Multimodal news story clustering with pairwise visual near-duplicate constraint," *IEEE Transactions on Multimedia*, vol. 10, no. 2, pp. 188–199, 2008.

[50] R.Bekkerman and J.Jiwoon, "Multi-modal clustering for multimedia collections," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[51] V.Mekthanavanh, L.Tianrui *et al.*, "Social web video clustering based on multimodal and clustering ensemble," *Neurocomputing*, vol. 366, pp. 234–247, 2019.

[52] J.Sungkyu and M. Stephen, "Pca consistency in high dimension. low sample size context," *Ann. Statist*, 2009.

[53] X.Chang, T.Dacheng *et al.*, "Multi-view learning with incomplete views," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5812–5825, 2015.

[54] C.Ying, F.Z.Xiaoli *et al.*, "Non-redundant multi-view clustering via orthogonalization," in *Seventh IEEE international conference on data mining (ICDM 2007)*. IEEE, 2007, pp. 133–142.

[55] N.Donglin, D.G.Jennifer, M. Jordan *et al.*, "Multiple non-redundant spectral clustering views," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 831–838.

[56] C.Yale, C.Junxiang *et al.*, "Multiple clustering views from multiple uncertain experts," in *International Conference on Machine Learning*. PMLR, 2017, pp. 674–683.

[57] K.S.Yuan, *Kernel methods and machine learning*. Cambridge University Press, 2014.

[58] F.P.Deena and R.Kumudha, "Major advancements in kernel function approximation," *Artificial Intelligence Review*, vol. 54, pp. 843–876, 2021.

[59] G.A.Tsihrintzis, V.Maria *et al.*, *Machine Learning Paradigms: Applications of Learning and Analytics in Intelligent Systems*. Springer, 2019, vol. 1.

[60] B.Vidakovic, *Engineering biostatistics: an introduction using MATLAB and WinBUGS*. John Wiley & Sons, 2017.

[61] K.Leonard, "Clustering by means of medoids," in *Proc. Statistical Data Analysis Based on the L1 Norm Conference, Neuchatel, 1987*, 1987, pp. 405–416.

[62] B.Aruna, "K-medoids clustering using partitioning around medoids for performing face recognition," *International Journal of Soft Computing, Mathematics and Control*, vol. 3, no. 3, pp. pp. 1–12, 2014.

[63] E.Elhamifar, S.Guillermo *et al.*, "See all by looking at a few: Sparse modeling for finding representative objects," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 1600–1607.

[64] B.Stephen, P.Neal *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[65] C.Dong, C.Xudong *et al.*, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3025–3032.

[66] N.Apostol, H.Alexander *et al.*, "Semantic concept-based query expansion and re-ranking for multimedia retrieval," in *Proceedings of the 15th ACM international conference on Multimedia*, 2007, pp. 991–1000.

[67] L.Zhongyu, Z.Xiaofan *et al.*, "Large-scale retrieval for medical image analytics: A comprehensive review," *Medical image analysis*, vol. 43, pp. 66–84, 2018.

[68] L.Wei, W.Jun *et al.*, "Supervised hashing with kernels," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 2074–2081.

[69] S.Malcolm and C.Michael, "Locality-sensitive hashing for finding nearest neighbors [lecture notes]," *IEEE Signal processing magazine*, vol. 25, no. 2, pp. 128–131, 2008.

[70] Z.Han, L.Mingsheng *et al.*, "Deep hashing network for efficient similarity retrieval," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[71] L.Wei, W.Jun *et al.*, "Hashing with graphs," *ICML'11*, 2011.

[72] A.Alexandr and I.Piotr, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Communications of the ACM*, vol. 51, no. 1, pp. 117–122, 2008.

[73] R.Maxim and L.Svetlana, "Locality-sensitive binary codes from shift-invariant kernels," *Advances in neural information processing systems*, vol. 22, 2009.

[74] T.Antonio, R.Fergus *et al.*, "Small codes and large image databases for recognition," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[75] K.Brian and D.Trevor, "Learning to hash with binary reconstructive embeddings," *Advances in neural information processing systems*, vol. 22, 2009.

[76] X.Rongkai, P.Yan *et al.*, "Supervised hashing for image retrieval via image representation learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 28, no. 1, 2014.

[77] C.Zhixiang, Y.Xin *et al.*, "Deep hashing via discrepancy minimization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6838–6847.

[78] S.Jingkuan, Y.Yang *et al.*, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 423–432.

[79] L.Xianglong, H.Junfeng *et al.*, "Multiple feature kernel hashing for large-scale visual search," *Pattern Recognition*, vol. 47, no. 2, pp. 748–757, 2014.

[80] L.Xianglong, L.Huang *et al.*, "Multi-view complementary hash tables for nearest neighbor search," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1107–1115.

[81] S.Jingkuan, Y.Yang *et al.*, "Effective multiple feature hashing for large-scale near-duplicate video retrieval," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1997–2008, 2013.

[82] S.Xiaobo, S.Fumin *et al.*, "Multi-view latent hashing for efficient multimedia search," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 831–834.

[83] Z.Dan, W.Fei *et al.*, "Composite hashing with multiple information sources," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011, pp. 225–234.

[84] K.Saehoon and C.Seungjin, "Multi-view anchor graph hashing," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 3123–3127.

[85] Y.Yang, X.Dong *et al.*, "Image clustering using local discriminant models and global integration," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2761–2773, 2010.

[86] K.W.Harold, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[87] A.Rosenberg and J.Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007, pp. 410–420.

[88] M.F.Aaron, G.Derek *et al.*, "Normalized mutual information to evaluate overlapping community finding algorithms," *arXiv preprint arXiv:1110.2515*, 2011.

[89] Z.Ying and K.George, "Criterion functions for document clustering: Experiments and analysis," *University of Minnesota*, 2001.

[90] C.Guoqing, S.Shiliang *et al.*, "A survey on multiview clustering," *IEEE transactions on artificial intelligence*, vol. 2, no. 2, pp. 146–168, 2021.

[91] N.Andrew, J.Michael *et al.*, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 14, 2001.

[92] S.Jianbo and M.Jitendra, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[93] W.Shiping and G.Wenzhong, "Sparse multigraph embedding for multimodal feature representation," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1454–1466, 2017.

[94] G.Quanxue, W.Zhizhen *et al.*, "Multi-view projected clustering with graph learning," *Neural Networks*, vol. 126, pp. 335–346, 2020.

[95] K.Abhishek and D.Hal, "A co-training approach for multi-view spectral clustering," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 393–400.

[96] K.Abhishek, R.Piyush *et al.*, "Co-regularized multi-view spectral clustering," *Advances in neural information processing systems*, vol. 24, 2011.

[97] X.Tian, T.Dacheng *et al.*, "Multiview spectral embedding," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 6, pp. 1438–1446, 2010.

[98] N.Feiping, Li.Jing *et al.*, "Self-weighted multiview clustering with multiple graphs." in *IJCAI*, 2017, pp. 2564–2570.

[99] N.Feiping, L.Jing *et al.*, "Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification." in *IJCAI*, 2016, pp. 1881–1887.

[100] S.Shaojun, N.Feiping *et al.*, "Auto-weighted multi-view clustering via spectral embedding," *Neurocomputing*, vol. 399, pp. 369–379, 2020.

[101] N.Feiping, T.Lai *et al.*, "Multiview clustering via adaptively weighted procrustes," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 2022–2030.

[102] W.Hao, Y.Yan *et al.*, "Gmc: Graph-based multi-view clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1116–1129, 2019.

[103] Z.Linlin, Z.Xianchao *et al.*, "Weighted multi-view spectral clustering based on spectral perturbation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[104] H.Shudong, K.Zhao *et al.*, "Auto-weighted multi-view clustering via kernelized graph learning," *Pattern Recognition*, vol. 88, pp. 174–184, 2019.

[105] S. Hajjar, F.Dornaika *et al.*, "One-step multi-view spectral clustering with cluster label correlation graph," *Information Sciences*, vol. 592, pp. 97–111, 2022.

[106] V.Rene, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

[107] Z.Changqing, H.Qinghua *et al.*, "Latent multi-view subspace clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4279–4287.

[108] W.Xiaobo, G.Xiaojie *et al.*, "Exclusivity-consistency regularized multi-view subspace clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 923–931.

[109] L.Shirui, Z.Changqing *et al.*, "Consistent and specific multi-view subspace clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[110] B.Maria and K.Ivica, "Multi-view low-rank sparse subspace clustering," *Pattern Recognition*, vol. 73, pp. 247–258, 2018.

[111] W.Yang, L.Xuemin, *et al.*, "Robust subspace clustering for multi-view data by exploiting correlation consensus," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3939–3949, 2015.

[112] Z.Changqing, F.Huazhu *et al.*, "Generalized latent multi-view subspace clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 1, pp. 86–99, 2018.

[113] L.Shao-Yuan, J.Yuan *et al.*, "Partial multi-view clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 28, no. 1, 2014.

[114] Z.Handong, L.Hongfu *et al.*, "Incomplete multi-modal visual data grouping." in *IJCAI*, 2016, pp. 2392–2398.

[115] Y.Qiyue, W.Shu *et al.*, "Incomplete multi-view clustering via subspace learning," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 383–392.

[116] C.Xiaochun, Z.Changqing *et al.*, "Diversity-induced multi-view subspace clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 586–594.

[117] T.Hong, H.Chenping *et al.*, "Multiview classification with cohesion and diversity," *IEEE transactions on cybernetics*, vol. 50, no. 5, pp. 2124–2137, 2018.

[118] G.Hongchang, N.Feiping *et al.*, "Multi-view subspace clustering," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4238–4246.

[119] C.Yongyong, X.Xiaolin *et al.*, "Adaptive transition probability matrix learning for multiview spectral clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4712–4726, 2021.

[120] C.Xiao, N.Feiping *et al.*, "Multi-view k-means clustering on big data," in *Twenty-Third International Joint conference on artificial intelligence*, 2013.

[121] X.Jinglin, H.Junwei *et al.*, "Re-weighted discriminatively embedded k-means for multi-view clustering," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 3016–3027, 2017.

[122] L.Hongfu and F.Yun, "Consensus guided multi-view clustering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 12, no. 4, pp. 1–21, 2018.

[123] C.Deng, H.Xiaofei *et al.*, "Non-negative matrix factorization on manifold," in *2008 eighth IEEE international conference on data mining*. IEEE, 2008, pp. 63–72.

[124] C.Deng and H.Xiaofei, "Graph regularized nonnegative matrix factorization for data representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1548–1560, 2010.

[125] Z.Linlin, Z.Xianchao *et al.*, "Multi-view clustering via multi-manifold regularized non-negative matrix factorization," *Neural Networks*, vol. 88, pp. 74–89, 2017.

[126] W.Hao, Y.Yan *et al.*, "Multi-view clustering via concept factorization with local manifold regularization," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 1245–1250.

[127] P.Jiameng, Z.Qian *et al.*, "Multiview clustering based on robust and regularized matrix approximation," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 2550–2555.

[128] C.Guoqing and S.Shiliang, "Consensus and complementarity based maximum entropy discrimination for multi-view classification," *Information Sciences*, vol. 367, pp. 296–310, 2016.

[129] C.Kamalika, K.M.Sham *et al.*, "Multi-view clustering via canonical correlation analysis," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 129–136.

[130] B. B and L.H.Christoph, "Correlational spectral clustering," in *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 2008, pp. 1–8.

[131] L.Xinhai, J.Shuiwang *et al.*, "Multiview partitioning via tensor methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 5, pp. 1056–1069, 2012.

[132] Z.Yuanpeng, C.Fu-Lai *et al.*, "A multiview and multiexemplar fuzzy clustering approach: theoretical analysis and experimental studies," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 8, pp. 1543–1557, 2018.

[133] L.Yingming, Y.Ming *et al.*, "A survey of multi-view representation learning," *IEEE transactions on knowledge and data engineering*, vol. 31, no. 10, pp. 1863–1883, 2018.

[134] Z.Pengfei, H.Binyuan *et al.*, "Multi-view deep subspace clustering networks," *arXiv preprint arXiv:1908.01978*, 2019.

[135] L.Zhaoyang, W.Qianqian *et al.*, "Deep adversarial multi-view clustering network." in *IJCAI*, 2019, pp. 2952–2958.

[136] Z.Runwu and S.Yi-Dong, "End-to-end adversarial-attention network for multimodal clustering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 619–14 628.

[137] S. Farial, B.Michael, *et al.*, "Document clustering using nonnegative matrix factorization," *Information Processing & Management*, vol. 42, no. 2, pp. 373–386, 2006.

[138] H. Teny, H.Lely *et al.*, "Intelligent kernel k-means for clustering gene expression," *Procedia Computer Science*, vol. 59, pp. 171–177, 2015.

[139] S.Gang, Y.Dongmei *et al.*, "A distance-based spectral clustering approach with applications to network community detection," *Journal of Industrial Information Integration*, vol. 6, pp. 22–32, 2017.

[140] P.Van, N.Pham *et al.*, "Multi-view clustering and multi-view models," in *Recent Advancements in Multi-View Data Analytics*. Springer, 2022, pp. pp. 55–96.

[141] L.Yeqing, N.Feiping *et al.*, "Large-scale multi-view spectral clustering via bipartite graph," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.

[142] S. Hajjar, F. Dornaika *et al.*, "Consensus graph and spectral representation for one-step multi-view kernel based clustering," *Knowledge-Based Systems*, vol. 241, p. pp. 108250, 2022.

[143] J.Wang, Z.Ting *et al.*, "A survey on learning to hash," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. pp. 769–790, 2017.

[144] S.Fumin, Z.Xiang *et al.*, "A fast optimization method for general binary code learning," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5610–5621, 2016.

[145] S.Weglarczyk, "Kernel density estimation and its application," *ITM Web Conf.*, vol. 23, p. pp. 00037, 2018.

[146] J.Wang, S.Kumar *et al.*, "Semi-supervised hashing for scalable image retrieval," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3424–3431.

[147] B.Wang, Y.Xiao *et al.*, "Robust self-weighted multi-view projection clustering," *2020*, vol. 34, pp. pp. 6110–6117, Apr. AAAI Press.

[148] A.Laith, Z.Jinglan *et al.*, "Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, no. 1, pp. pp. 1–74, 2021.

[149] A.Jamil, M.Khan *et al.*, "Efficient conversion of deep features to compact binary codes using fourier decomposition for multimedia big data," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. pp. 3205–3215, 2018.

[150] A.Valdez, P.Megan *et al.*, "Distributed representation of visual objects by single neurons in the human brain," *Journal of Neuroscience*, vol. 35, no. 13, pp. pp. 5180–5186, 2015.

[151] A.Jamil, M.Khan *et al.*, "Medical image retrieval with compact binary codes generated in frequency domain using highly reactive convolutional features," *Journal of medical systems*, vol. 42, no. 2, pp. pp. 1–19, 2018.

[152] L.Fei-Fei, R.Fergus *et al.*, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, 2004, pp. 178–178.

[153] C.Tat-Seng, T.Jinhui *et al.*, "Nus-wide: a real-world web image database from national university of singapore," *Association for Computing Machinery*, pp. 1–9, 2009.

[154] S.Lazebnik, S.Cordelia *et al.*, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," *IEEE*, vol. 2, pp. 2169–2178, 2006.

[155] N.Liang, Z.Yang *et al.*, "Multi-view clustering by non-negative matrix factorization with co-orthogonal constraints," *Knowledge-Based Systems*, vol. 194, p. pp. 105582, 2020.

[156] R.Xia, Y.Pan *et al.*, "Robust multi-view spectral clustering via low-rank and sparse decomposition," *AAAI*, vol. 28, 2014.

[157] Z.Xiaofeng, Z.Shichao *et al.*, "One-step multi-view spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 10, pp. pp. 2022–2034, 2018.

[158] K.Zhao, Z.Wangtao *et al.*, "Large-scale multi-view subspace clustering in linear time," *AAAI*, vol. 34, 2020.

[159] H.Zhanxuan, N.Feiping *et al.*, "Multi-view spectral clustering via integrating nonnegative embedding and spectral embedding," *Information Fusion*, vol. 55, pp. pp. 251–259, 2020.

[160] C.Peng, L.Liang *et al.*, "Smoothed multi-view subspace clustering," in *Neural Computing for Advanced Applications*. Singapore: Springer Singapore, 2021, pp. 128–140.

[161] Y.Miin-Shen and S.Kristina, "Collaborative feature-weighted multi-view fuzzy c-means clustering," *Pattern Recognition*, vol. 119, p. pp. 108064, 2021.

[162] L.Stuart, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[163] A.David and S.Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.

[164] A.Mihael, B.M.Markus *et al.*, "Optics: Ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.