Discovering visual patterns for image data analysis

Wang, Hongxing

2015

Wang, H. (2015). Discovering visual patterns for image data analysis. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/62511>

<https://doi.org/10.32657/10356/62511>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**DISCOVERING VISUAL PATTERNS FOR
IMAGE DATA ANALYSIS**

**WANG HONGXING
SCHOOL OF ELECTRICAL & ELECTRONIC
ENGINEERING
2015**

DISCOVERING VISUAL PATTERNS FOR IMAGE DATA ANALYSIS

WANG HONGXING

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2015

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor, Professor Jun-song Yuan, for his insightful guidance, enthusiastic support, and tireless encouragement in my research. It is Prof. Yuan, who directs me into such a beautiful and fantastic research realm of computer vision and data mining.

My sincere thanks also goes to my co-supervisor, Professor Yap-Peng Tan, for his constructive feedback and warm encouragement on my research progress.

I wish to thank Professor Yoshinobu Kawahara, for offering me a great deal of assistance and guidance during my internship.

Thanks are in order to my fellow mates in Media Technology Lab and Rapid-Rich Object Search Lab, for the stimulating discussions and all the fun we have had. I learnt from them everyday.

Last but not the least, I would like to thank my dear family and all my friends, for their never-wavering support and concern.

Table of Contents

Acknowledgements	i
Table of Contents	iii
Summary	v
List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 Visual Pattern Discovery in Image Spatial Domain	3
1.2 Multi-feature Fusion for Visual Pattern Discovery	5
1.3 Main Contributions	6
1.4 Structure of the Thesis	7
2 Literature Review	9
2.1 Spatial Visual Pattern Discovery	9
2.2 Multi-feature Fusion	13
3 Context-Aware Discovery of Visual Co-occurrence Patterns	17
3.1 Introduction	17
3.2 Multi-Context-Aware Clustering	19
3.2.1 Problem Statement	19
3.2.2 Algorithm	25
3.3 Experiments	27
3.3.1 Spatial Visual Pattern Discovery	29
3.3.2 Image Region Clustering using Multiple Contexts	30
3.3.3 Bag-of-Words for Image Classification	38
3.3.4 Parameter Comparison and Selection	38
3.4 Conclusion	41
4 Multi-feature Spectral Clustering with Minimax Optimization	43
4.1 Introduction	43
4.2 Proposed Method	45
4.2.1 Laplacian Embedding	45

4.2.2	Regularized Data-Cluster Similarity Matrix	46
4.2.3	Towards Agreement among Regularized Data-Cluster Similarity Matrices	46
4.2.4	Optimization	48
4.3	Experiments	51
4.3.1	Datasets and Experimental Setting	51
4.3.2	Baseline Algorithms	55
4.3.3	Experimental Results	57
4.3.4	Convergence Analysis	62
4.3.5	Sensitivity of Parameters	62
4.4	Conclusion	63
5	Collaborative Multi-feature Fusion for Transductive Spectral Learning	65
5.1	Introduction	65
5.2	Proposed Method	68
5.2.1	Spectral Embedding of Multi-feature Data	69
5.2.2	Building Feature Co-occurrence Patterns and Multi-feature Similarity Graph	70
5.2.3	Multi-feature Fusion with Transductive Learning	72
5.2.4	Optimization: Collaboration between Clustering and Classification	73
5.3	Experiments	77
5.3.1	Experiment Setting	77
5.3.2	Synthetic Data	79
5.3.3	UCI Handwritten Digit Dataset	80
5.3.4	Oxford Flower Dataset	84
5.3.5	Human Body Motion Dataset	85
5.3.6	UC Merced Land Use Dataset	87
5.4	Conclusion	89
6	Conclusion and Future Work	91
	Appendix	95
A.	Matrix Form of Hamming Distance	95
B.	Derivation of Eqs. 3.7, 3.8, 3.9	96
C.	Deviation of Eq. 4.8	97
D.	Derivation of Eq. 5.16	98
	Author's Publications	101
	Bibliography	103

Summary

Discovering meaningful patterns in image data can help to better understand and search the visual contents that convey rich information. Unlike conventional pattern mining in transaction or text data, visual patterns usually contain complex spatial structures and exhibit large variations, thus making the image data mining task very challenging. This thesis is devoted to developing effective methods for visual pattern discovery, and has the following three contributions.

The first contribution of this thesis is on spatial visual pattern discovery. We focus on the composition of visual patterns from visual primitives (*e.g.*, local interest points or regions in images). Conventional clustering of visual primitives, *e.g.*, bag-of-words, usually ignores the spatial and feature structure among the visual primitives, thus cannot discover high-level visual patterns of complex structure. To overcome this problem, we propose to consider both spatial and feature contexts among visual primitives for spatial pattern discovery. We formulate the pattern discovery problem as a regularized k -means clustering, and propose a self-learning procedure to gradually refine the result until it converges. Our experiments validate that by discovering both spatial co-occurrence patterns among visual primitives and feature co-occurrence patterns among different types of features, our method can effectively reduce the ambiguities of visual primitives.

The second contribution of this thesis is on multi-feature clustering for visual pattern discovery. Given multiple-feature representation for each image, we propose a novel minimax formulation to reach a consensus clustering, without requiring to specify the weighting parameter to fuse the multiple feature modalities. Our objective of consensus clustering is to find a universal feature embedding, which not only fits each feature modality well, but also unifies different modalities by minimizing the pairwise disagreement between any two of them. The experiments with real image data show the advantages of the proposed multi-feature clustering method when compared with existing methods.

The last contribution of this thesis is on semi-supervised visual pattern discovery. We study visual pattern classification using collaborative multi-feature fusion in a transductive learning framework, where the labelled data can transfer the labels to the unlabelled data. To enable transductive spectral learning, we formulate a new objective function with three objectives, namely the good quality of spectral clustering in individual feature types, the label smoothness of data samples in terms of their feature co-occurrence representations, and the fitness to the labels provided by the training data. However, the spectral clustering results in different feature types and the formed co-occurrence patterns influence each other under the transductive learning formulation, which makes the optimization of this objective function challenging. To address this problem, we propose an iterative optimization approach that can decouple these factors. During the iterations, the clustering results of individual feature types and the smoothness of the labelling of data samples will help each other, leading to a better transductive learning and a pattern classification result. The effectiveness of the proposed method is validated by both synthetic and real image data.

List of Figures

1.1	Diverse visual patterns: (a) the repetitive “double-G” textures generate the texton patterns in a Gucci bag; (b) two eyes, a nose, and a mouth sketch a face pattern. Source: Images are from Caltech 101 dataset [1]; (c) a bed, a lamp <i>etc.</i> usually make up a bedroom. Source: Images are from MIT Indoor dataset [2]; (d) upturning of the torso and bending of the free leg together show the bent-leg layover spin action [3].	1
1.2	Problems studying in this thesis.	2
2.1	Bottom-up (a) and top-down (b) visual pattern discovery.	9
3.1	A spatial pattern $\{left\ eye, right\ eye, nose, mouth\}$ co-occur frequently which forms multiple faces of different people.	19
3.2	Multi-context-aware clustering via a self-learning procedure between low-level feature clustering and high-level pattern discovery.	21
3.3	Pattern discovery from a mono-colored LV monogram picture. Best viewed in color.	28
3.4	Pattern discovery from a colored group photo. Best viewed in color. . .	30
3.5	Two scene categories: “sheep+grass” and “bicycle+road”.	31
3.6	Class disambiguation by using spatial contexts.	32
3.7	Confusion matrices of k -means clustering and multi-context-aware clustering for image regions from image collection shown in Figure 3.5. . .	33
3.8	Sample images of five region compositions: “sheep+grass”, “cow+grass”, “aeroplane+grass+sky”, “boat+water”, and “bicycle+road”.	34
3.9	Confusion matrices of k -means clustering and multi-context-aware clustering for image regions from image collection shown in Figure 3.8. . .	36
3.10	Exemplar clustering results of multi-context-aware clustering.	37
3.11	Sample images of different sports events in UIUC sports dataset.	39
3.12	Clustering error rate of multi-context-aware clustering for image regions from image collection shown in Figure 3.5 w.s.t. parameters: τ_s and τ_v	39
4.1	Visualization of some sampled images of UCI Handwritten Digits. Each row shows 30 images from the same class of digits.	53
4.2	Sample images from Oxford 17-Category Flower Dataset. Five images are shown for each category. Each category contains instances of pose variations, scale changes, illumination variations, large intra-class variations and self-occlusion.	53

4.3	Sample images from UC Merced 21-Category Land Use Dataset. Five samples from each category are shown above.	54
4.4	Sample videos from Human Body Motion Dataset. One sample from each category is shown above.	55
4.5	Iteration comparisons of PRSC, CRSC and our method w.r.t. NMI performance of each feature embedding and the fusion result on the body motion and Oxford flower datasets. Best viewed in color.	58
4.6	Convergence study of our algorithm: NMI and objective values with iterations.	60
4.7	Parameter study of our algorithm: NMI values with different parameters.	61
5.1	Label propagation of unlabeled data by the discovery of the co-occurrence patterns among different types of clusters. See text for details and best viewed in color.	67
5.2	Classification on synthetic toy data with two feature types. Different markers, <i>i.e.</i> , “1”, “2”, “3” and “4”, indicate four different classes. Shading markers highlight the labeled data. The first column shows the synthetic toy data. The last three columns show the classifying results of RWMV [4], GTAM [5] and our proposed approach. Best viewed in color.	78
5.3	Performance comparison on UCI handwritten digits.	81
5.4	Performance comparison on Oxford 17-category flowers.	82
5.5	Confusion matrix comparison on Oxford 17-category flowers.	83

List of Tables

3.1	Clustering performance of image regions from image collection shown in Figure 3.5.	31
3.2	Clustering performance of image regions from image collection shown in Figure 3.8.	34
3.3	Classification Accuracy on UIUC sports dataset.	38
4.1	The image/video datasets used in our experiment and their feature descriptors.	52
4.2	Comparisons of various baselines with the proposed approach.	56
4.3	Performance of flower clustering with different feature input orders using our algorithm.	63
5.1	Performance of our approach on UCI handwritten digits under different cluster numbers per feature type. The size of labeled data is 20.	81
5.2	Performance comparison on Human Body Motion videos.	86
5.3	Performance comparison on UC Merced Land Use images.	88

Chapter 1

Introduction

Similar to frequent patterns in transaction data, visual patterns are compositions of visual primitives that appear frequently in image and video data [6, 7]. The visual primitives that construct visual patterns can be very diverse, *e.g.*, local image patches, semantic visual parts or visual objects. As we show in Figure 1.1, the visual pattern in image or video data can be a texton that captures the repetitiveness of image texture [8], *e.g.*, the “double-G” pattern in a Gucci bag; an abstract object model that describes its composition of visual parts [9], *e.g.*, a face pattern composing of two eyes, a nose, and a mouth; a scene layout pattern that captures the key objects which compose the scene [10], *e.g.*, a bedroom composing of a bed, a lamp *etc.*; or a human action that describes postures

FIGURE 1.1. Diverse visual patterns: (a) the repetitive “double-G” textures generate the texton patterns in a Gucci bag; (b) two eyes, a nose, and a mouth sketch a face pattern. Source: Images are from Caltech 101 dataset [1]; (c) a bed, a lamp *etc.* usually make up a bedroom. Source: Images are from MIT Indoor dataset [2]; (d) upturning of the torso and bending of the free leg together show the bent-leg layover spin action [3].

FIGURE 1.2. Problems studying in this thesis.

and motions of human body, *e.g.*, a bent-leg layover spin action showing by upturning the torso and bending the free leg.

Ubiquitous visual patterns show protean images. Just like the perception of repeated structures is well-nigh fundamental to the understanding the world around us [11], the recognition of visual patterns is essential to the understanding of image data. In practice, visual patterns can be used to model images, which have extensive applications in image analysis, such as image search, object categorization and human action recognition. It therefore offers an interesting, practical, but challenging issue for us to mine visual patterns from images.

Although frequent pattern mining has been well studied in data mining community [12], the existing frequent pattern mining methods cannot be applied to image data directly. This is because the complex spatial structures and heterogeneous feature descriptions among visual data make the problem of visual pattern discovery more challenging. First of all, to study the spatial structures among visual data, it is usually required to extract stable visual primitives from image data. To obtain visual primitives, many local feature detectors have been proposed [13, 14], *e.g.*, Difference of Gaussian (DoG) [15] and Hessian-Laplacian [16]. In addition, segmentation methods, *e.g.*, Normalized cuts [17], can be used to collect primitive regions; and object detection methods, *e.g.*, deformable part models [18] and object proposal [19], can provide object primitives appearing in image data. Once we have visual primitives, we can encode their appearance using feature descriptors [20]. For example, Scale-Invariant Feature Transform (SIFT) [15] and

Histograms of Oriented Gradients (HOG) [21] are the widely used gradient features. The efficient binary features [22] are such as Binary Robust Independent Elementary Features (BRIEF) [23] and Oriented FAST and Rotated BRIEF (ORB) [24]. To guarantee performance, Fisher Vector [25–27], Vector of Locally Aggregated Descriptors (VLAD) [28], and Convolutional Neural Networks (CNN) based features [29–31] are suggested to use. Instead of describing them using raw features, we can also use clustering method, *e.g.*, k -means, to further quantize feature descriptors into discrete visual words. After that, each visual primitive can be identified by the corresponding visual word. Meanwhile, each image can be represented as a bag of visual words using a histogram vector. With these visual words, it will be interesting in leveraging spatial dependencies among visual primitives for compositional pattern discovery in image spatial domain. Apart from that, image data naturally exhibit multiple feature modalities, *e.g.*, color and texture, which are complementary. It is also worth considering multi-feature fusion rather than using a single feature type for a better visual pattern discovery. Figure 1.2 shows the problems that will be studied in this thesis. We will study visual pattern discovery in two aspects: (1) decomposing each image explicitly into visual primitives, followed by spatial co-occurrence pattern mining on visual primitives; and (2) considering each image as a composition of latent visual primitives, followed by multi-feature clustering/classification on original images.

1.1 Visual Pattern Discovery in Image Spatial Domain

In the past decade, there have been increasing efforts to address visual pattern discovery in the literature [32]. Although it has been a common practice to build a visual vocabulary for image analysis, it remains a challenging problem to cluster visual primitives into meaningful patterns. Most existing visual primitive clustering methods ignore the spatial structure among the visual primitives [6], thus bringing unsatisfactory results. For example, a simple k -means clustering of visual primitives can lead to synonymous

visual words that over-represent visual primitives, as well as polysemous visual words that bring large uncertainties and ambiguities in the representation [33, 34].

In image and video data, visual primitives are not independent to each other. Therefore, visual polysemous and synonymous phenomena can be partially resolved through analyzing their spatial contexts [35, 36], *i.e.*, other primitives in the spatial neighborhood. Two visual primitives, although exhibit dissimilar visual features, may belong to the same pattern if they have the same spatial contexts. What is more, even though two visual primitives share similar features, they may not belong to the same visual pattern if their spatial contexts are completely different. By considering the spatial dependency among visual primitives, much previous work has tried to discover the spatial co-occurrence patterns of visual primitives [37–40]. Such co-occurrence patterns, once discovered, can be utilized to better address the ambiguities and uncertainties among visual primitives and have proven to be helpful for object categorization and image search.

Despite previous success of utilizing spatial visual patterns, there lacks a principled solution that can leverage visual patterns to improve the clustering of visual primitives. For example, [37] applies data mining methods first to find the spatial visual patterns. After that it relies on a separate sub-space learning to refine the presentation and clustering of visual primitives. As the clustering of visual primitives and the discovery of visual pattern will influence each other, it is preferred to provide a uniformed solution that can integrate visual pattern discovery into the process of visual primitives clustering [41]. Moreover, image and video data naturally exhibit multiple feature modalities that are complementary. Most existing approaches discover visual patterns using a single feature modality. However, for a better visual pattern discovery, a suitable integration of multiple complementary features is also extremely preferred [32], which will also be studied in this thesis [41].

1.2 Multi-feature Fusion for Visual Pattern Discovery

As noted above, besides the spatial dependencies among visual primitives, a visual pattern can exhibit certain feature dependencies as well. For example, an image patch can be naturally represented using different types (modalities) of features, *e.g.*, color, shape, and texture features. Instead of using a single feature modality, a suitable integration of multiple complementary features can result in a better clustering or classification result. Such a multi-feature fusion, however, is challenging due to the heterogeneous co-occurrence of different features. Moreover, the co-occurrence is difficult to reveal by a simple feature concatenation [42, 43]. As a result, it is equally interesting in discovering feature co-occurrence patterns across different feature domains.

Since clustering can faithfully reflect the data characteristics in different feature types, *e.g.*, color feature space can be categorized into a number of typical colors, while texture feature space categorized into a different number of texture patterns. To integrate the clustering results from different feature types, a superb choice is to pursue a universal representation for each data by a co-occurrence of feature patterns, *e.g.*, a composition of typical color and texture patterns. Much previous work has studied how to leverage multiple feature types to improve the clustering or classification performance [44]. Based on this, this thesis will discuss our recent progress on multi-feature fusion using multiple feature clustered patterns [45, 46]. Unlike much previous work on co-occurrence pattern discovery [32] in spatial domain, *e.g.*, [35, 36] and [37], it mainly aims to capture compositional patterns across multiple feature modalities, thus can naturally combine multiple features. Comparing with a direct concatenation of multiple types of features, such a treatment can encode the latent compositional structure among multiple feature types, thus resulting in a better representative power.

1.3 Main Contributions

The major contributions of this thesis are summarized below.

Firstly, we propose a uniformed solution that can integrate visual pattern discovery into the process of visual primitives clustering. The contributions of this work are mainly in the following two aspects:

- We provide a regularized k -means clustering formulation of visual co-occurrence pattern discovery, and propose a novel self-supervised learning procedure to leverage the discovered visual co-occurrence patterns to guide the clustering of visual primitives. Such a self-supervised learning is proven to converge in finite steps.
- Our proposed uniformed formulation can leverage both spatial and feature co-occurrence patterns and let them boost each other to improve the pattern discovery results.

Secondly, we propose a novel minimax formulation to reach a consensus clustering, without requiring to specify the weighting parameter to fuse the multiple feature modalities. The novel minimax formulation has the following advantages:

- It has only one global parameter, while all fusing weights can be automatically determined via minimax optimization.
- It reaches a harmonic consensus by weighting the cost terms differently during minimax optimization, so that the disagreements among different feature modalities can be effectively reconciled.

Lastly, we study the collaborative multi-feature fusion in a transductive learning framework, where the labeled data can transfer the labels to the unlabeled data. The novelty of this work lies in:

- It allows different clustering results in different feature types, thus the feature co-occurrence patterns can be more flexible.
- Relying on the new feature co-occurrence representations of the data samples, we can propagate labels among data samples of similar feature co-occurrence pattern.
- Our new feature co-occurrence representation does not need to optimize individual feature type iteratively like in co-training, thus is less sensitive to noisy feature types.

1.4 Structure of the Thesis

The remainder of this thesis is organized as follows: Chapter 2 gives the literature review on visual pattern discovery and multi-feature fusion. Chapter 3 shows the first work where a multi-context-aware clustering algorithm is proposed for context-aware discovery of visual co-occurrence patterns. Chapter 4 presents the second work where a multi-feature spectral clustering algorithm is proposed for visual clustering and pattern discovery. Chapter 5 exhibits the third work where a multi-feature transductive learning algorithm is proposed for visual classification and pattern discovery. Chapter 6 concludes the thesis and addresses potential future work.

Chapter 2

Literature Review

2.1 Spatial Visual Pattern Discovery

Many approaches have been proposed to discover frequent spatial patterns of visual primitives. These methods can be generally divided into bottom-up visual pattern mining and top-down generative visual pattern modeling.

The bottom-up pattern discovery methods start with visual primitives and then merge these primitives until the larger visual patterns are found. The basic idea is shown in Figure 2.1 (a). First, each image is decomposed into a number of visual primitives.

FIGURE 2.1. Bottom-up (a) and top-down (b) visual pattern discovery.

Then, the visual primitives are quantized into visual words (colored in blue) by clustering. After that, by investigating frequent visual word configurations in image spatial space, two types of word co-occurrence compositions, *i.e.*, visual patterns {“cross”, “star”} and {“parallelogram”, “diamond”, “trapezoid”} are found. Finally, we locate all instances of both types of visual patterns. In contrast, the top-down methods start with the modeling of images and visual patterns and then infer the pattern discovery result. Figure 2.1 (b) illustrates the top-down method by using the latent Dirichlet allocation (LDA) to model images and visual patterns [47]. The basic idea is that images are represented as mixtures over visual patterns, where each pattern is characterized by a distribution over visual words. This is similar to describing a document by mixtures of topics, where each topic has its own word distribution. The pattern discovery is achieved by inferring the posterior distribution of visual pattern mixture variable given an image.

For the bottom-up visual pattern mining methods, the spatial co-occurrence is an underlying principle. The early studies usually rely on the transaction-based neighborhood representation, which encodes which visual words appear together in a given spatial neighborhood. For example, the representative methods include [36, 48, 49]. The method proposed in [48] performs clustering for visual pattern discovery. The frequent pattern mining methods [12] are used in [36, 49].

Since the frequency information of visual words is usually not included in the transaction representation, several methods propose to consider the word frequencies together with spatial neighborhood representation such as in [50, 51]. In [50], the authors propose a bag-to-set (B2S) representation to transform word frequencies into a longer transaction. However this representation cannot avoid the generation of artificial visual patterns. Recently the bag of frequent local histograms method proposed in [51] keeps all the frequency information without bringing artificial patterns. Similarly, the method in [52] attempts to find out histograms of pattern sets by multiple random selections.

The above methods are all proposed to mine frequent visual word compositions. However, we have to first quantize visual features of visual primitives to obtain the visual word vocabulary. A poor word quantization may degrade the visual pattern mining performance. Therefore there are also some word-free visual pattern mining methods such as in [53], [9], [54]. In [53], each image is randomly partitioned several times to generate a pool of subimages, followed by the common visual pattern discovery from the frequent appearing subimages. In [9], by feature indexing and matching, a hierarchical representation of spatially feature compositions are learned. The method proposed in [54] utilizes the property that the instances from the same visual pattern can be generated by each other to develop a visual category pattern discovery method called clustering by composition.

Besides the spatial co-occurrence visual pattern discovery, some researchers focus on discovering geometry preserving visual patterns. In [55], the authors use the relative spatial distance, relative scale difference and relative headings between pairwise visual words to represent the geometric relationship between them. The consistent geometric visual patterns are discovered by a further frequent subgraph pattern mining. A recent addition to common graph pattern methods is proposed in [56]. In [39], the authors derive an offset space by the relative location difference between each visual word pair in two images. Because the visual words exhibiting the same geometric relationship will be mapped to the same place on the offset space, the high-order geometry preserving visual patterns can be mined by Hough Voting. To preserve geometric information of visual patterns, one can also apply global geometric consistency checking Random sample consensus (RANSAC) [57], spatial coding [58] and geometric coding [59].

In addition to the above bottom-up visual pattern mining, there are also considerable methods in modeling visual patterns from top down, such as the constellation model in [60], the pictorial model in [61], the bayesian model in [62], the tree model in [63], [64], the active basis model in [65], the hierarchical model in [66–68], the conditional random field model in [69]. Specially, in [60], object classes are represented

as constellations of visual parts in a probabilistic model. In [61], the visual parts of a category of objects are modeled with deformable configuration in a pictorial structure. In [62], the spatial compositional structure of visual objects is learned using Bayesian network. In [63], object categories are detected by similar geometric, photometric and topological properties in a tree model. Another tree-structured graphical model proposed in [64] learns the object co-occurrences and spatial dependency relationships. In [65], it learns frequent occurring spatial structures in natural images using active basis model. The hierarchical spatial patterns presented in images are studied in [66–68]. The configurations of objects in scenes are discovered by using a conditional random field in [69].

In the previous approaches, visual patterns are discovered either via top down generative model, or from bottom up mining. However, the composition of visual patterns and the clustering of visual primitives influence each other. Therefore, a principled solution that can integrate the top-down and bottom-up processes is preferred [35, 37], where it can leverage visual patterns to improve the clustering of visual primitives. Moreover, besides spatial composition of visual primitives, because of the feature dependencies of a visual pattern across multiple types of features, there also needs to utilize feature co-occurrences of visual primitives for an enhanced discovery of visual patterns and clustering of visual primitives. But much less work can address such feature co-occurrence pattern mining. To address the problems mentioned above, this thesis will in Chapter 3 provide a self-supervised clustering procedure that can allow the spatial co-occurrence pattern discovery, feature co-occurrence discovery and visual primitive clustering to help one another.

2.2 Multi-feature Fusion

Multi-feature fusion is also known as multi-view learning [44], which targets at combining multiple complementary feature modalities to improve the result of clustering [70, 71], classification [72, 73] and image search [74–76]. Such a multi-feature fusion, however, is challenging due to the possible incompatibility of heterogeneous features. As a result, a simple concatenation of them does not guarantee good performance [42, 73]. To deal with diverse features, various approaches have been proposed [44], such as multiple kernel learning for a similarity level fusion [77–79], subspace learning for a common feature representation [80, 81], and co-training for a modality regularization [43, 82, 83].

To perform multi-feature fusion, different feature types can be separately exploited, and afterwards the results are combined with specific weights. The traditional way is to build a probabilistic model for each feature type, and then estimate a mixture of them [84, 85]. Alternatively, relying on kernel combination, one can use each feature type to compute a similarity kernel matrix for a weighted sum [86, 87]. In such a case, similarity matrices are weighted and combined for graph structure fusion [4, 70, 88]. In these methods, weighting scheme is critical to the effectiveness of multi-feature fusion.

Some other work chooses to seek a common subspace that shared by different feature types. In clustering level, one can learn a common pattern space by comparing with the pattern spaces of multiple feature modalities, such as pattern projection method [89], common non-negative matrix factorization (NMF) [90]. In feature level, one are able to look for a shared feature subspace by using, for example, canonical correlation analysis (CCA) [80, 91], general sparse coding [92], convex multi-view subspace learning [93], Pareto embedding [94], and structured feature selection [95].

Mutual regularization is also a successful approach in multi-feature clustering. The regularization is usually performed by disagreement minimization, which is widely investigated with k -means regularization [41, 96, 97], NMF regularization [98], topic

model regularization [99], and spectral embedding regularization [42, 43, 100], kernel multiplication [101] and low-rank affinity regularization [102]. Among these methods, pairwise regularization is a representative strategy. However, it generally outputs different solutions from multiple feature types such that a late fusion step is required. Some of these approaches actually apply the regularization between each feature modality and a centroid modality, *e.g.*, [42, 43, 96, 101, 102]. However, both the pairwise and centroid regularization still need to specify the weights to reflect the confidence of each regularization cost, which is difficult to determine without prior knowledge. For example with the recent approaches reported in [43], the first one is a pairwise regularization to push pairs of Laplacian embeddings of different feature types close to each other, while the other one is a centroid regularization to push all Laplacian embeddings of different feature types towards a consensus embedding for multi-feature clustering. Both the two methods, however, are formulated to assign equal weights to individual feature modalities, which are usually not the best choice in multi-feature clustering.

It is also worth noting that, in terms of multi-feature fusion, the existing work usually enforce the agreement among different feature types. For example, the method in [103] minimizes the disagreement of classifiers between two feature modalities. Similarly, the co-training methods train two classifiers separately from different feature types and make both classifiers agree on the labeling of the unlabeled data [82, 104]. As noted before, the way of CCA is to extract shared features from multiple feature types [80, 105, 106]. Learning an ensemble kernel from different feature types is adopted in [77–79]. To obtain a common embedding, many methods are proposed, such as multiview stochastic neighbor embedding [107], consensus pattern embedding [89], and spectral embedding [108, 109]. In addition, some other methods use metric fusion [71, 110].

Despite these previous advances, there is limited work that address the disagreement problem of different feature types in multi-feature learning. A conditional entropy criterion is introduced to detect modality disagreement caused by modality corruption or noise in [111]. However, even without the influence of modality corruption and noise,

samples in individual feature types still need not to belong to the same class. The recent work include context-aware clustering in [97] and [41], hierarchical sparse coding in [112], latent subspace Markov network in [113], and dual assignment clustering in [114] that incorporates the individual feature structures of multiple feature types for pattern clustering or classification. Especially in [97], the authors use the co-occurrences of feature clusters in different feature types to represent data samples. Because of this manipulation, the individual feature spaces from different feature types can have different data distributions. Although [97] can also handle multi-feature fusion, it targets at unsupervised clustering only. It is still not clear how to apply it in semi-supervised situation. With semi-supervised learning, we are able to leverage a small amount of labelled image data to enhance unsupervised clustering of visual patterns. We thus in Chapter 5 study how to address the disagreement problem in the transductive learning scenario.

Chapter 3

Context-Aware Discovery of Visual Co-occurrence Patterns

3.1 Introduction

Images can be decomposed into visual primitives, *e.g.*, local interest points or image regions. With each visual primitive described by a feature vector, it is of great interests to cluster these visual primitives into prototypes, *e.g.*, visual words. Then by representing an image as a visual document, conventional text analysis methods can be directly applied. Although it has been a common practice to build a visual vocabulary for image analysis, it remains a challenging problem to cluster visual primitives into meaningful patterns. Most existing visual primitive clustering methods ignore the spatial structure among the visual primitives [6], thus bringing unsatisfactory results. For example, a simple k -means clustering of visual primitives can lead to synonymous visual words that over-represent visual primitives, as well as polysemous visual words that bring large uncertainties and ambiguities in the representation [33, 34].

As visual primitives are not independent to each other, to better address the visual polysemous and synonymous phenomena, the ambiguities and uncertainties of visual primitives can be partially resolved through analyzing their spatial contexts [35, 37], *i.e.*,

other primitives in the spatial neighborhood. Two visual primitives, although exhibit dissimilar visual features, may belong to the same pattern if they have the same spatial contexts. Even though they share similar features, they may not belong to the same visual pattern if their spatial contexts are completely different. By considering the spatial dependency among visual primitives, much previous work has tried to discover the spatial co-occurrence patterns of visual primitives, *e.g.*, visual phrases [36, 38, 39]. Such co-occurrence patterns, once discovered, can be utilized to better address the ambiguities and uncertainties among visual primitives and have proven to be helpful for object categorization and image search.

Despite previous success of utilizing spatial visual patterns, there lacks a principled solution that can leverage visual patterns to improve the clustering of visual primitives. For example, [36] applies data mining methods first to find the spatial visual patterns. After that it relies on a separate sub-space learning to refine the presentation and clustering of visual primitives. As the clustering of visual primitives and the discovery of visual pattern will influence each other, it is preferred to provide a uniformed solution that can integrate visual pattern discovery into the process of visual primitives clustering. Moreover, besides the spatial dependencies among visual primitives, a visual pattern can exhibit certain feature dependencies as well. For example, instead of using a single feature vector, a sheep image region (as a visual primitive) can also be described by a co-occurrence of the white (color) and fur (texture) attributes. Since visual primitives can be easily described by multiple types of features or attributes, besides discovering spatial co-occurrence patterns, it is equally interesting in discovering feature/attribute co-occurrence patterns in an image. Such a feature/attribute co-occurrence discovery problem, however, is seldom addressed in the literature.

In this work, we propose to consider spatial and feature contexts among visual primitives for pattern discovery. By discovering spatial co-occurrence patterns among visual primitives and feature co-occurrence patterns among different types of features, our

FIGURE 3.1. A spatial pattern $\{\text{left eye}, \text{right eye}, \text{nose}, \text{mouth}\}$ co-occur frequently which forms multiple faces of different people.

method can effectively reduce the ambiguities of clustering visual primitives. We formulate the pattern discovery problem as a regularized k -means clustering where spatial and feature contexts are served as constraints to improve the pattern discovery results. A novel self-learning procedure is proposed to utilize the discovered spatial or feature patterns to gradually refine the clustering result. Our self-learning procedure is guaranteed to converge and experiments on real images validate the effectiveness of our method.

3.2 Multi-Context-Aware Clustering

3.2.1 Problem Statement

We will propose in this section the multi-context-aware clustering that utilizes both feature and spatial contexts. In this multi-context-aware clustering, each visual primitive

$v_n \in \mathcal{D}_v$ is characterized by c types of features: $v_n = \{\mathbf{f}_n^{(i)}\}_{i=1}^c$, where $\mathbf{f}_n^{(i)} \in \mathbb{R}^{d_i}$. For example, an image can be represented by color, shape, texture and any other visual features. These features of v_n correspond to a feature context group $\mathcal{G}_n^{(v)}$. Meanwhile, collocating with a visual primitive in a local spatial neighborhood, the inclusive visual primitives constitute the spatial contexts of the central one. Take human face image in Figure 3.1 for example, the spatial contexts of a nose contain two eyes and a mouth. For each visual primitive $v_n \in \mathcal{D}_v$, we denote by $\mathcal{G}_n^{(s)} = \{v_n, v_{n_1}, v_{n_2}, \dots, v_{n_K}\}$ its *spatial context group*, which can be built by K -nearest neighbors (K -NN) or ϵ -nearest neighbors (ϵ -NN).

By k -means clustering, each type of features $\{\mathbf{f}_n^{(i)}\}_{n=1}^N$ can produce a feature word lexicon Ω_i ($|\Omega_i| = M_i$). However, k -means clustering suffers from several flaws. First, it assumes all the visual primitives are independent to each other, and each potential cluster is convex in the feature space. Such assumptions are too strong to hold since primitives have spatial dependencies with one another and the clusters may have arbitrary shapes. Second, there are also feature dependencies among different types of features, which are discarded in the traditional k -means clustering method. As a result, we should take into account multiple types of features. To this end, we let each $v_n \in \mathcal{D}_v$ generate a feature context transaction $\mathbf{t}_n^{(v)} \in \mathbb{R}^{\sum_{i=1}^c M_i}$ to represent $\mathcal{G}_n^{(v)}$, which is defined in the following.

Definition 3.1 (Feature context transaction). The feature context transaction of data v_n , denoted by $\mathbf{t}_n^{(v)}$, refers to the co-occurrences of multiple types of feature words in the feature context group of v_n .

FIGURE 3.2. Multi-context-aware clustering via a self-learning procedure between low-level feature clustering and high-level pattern discovery.

Using label indicator matrices $\{\mathbf{R}_i\}_{i=1}^c$ obtained from k -means clustering on the c types of features, we can represent the feature context transaction database as a binary matrix:

$$\mathbf{T}_v = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \vdots \\ \mathbf{R}_c \end{bmatrix}, \quad (3.1)$$

where $\mathbf{R}_i \in \mathbb{R}^{M_i \times N}$, and the binary entry $r_{mn}^{(i)} = 1$ only if v_n is labeled with the m^{th} discovered feature word based on the i^{th} types of features $\{\mathbf{f}_n^{(i)}\}_{n=1}^N$; $\mathbf{T}_v \in \mathbb{R}^{\sum_{i=1}^c M_i \times N}$, and the n^{th} column of $\mathbf{T}_v$ is just the feature context transaction of data v_n , i.e., $\mathbf{t}_n^{(v)}$.

After clustering these N feature context transactions, the data points can be labeled by a high level feature pattern lexicon Ψ_v ($|\Psi_v| = M_v$), which partition the given data in $\mathcal{D}_v$ using multiple features.

Besides feature contexts, we can further explore the spatial dependencies among primitives to find a higher level spatial pattern lexicon Ψ_s ($|\Psi_s| = M_s$). For example in Figure 3.1, the existence of a spatial pattern $\mathcal{P} = \{left\ eye, right\ eye, nose, mouth\}$ shows that these four categories of visual primitives: left eye, right eye, nose and mouth co-occur frequently and form a meaningful visual pattern, *i.e.*, face. To this end, we use the concept–spatial context transactions–to record the spatial dependencies among visual primitives.

Definition 3.2 (Spatial context transaction). The spatial context transaction of the visual primitive v_n , denoted by $\mathbf{t}_n^{(s)}$, refers to the co-occurrences of different categories of visual primitives appearing in the spatial context group of v_n .

The spatial context transaction database can be represented as a sparse integer matrix $\mathbf{T}_s \in \mathbb{R}^{M_v \times N}$, where each column is a spatial context transaction $\mathbf{t}_n^{(s)} \in \mathbb{Z}^{M_v}$. The entry $t_{mn}^{(s)} = x$ indicates that the n^{th} transaction contains x visual primitives belonging to the m^{th} category.

Once we find spatial patterns, we can use them to tune the primitive clustering of uncertainty. Through such a top down refinement, spatial patterns can help to improve feature pattern constructions. Afterwards, each type of feature words will also be adjusted due to the tuned feature patterns. Then the multiple types of updated feature words can learn more accurate feature patterns and spatial patterns from bottom up again. The idea described above is shown in Figure 3.2 using three types of features. To achieve this objective, we propose a regularized k -means with feature contexts and spatial contexts together:

$$\begin{aligned}
 \mathbf{J} &= \sum_{i=1}^c \sum_{m=1}^{M_i} \sum_{n=1}^N r_{mn}^{(i)} \|\mathbf{u}_m^{(i)} - \mathbf{f}_n^{(i)}\|^2 \\
 &\quad + \lambda_v \sum_{m=1}^{M_v} \sum_{n=1}^N r_{mn}^{(v)} d_H(\mathbf{u}_m^{(v)}, \mathbf{t}_n^{(v)}) \\
 &\quad + \lambda_s \sum_{m=1}^{M_s} \sum_{n=1}^N r_{mn}^{(s)} d_H(\mathbf{u}_m^{(s)}, \mathbf{t}_n^{(s)}) \\
 &= \sum_{i=1}^c \text{tr}(\mathbf{R}_i^T \mathbf{D}_i) + \lambda_v \text{tr}(\mathbf{R}_v^T \mathbf{D}_v) + \lambda_s \text{tr}(\mathbf{R}_s^T \mathbf{D}_s) \\
 &= \underbrace{\text{tr}(\mathbf{R}^T \mathbf{D})}_{\mathbf{J}_\alpha} + \underbrace{\lambda_v \text{tr}(\mathbf{R}_v^T \mathbf{D}_v)}_{\mathbf{J}_\beta} + \underbrace{\lambda_s \text{tr}(\mathbf{R}_s^T \mathbf{D}_s)}_{\mathbf{J}_\gamma}, \tag{3.2}
 \end{aligned}$$

where,

- $\lambda_v > 0$ and $\lambda_s > 0$ are constants for regularization;
- $\mathbf{J}_\alpha$ is the total quantization distortions of multiple types of features, in which
  - $\{\mathbf{u}_m^{(i)}\}_{m=1}^{M_i}$ denote M_i quantized feature words based on the i_{th} type of features $\{\mathbf{f}_n^{(i)}\}_{n=1}^N$, and form a feature word matrix $\mathbf{U}_i \in \mathbb{R}^{d_i \times M_i}$;
  - $\mathbf{R}_i \in \mathbb{R}^{M_i \times N}$ is a binary label indicator matrix, the entry $r_{mn}^{(i)} = 1$ only if $\mathbf{f}_n^{(i)}$ is labeled with the m^{th} discovered feature word $\mathbf{u}_m^{(i)}$ based on the i^{th} type of features $\{\mathbf{f}_n^{(i)}\}_{n=1}^N$;
  - $\mathbf{D}_i \in \mathbb{R}^{M_i \times N}$ denotes a distortion matrix, the entry of its m^{th} row and n^{th} column is the Euclidean square distortion between $\mathbf{u}_m^{(i)}$ and $\mathbf{f}_n^{(i)}$;
  - $\mathbf{R}$ and $\mathbf{D}$ are block diagonal matrices from $\{\mathbf{R}_i\}_{i=1}^c$ and $\{\mathbf{D}_i\}_{i=1}^c$;
- $\mathbf{J}_\beta$ is the quantization distortion of feature context transactions, in which
  - $\{\mathbf{u}_m^{(v)}\}_{m=1}^{M_v}$ denote M_v quantized feature patterns, forming a feature pattern matrix $\mathbf{U}_v \in \mathbb{R}^{\sum_{j=1}^c M_j \times M_v}$;
  - $\mathbf{R}_v \in \mathbb{R}^{M_v \times N}$ is a binary label indicator matrix, the entry $r_{mn}^{(v)} = 1$ only if v_n is included the m^{th} discovered feature pattern $\mathbf{u}_m^{(v)}$;

- $\mathbf{D}_v \in \mathbb{R}^{M_v \times N}$ denotes a distortion matrix, the entry of its m^{th} row and n^{th} column is the Hamming distortion between $\mathbf{u}_m^{(v)}$ and $\mathbf{t}_n^{(v)}$;
- $\mathbf{J}_\gamma$ is the quantization distortion of spatial context transactions, in which
  - $\{\mathbf{u}_m^{(s)}\}_{m=1}^{M_s}$ denote M_s quantized spatial patterns, forming a spatial pattern matrix $\mathbf{U}_s \in \mathbb{R}^{M_v \times M_s}$;
  - $\mathbf{R}_s \in \mathbb{R}^{M_s \times N}$ is a binary label indicator matrix, the entry $r_{mn}^{(s)} = 1$ only if v_n is included the m^{th} discovered spatial pattern $\mathbf{u}_m^{(s)}$;
  - $\mathbf{D}_s \in \mathbb{R}^{M_s \times N}$ denotes a distortion matrix, the entry of its m^{th} row and n^{th} column is the Hamming distortion between $\mathbf{u}_m^{(s)}$ and $\mathbf{t}_n^{(s)}$;

Because each data point has c types of features, the feature context relations can be represented as a concatenated matrix $\mathbf{Q}_v \in \mathbb{R}^{cN \times N}$ from c identity matrices of size $N \times N$, the following equation is hold:

$$\mathbf{T}_v = \mathfrak{R}\mathbf{Q}_v. \quad (3.3)$$

Besides, according to the local spatial neighbor relation matrix of the primitive collection $\mathcal{D}_v$, i.e., an $N \times N$ matrix $\mathbf{Q}_s$ whose entry $q_{ij} = 1$ only if v_i and v_j are local spatial neighbors, we can represent $\mathbf{T}_s$ as

$$\mathbf{T}_s = \mathbf{R}_v\mathbf{Q}_s. \quad (3.4)$$

The Hamming distortion matrix $\mathbf{D}_v$ can be represented by

$$\begin{aligned} \mathbf{D}_v &= -2\mathbf{U}_v^T\mathbf{T}_v + \mathbf{1}_{\mathbf{T}_v}\mathbf{T}_v + \mathbf{U}_v^T\mathbf{1}_{\mathbf{U}_v} \\ &= -2\mathbf{U}_v^T\mathfrak{R}\mathbf{Q}_v + \mathbf{1}_{\mathbf{T}_v}\mathfrak{R}\mathbf{Q}_v + \mathbf{U}_v^T\mathbf{1}_{\mathbf{U}_v}, \end{aligned} \quad (3.5)$$

where $\mathbf{1}_{\mathbf{T}_v}$ is an $M \times \sum_{i=1}^c M_i$ all 1 matrix, and $\mathbf{1}_{\mathbf{U}_v}$ is a $\sum_{i=1}^c M_i \times N$ all 1 matrix .

In a similar way, the Hamming distortion matrix $\mathbf{D}_s$ can be formulated as¹

$$\begin{aligned}\mathbf{D}_s &= -2\mathbf{U}_s^T \mathbf{T}_s + \mathbf{1}_{\mathbf{T}_s} \mathbf{T}_s + \mathbf{U}_s^T \mathbf{1}_{\mathbf{U}_s} \\ &= -2\mathbf{U}_s^T \mathbf{R}_v \mathbf{Q}_s + \mathbf{1}_{\mathbf{T}_s} \mathbf{R}_v \mathbf{Q}_s + \mathbf{U}_s^T \mathbf{1}_{\mathbf{U}_s},\end{aligned}\quad (3.6)$$

where $\mathbf{1}_{\mathbf{T}_s}$ is an $M_s \times M$ all 1 matrix, and $\mathbf{1}_{\mathbf{U}_s}$ is an $M \times N$ all 1 matrix.

As $\mathbf{J}_\alpha$, $\mathbf{J}_\beta$ and $\mathbf{J}_\gamma$ are correlated among each other, it is intractable to minimize $\mathbf{J}$ by minimizing the three terms separately, which makes the objective function of Eq. 3.2 a challenge. We will in Sec. 3.2.2 show how to decouple the dependencies among them and propose our algorithm to solve this optimization function.

3.2.2 Algorithm

Since there are correlations among unknown variables simultaneously in Eq. 3.2, we cannot estimate them simultaneously. So we have to decouple the dependencies among the terms of Eq. 3.2. Above all, we initialize feature words, feature patterns and spatial patterns gradually by k -means. Next, we can take each of $\mathbf{R}_v$, $\mathbf{R}$, and $\mathbf{R}_s$ as the common factor for extraction. We derive Eq. 3.2 as:

$$\begin{aligned}\mathbf{J}(\mathfrak{R}, \mathbf{R}_v, \mathbf{R}_s, \mathfrak{D}, \mathbf{D}_v, \mathbf{D}_s) \\ = \text{tr}(\mathbf{R}_v^T \mathbf{H}_v) + \text{tr}(\mathfrak{R}^T \mathfrak{D}) + \lambda_s \text{tr}(\mathbf{R}_s^T \mathbf{U}_s^T \mathbf{1}_{\mathbf{U}_s})\end{aligned}\quad (3.7)$$

$$= \text{tr}(\mathfrak{R}^T \mathfrak{H}) + \lambda_s \text{tr}(\mathbf{R}_s^T \mathbf{D}_s) + \lambda_v \text{tr}(\mathbf{R}_v^T \mathbf{U}_v^T \mathbf{1}_{\mathbf{U}_v})\quad (3.8)$$

$$= \text{tr}(\mathbf{R}_s^T \mathbf{H}_s) + \text{tr}(\mathfrak{R}^T \mathfrak{D}) + \lambda_v \text{tr}(\mathbf{R}_v^T \mathbf{D}_v),\quad (3.9)$$

¹Although spatial context transactions are not necessarily binary, the distortion derived from Hamming distance can also be perceived as a penalty measure, which is called Hamming distortion.

where

$$\mathbf{H}_v = \lambda_v \mathbf{D}_v - \lambda_s (2\mathbf{U}_s^T - \mathbf{1}_{T_s})^T \mathbf{R}_s \mathbf{Q}_s^T, \quad (3.10)$$

$$\mathfrak{H} = \mathfrak{D} - \lambda_v (2\mathbf{U}_v^T - \mathbf{1}_{T_v})^T \mathbf{R}_v \mathbf{Q}_v^T, \quad (3.11)$$

$$\mathbf{H}_s = \lambda_s \mathbf{D}_s, \quad (3.12)$$

where the size of $\mathbf{H}_v$, $\mathfrak{H}$ and $\mathbf{H}_s$ are $M \times N$, $\sum_{i=1}^c M_i \times cN$ and $M_s \times N$, and $\mathbf{H}$ contains c diagonal blocks $\{\mathbf{H}_i\}_{i=1}^c$.

We can successively update the three label indicator matrices $\mathbf{R}_v$, $\mathfrak{R}$, and $\mathbf{R}_s$ when the cluster centroid matrices $\mathbf{U}_v$, $\{\mathbf{U}_i\}_{i=1}^c$, and $\mathbf{U}_s$ are fixed. To minimize Eq. 3.2, the following label indicator matrices update criteria will be adopted, $\forall n = 1, 2, \dots, N$,

$$r_{mn}^{(v)} = \begin{cases} 1 & m = \arg \min_k h_{kn}^{(v)} \\ 0 & otherwise \end{cases}, \quad (3.13)$$

$$r_{mn}^{(j)} = \begin{cases} 1 & m = \arg \min_k h_{kn}^{(j)} \\ 0 & otherwise \end{cases}, \quad (3.14)$$

$$r_{mn}^{(s)} = \begin{cases} 1 & m = \arg \min_k h_{kn}^{(s)} \\ 0 & otherwise \end{cases}, \quad (3.15)$$

where $h_{kn}^{(v)}$, $r_{mn}^{(v)}$, $h_{kn}^{(i)}$, $r_{mn}^{(i)}$, $h_{kn}^{(s)}$ and $r_{mn}^{(s)}$ are the entries of $\mathbf{H}_v$, $\mathbf{R}_v$, $\mathbf{H}_i$, $\mathbf{R}_i$, $\mathbf{H}_s$ and $\mathbf{R}_s$, respectively. As long as the objective function $\mathbf{J}$ defined by Eq. 3.2 is decreasing, $\mathbf{R}_v$ and $\mathbf{R}$ can be continually refined, followed by the bottom-up updates of $\mathbf{R}_v$ and $\mathbf{R}_s$.

Furthermore, provided the label indicator matrices $\mathbf{R}_v$, $\mathbf{R}$, and $\mathbf{R}_s$, the corresponding centroid matrices $\mathbf{U}_v$, $\{\mathbf{U}_i\}_{i=1}^c$, and $\mathbf{U}_s$ can be updated, and so as the corresponding distortion matrices $\mathbf{D}_v$, $\{\mathbf{D}_i\}_{i=1}^c$, and $\mathbf{D}_s$, which will also make the objective function of Eq. 3.2 decrease.

We propose an iterative algorithm of multi-context-aware clustering in Alg. 1. This

Algorithm 1: MULTI-CONTEXT-AWARE CLUSTERING

Input: data $\mathcal{D}_v = \{v_n\}_{n=1}^N$; feature neighbor relations $\mathbf{Q}_v$; spatial neighbor relations $\mathbf{Q}_s$; parameters: $\{M_i\}_{i=1}^c, M_v, M_s, \lambda_v, \lambda_s$

Output: feature word lexicons: $\{\Omega_i\}_{i=1}^c$ ($\{\mathbf{U}_i\}_{i=1}^c$); feature pattern lexicon: $\Psi_v(\mathbf{U}_v)$; spatial pattern lexicon: $\Psi_s(\mathbf{U}_s)$; clustering results $\{\mathbf{R}_i\}_{i=1}^c, \mathbf{R}_v, \mathbf{R}_s$

// Initialization

- 1: perform k -means clustering from bottom up to obtain $\{\mathbf{U}_i\}_{i=1}^c, \mathbf{U}_v, \mathbf{U}_s$
- // Main loop
- 2: **repeat**
- 3:   **repeat**
- 4:   **R-step:** fix $\{\mathbf{U}_i\}_{i=1}^c, \mathbf{U}_v, \mathbf{U}_s$, successively top-down / bottom-up update $\{\mathbf{R}_i\}_{i=1}^c, \mathbf{R}_v, \mathbf{R}_s$
- 5:   **until** $\mathbf{J}$ is not decreasing
- 6:   **D-step:** fix $\{\mathbf{R}_i\}_{i=1}^c, \mathbf{R}_v, \mathbf{R}_s$, update $\{\mathbf{U}_i\}_{i=1}^c, \mathbf{U}_v, \mathbf{U}_s$
- 7:   **until** $\mathbf{J}$ is converged
- // Solution
- 8: **return** $\{\mathbf{U}_i\}_{i=1}^c, \mathbf{U}_v, \mathbf{U}_s, \{\mathbf{R}_i\}_{i=1}^c, \mathbf{R}_v, \mathbf{R}_s$

Algorithm is convergent since the solution spaces of $\mathfrak{R}$, $\mathbf{R}_v$, and $\mathbf{R}_s$ are discrete and finite, and the objective function defined by Eq. 3.2 is monotonic decreasing at each step. Clearly, our multi-context-aware clustering will be degenerated to the spatial context-aware clustering [96] if there is only one type of feature and we set $\lambda_v = 0$ in Eq. 3.2 to remove the $\mathbf{J}_\beta$ term. The complexity of the proposed algorithm is similar to k-means, since our method only needs a finite runs of k-means clustering.

3.3 Experiments

In the following experiments, we set $M_i = M_v, \forall i = 1, 2, \dots, c$ in multi-context-aware clustering. Besides, to help parameter tuning, we let $\lambda_s = \tau_s |\mathbf{J}_1^0 / \mathbf{J}_2^0|$ in spatial context-aware clustering, $\lambda_v = \tau_v |\mathbf{J}_\alpha^0 / \mathbf{J}_\beta^0|$ and $\lambda_s = \tau_s |\mathbf{J}_\alpha^0 / \mathbf{J}_\gamma^0|$ in multi-context-aware clustering, where $\mathbf{J}_x^0$ ($x = 1, 2, \alpha, \beta, \gamma$) is the initial value of $\mathbf{J}_x$ defined by Eq. 3.2, and the nonnegative constants τ_v and τ_s are the auxiliary parameters to balance the influences from feature co-occurrences and spatial co-occurrences, respectively.

FIGURE 3.3. Pattern discovery from a mono-colored LV monogram picture. Best viewed in color.

3.3.1 Spatial Visual Pattern Discovery

To validate whether our methods can really capture spatial visual patterns [53], we test a number of images using spatial context-aware clustering (one feature type, [96]) and multi-context-aware clustering (multiple feature types, our extension) presented in Figs. 3.3 and 3.4. Given an image, we first extract one or more (*e.g.*, c types of) features for the detected visual primitives: $\mathcal{D}_v = \{v_n\}_{n=1}^N$, and apply spatial K -NN groups to build spatial context group database $\{\mathcal{G}_n^{(s)}\}_{n=1}^N$.

A mono-colored LV monogram fabric image is shown in Figure 3.3. Because of cloth warping, the monogram patterns are deformed, which makes pattern discovery more challenging. We detect 2604 image patches as visual primitives, and use SIFT features to describe them [15]. To build spatial context groups, K -NN with $K = 8$ is applied. Other parameters are set as $M_1 = 20$, $M_s = 4$, $\tau_s = 1$ for our spatial context-aware clustering. It is interesting to notice that we can also locate the monogram patterns of different spatial structures in Figure 3.3, where different colors indicate different (4 in total) discovered spatial patterns. In comparison, without considering spatial dependencies of visual primitives, k -means clustering cannot obtain satisfactory results.

A comparison between spatial context-aware clustering and multi-context-aware clustering is shown in Figure 3.4, where 422 image patches [15] are extracted². In spatial context-aware clustering, SIFT features [15] are used to describe these patches. While in multi-context-aware clustering, the patches are represented by SIFT features [15] and Color Histograms (CH) [115]. Both methods construct spatial context groups by K -NN with $K = 12$, and aim to detect three categories of spatial patterns: human faces, text logos, and background edges. We highlight the instances of each discovered spatial pattern. The 1st column shows the results of spatial context-aware clustering; parameters are: $M_1 = 10$, $M_s = 3$, $\tau_s = 0.8$. The results of the 2nd column is based on multi-context-aware clustering; parameters are: $M_i = 10$, $\forall i = 1, 2$, $M_v = 10$, $M_s = 3$, $\tau_v = 1.5$, $\tau_s = 0.8$. By using multiple features, the discovered patterns are more

²This image is from source: <http://www.abf-online.org/>.

Spatial pattern cluster 1

Spatial pattern cluster 1

Spatial pattern cluster 2

Spatial pattern cluster 2

Spatial pattern cluster 3

(a) Spatial context-aware clustering

Spatial pattern cluster 3

(b) Multi-context-aware clustering

FIGURE 3.4. Pattern discovery from a colored group photo. Best viewed in color.

accurate. Particularly, in spatial context-aware clustering, there are more confusions between face patterns and edge patterns than those in multi-context-aware clustering.

3.3.2 Image Region Clustering using Multiple Contexts

To evaluate how much feature contexts and spatial contexts can improve the clustering performance, in this section, we perform image region clustering to test our proposed multi-context-aware clustering on MSRC-V2 dataset [116]. The ground-truth labeling of MSRC-V2 is provided by [117]. To describe each region (*i.e.*, the visual primitive)

FIGURE 3.5. Two scene categories: “sheep+grass” and “bicycle+road”.

TABLE 3.1. Clustering performance of image regions from image collection shown in Figure 3.5.

Feature	Method	Error	Precision	Recall
TH	k -means	34.43%	65.41%	66.46%
CH	k -means	41.80%	53.56%	59.38%
pHOG	k -means	37.30%	55.07%	60.16%
All features	Concatenation for k -means	40.98%	55.22%	60.01%
All features	Concatenation with context transactions for k -means	10.65%	91.87%	89.57%
All features	Multi-context-aware	5.73%	94.67%	94.48%

in MSRC-V2 dataset, we employ three types of features as in [115]: Texton Histogram (TH), Color Histogram (CH), and pyramid of HOG (pHOG) [118]. The dimensionalities of TH, CH, pHOG, are 400, 69, 680, respectively.

Given an image region, all other regions in the same image are considered as in its spatial context group. Each scene category has its own region compositions and our goal is to cluster image regions by leveraging the spatial co-occurrence patterns. As baseline methods, k -means clustering results on each single feature type (*TH/CH/pHOG*, k -means) or feature concatenation (*All features*, *concatenation for k -means*) are compared. Besides that, as context features, feature context transactions and spatial context transactions are concatenated with all feature types for k -means clustering (*All features*, *concatenation with context transactions for k -means*) to compare with our multi-context-aware clustering on multiple types of features (*All features*, *multi-context-aware*).

FIGURE 3.6. Class disambiguation by using spatial contexts.

From MSRC-V2 dataset [116], we first select a collection of images with two region pairs that often appear together in an image: “sheep+grass” and “bicycle+road” as shown in Figure 3.5. Each region pair has 27 image instances. There are in total 31 sheep regions, 32 grass regions, 27 bicycle regions, and 32 road regions. Because the spatial contexts of a region are the regions occurring in the same image, the spatial contextual relations only appear between regions of “sheep” and “grass” or regions of “bicycle” and “road”.

Table 3.1 shows the clustering results of k -means clustering on individual features and concatenated features, where the best one is with error rate 10.65%, average of precision 91.87%, and average of recall 89.57%. In contrast, with suitable parameters, our multi-context-aware clustering can achieve a much better result with error rate 5.73%, average of precision 94.67%, and average of recall 94.48%, which significantly enhances the clustering performance. The parameters used are: $k = 4$ for k -means clustering; and $M_i = 4$, $\forall i = 1, 2, 3$, $M_v = 4$, $M_s = 2$, $\tau_v = 3.5$, $\tau_s = 1$ for multi-context-aware clustering. Each result in Table 3.1 is obtained by selecting the best (*i.e.*, the minimum total distortion) from 100 random repetitions.

In the case of using multiple features, we cannot expect the performance of feature concatenation is better than that of each single feature type. This is because it is unknown which of the feature types the concatenation will be partial to. So it is not surprised that feature concatenation for k -means only achieves the comparable accuracy among the best of some single feature type. Even if with context transactions concatenated,

FIGURE 3.7. Confusion matrices of k -means clustering and multi-context-aware clustering for image regions from image collection shown in Figure 3.5.

k -means clustering still suffers from the confusion between “sheep” class and “road” class as shown in Figure 3.6, where the “sheep” regions are mislabeled as the “road” class. However, by exploring spatial contexts of image regions, our multi-context-aware clustering can better distinguish the two classes. Specifically, “grass” regions are in favor of labeling their co-occurring image regions as the “sheep” class; and similarly, the “bicycle” regions with correct labels can support the co-occurring “road” regions.

In order to further evaluate how multiple contexts improve the clustering of individual region classes, we show the confusion matrices of k -means clustering and our multi-context-aware clustering in Figure 3.7. We observe that k -means clustering easily mislabeled “bicycle” as “sheep” when using TH features. This is because these TH features encode the texture of regions, and “sheep” regions have similar texture to “bicycle” regions. When using CH features, it is easy to mislabel “sheep” regions as “road” regions because of their similar colors. Also, with similar shape features, quite a lot of “sheep”

FIGURE 3.8. Sample images of five region compositions: “sheep+grass”, “cow+grass”, “aeroplane+grass+sky”, “boat+water”, and “bicycle+road”.

regions are mislabeled as “bicycle” class when using pHOG features. Besides the limited description ability of a single type of feature, as k -means does not consider the spatial dependencies among regions, it also cause confusions among different classes. By considering the feature co-occurrences of CH, TH and pHOG, and the spatial co-occurrences of “sheep” and “grass” regions, as well as “bicycle” and “road” regions, our multi-context-aware clustering can well improve the clustering results on individual features, and finally reduce the confusion among the region classes. Specifically, our method can leverage the “grass” regions to correct the confused “sheep” regions, and vice versa. A similar improvement can be observed for “bicycle” and “road”.

In the above experiment, we show the advantage of our multi-context-aware clustering in dealing with image regions of clear spatial contexts. We next evaluate our method on image regions of ambiguous spatial contexts. In this experiment, we focus on images describing frequent occurring region compositions and select 5 different compositions

TABLE 3.2. Clustering performance of image regions from image collection shown in Figure 3.8.

Feature	Method	Error	Precision	Recall
TH	k -means	44.31%	55.60%	55.69%
CH	k -means	55.21%	41.60%	39.64%
pHOG	k -means	47.63%	46.33%	55.23%
All features	Concatenation for k -means	38.39%	57.56%	54.96%
All features	Concatenation with context transactions for k -means	33.65%	65.09%	66.79%
All features	Multi-context-aware	29.86%	71.99%	69.86%

with in total 9 categories of image regions from MSRC-V2 dataset [116]. Some sample images are shown in Figure 3.8. In this image collection, there are 30 “sheep+grass”, 29 “cow+grass”, 30 “aeroplane+grass+sky”, 31 “boat+water”, and 30 “bicycle+road”. The numbers of “sheep”, “grass”, “cow”, “sky”, “aeroplane”, “boat”, “water”, “bicycle”, “road” are 34, 104, 34, 53, 30, 47, 39, 30, and 51, respectively. Notice that in this challenging dataset, different image regions may share the same spatial context. For example, “grass” occurs in three different scenes: “sheep+grass”, “cow+grass”, and “aeroplane+grass+sky”.

The results of k -means clustering and multi-context-aware clustering are shown in Table 3.2, where the same 10% seeds per category from ground-truth are randomly chosen for initialization. The clustering error rate of our multi-context-aware clustering is 29.86%. It brings a considerable improvement than the best one (*i.e.*, 33.65%) obtained by k -means clustering on the individual features or the concatenated multiple features. We can obtain similar observation in terms of average of precision and average of recall. In k -means clustering, we set $k = 9$ as there are 9 different types of image regions. Similar to the setting on the dataset shown in Figure 3.5, we also set $\tau_v = 3.5$, $\tau_s = 1$ in multi-context-aware clustering. Other parameters used in our approach are $M_i = 9$, $\forall i = 1, 2, 3$, $M_v = 9$, $M_s = 5$.

Besides the total clustering error rates, we also compare the clustering performance of individual region classes using k -means clustering and our multi-context-aware clustering in Figure 3.9. Because both “sheep” and “cow” share the same spatial context, *i.e.*, “grass”, it is difficult to utilize “grass” to distinguish “sheep” and “cow” using the TH and pHOG features. However, our approach can leverage the CH features to distinguish “sheep” and “cow”. Meanwhile, with the help of spatial context regions, the mislabeled “bicycle” and “aeroplane” regions in CH features and the mislabeled “boat” regions in TH and pHOG features can be partially corrected using our method. Overall, due to the regularization of feature contexts and spatial contexts, our approach can effectively refine the mislabeling results of k -means clustering.

FIGURE 3.9. Confusion matrices of k -means clustering and multi-context-aware clustering for image regions from image collection shown in Figure 3.8.

Category	True positive	False positive
Grass		
Cow		
Sheep		
Sky		
Aeroplane		
Water		
Bicycle		
Road		
Boat		

FIGURE 3.10. Exemplar clustering results of multi-context-aware clustering.

Some representative clustering results of our approach are shown in Figure 3.10. Despite large intra-class variations, our method can still obtain a satisfactory clustering results by using both spatial and feature contexts. For example, the “cow” regions are with different colors and perspectives. We also note that in Figure 3.8, there contain “water” regions in some “sheep+grass” and “cow+grass” region compositions. These small amount of “water” regions are mislabeled as “grass” class because of its preference of “cow”/“sheep” contexts. Moreover, because the feature appearance and spatial contexts are similar, there still exist confusions between a few regions of “sheep” and “cow”, “bicycle” and “sheep”, “boat” and “aeroplane”, “water” and “sky”, “boat” and “bicycle”, and “water” and “road”. Nevertheless, the mislabeling results are only among the minority.

TABLE 3.3. Classification Accuracy on UIUC sports dataset.

Feature	BOW (k -means)	BOW (our method)
Accuracy	$71.78 \pm 1.52\%$	$73.42 \pm 1.42\%$

3.3.3 Bag-of-Words for Image Classification

To justify the superior of our method in building a better visual vocabulary than k -means clustering, we conduct an experiment on the UIUC sports dataset [119]. It contains 8 categories of sport events: rowing, badminton, polo, bocce, snowboarding, croquet, sailing, and rock climbing, with 137 to 250 images in each. Some sample images are shown in Figure 3.11. We use MSER detector [120] to generate visual primitives and use SIFT descriptor [15] for primitive representation. Finally, we obtain 4,997,849 visual primitives totally. After that, we can build a bag-of-word representation (BOW) for each image using k -means clustering or our method. We then train a SVM classifier with RBF kernel using 90 images per category and test on the rest over 5 random training/test splits. In k -means clustering, we set k to 256. In our spatial context-aware clustering, K -NN with $K = 8$ is applied to build spatial context groups. In addition, we set $\tau_s = 0.5$, $M_1 = 256$, and $M_s = 64$. We report the results in Table 3.3. As our method learns visual vocabulary with the spatial configuration information of visual primitives which ignored by k -means clustering, we can see that the BOW features built by our method outperforms than those built by k -means clustering.

3.3.4 Parameter Comparison and Selection

We use the image dataset shown in Figure 3.5 to study how the two parameters τ_v and τ_s influence the image region clustering result. It is worth noting that in our multi-context clustering, a larger τ_v and a smaller τ_s prefers to follow the multiple feature spaces and group image regions of similar features together; while a smaller τ_v and a larger τ_s favor the clustering results that support the discovered spatial patterns, thus regions are more

FIGURE 3.11. Sample images of different sports events in UIUC sports dataset.

 FIGURE 3.12. Clustering error rate of multi-context-aware clustering for image regions from image collection shown in Figure 3.5 w.s.t. parameters: τ_s and τ_v

likely to be grouped together in view of similar spatial contexts. To study the impact of parameters, we fix $M_i (i = 1, 2, 3)$, M_v , M_s , but vary τ_v and τ_s in turn. We then draw the curves of clustering error rates in Figure 3.12. When we fix $\tau_s = 1$ and alter τ_v from $[1, 5]$, the clustering error rates fall into $[5.73\%, 10.66\%]$. The best result (*i.e.*, 5.73% error rate) can be obtained when $\tau_v = 3.5$, which balances the distortions between primitive features and visual patterns. Then we fix $\tau_v = 3.5$ and alter τ_s from $[0.1, 2.0]$. The largest error rate of 12.3% is still much better than the best one using k -means clustering. The best agreement is reached when $\tau_s = 1$ by balancing the feature domain and spatial domain, where the lowest error rate 5.73% is attained. The experiment also suggests the optimal parameters for this dataset are: $\tau_v = 3.5, \tau_s = 1$, which we have used to report the result in Table 3.1.

Besides the two explicit parameters τ_v and τ_s studied in Figure 3.12, we discuss how to set the other parameters involved in our method. Firstly, similar to the build of visual vocabulary using k -means clustering, it is data dependent to set the feature word number (M_i), feature pattern number (M_v) and spatial pattern number (M_s). For example, in image region clustering experiment (Sec. 3.3.2), the numbers of feature words and feature patterns are determined by the region class number, while the number of spatial patterns is determined by the image class number. Secondly, regarding spatial neighborhood size (K) of visual primitives, we notice that a too small or too big K will not provide a satisfactory result. When treating a single image (*e.g.*, images shown in Figure 3.3 and Figure 3.4), the optimal choice of K depends on sizes of the spatial patterns. However, in the image region clustering experiment, we do not need to specify the size of spatial neighborhood, as given a region of an image, all other regions in the same image are considered as in its spatial context group. In such a case, the neighborhood size is automatically set. Moreover, in the image classification experiment (Sec. 3.3.3), it does not require discovering the accurate spatial patterns, but focuses on encoding the spatial configuration of visual primitives into visual vocabulary. Such spatial structure can be learned based on a flexible choice of the sizes of feature words, spatial patterns

and spatial neighborhood.

3.4 Conclusion

Because of the structure and content variations of complex visual patterns, they greatly challenge most existing methods to discover meaningful visual patterns in images. We propose a novel pattern discovery method to construct low-level visual primitives, *e.g.*, local image patches or regions, into high-level visual patterns of spatial structures. Instead of ignoring the spatial dependencies among visual primitives and simply performing k -means clustering to obtain the visual vocabulary, we explore spatial contexts and discover the co-occurrence patterns to resolve the ambiguities among visual primitives. To solve the regularized k -means clustering, an iterative top-down/bottom-up procedure is developed. Our proposed self-learning procedure can iteratively refine the pattern discovery results and guarantee to converge. Furthermore, we explore feature contexts and utilize the co-occurrence patterns among multiple types of features to handle the content variations of visual patterns. By doing so, our method can leverage multiple types of features to further improve the performance of clustering and pattern discovery. The experiments on spatial visual pattern discovery, image region clustering and image classification validate the advantages of the proposed method.

Chapter 4

Multi-feature Spectral Clustering with Minimax Optimization

4.1 Introduction

In visual recognition, the image or video data can be naturally characterized by multiple types (modalities) of features to describe different aspects of visual characteristics, such as color, texture, or motion. Instead of using a single feature type to perform image or video categorization, it is of great interests to combine multiple complementary feature modalities to improve the clustering or classification result [70–72]. Although in Chapter 3 the proposed method can handle multiple feature types, it only applies to spatial co-occurrence pattern discovery by combining spatial contexts and feature contexts of visual primitives. By contrast, this chapter aims to capture image or video clusters by multi-feature clustering for visual pattern discovery.

Such a multi-feature fusion, however, is challenging due to the possible incompatibility of heterogeneous features. For example, a simple concatenation of them does not guarantee good performance [42]. To perform better multi-feature clustering, some previous approaches choose to exploit different feature types separately followed by

combining the results through a weighted fusion. However, it is difficult to determine the fusion coefficients for different feature modalities. Kernel fusion k -means [87] and affinity aggregation spectral clustering [70] are recent additions to this family. Alternatively, some approaches aim to seek a feature representation or clustering result that can fit in different feature types. Different fitting criteria can bring a variety of consensus methods, such as correlational spectral clustering [80] and common convex representation [93]. Some recent methods propose to enforce the clustering results of different feature types to agree with each other by mutual regularization of each pair of feature modalities. For example, Kumar *et al.* [43] performed pairwise regularization to push pairs of Laplacian embeddings of different feature types close to each other, as well as centroid regularization to push all Laplacian embeddings of different feature types towards a consensus embedding for multi-feature clustering. However, both the pairwise and centroid regularization still need to specify the weights to reflect the confidence of each feature modality, which is difficult to determine without prior knowledge.

In this chapter, we propose a novel formulation for multi-feature clustering using minimax optimization. To find a consensus that is agreeable to all feature modalities, our objective is to find a universal feature embedding, which not only fits each individual feature modality well, but also unifies different feature modalities by minimizing their pairwise disagreements. The loss function consists of both (1) unary embedding cost for each modality, and (2) pairwise disagreement cost for each pair of modalities, with weighting parameters automatically selected to maximize the loss. By performing minimax optimization, we can minimize the loss for the worst case with maximum disagreements, thus can better reconcile different feature modalities. To solve the minimax optimization, an iterative solution is proposed to update the universal embedding, individual embedding, and fusion weights, separately. Our minimax optimization has only one global parameter. The superior results on various multi-feature clustering tasks validate the effectiveness of our approach when compared with the state-of-the-art methods.

4.2 Proposed Method

4.2.1 Laplacian Embedding

Spectral embedding of data Laplacian matrix is widely used to disclose data clustering structure [17, 121]. Given N data samples $\mathcal{X} = \{x_i\}_{i=1}^N$ and the corresponding feature descriptors $\mathcal{F} = \{\mathbf{f}_i\}_{i=1}^N$ of a specific feature type, one can compute the similarity between any sample pair using Gaussian kernel:

$$w_{ij} = \exp\left\{\frac{-\text{dist}^2(\mathbf{f}_i, \mathbf{f}_j)}{2\sigma^2}\right\}, \quad (4.1)$$

where $\text{dist}(\mathbf{f}_i, \mathbf{f}_j)$ denotes the distance between a pair of feature descriptors; σ is the bandwidth parameter. All pairwise similarities compose the similarity matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$, which is further transformed to the $N \times N$ normalized Laplacian matrix:

$$\mathcal{L} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{I} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}}, \quad (4.2)$$

where $\mathbf{D} \in \mathbb{R}^{N \times N}$ is the diagonal degree matrix, whose diagonal entries are given by $d_{ii} = \sum_{j=1}^N w_{ij}$. Spectral embedding is to optimize the following problem [17, 121]:

$$\begin{aligned} & \underset{\mathbf{U} \in \mathbb{R}^{N \times K}}{\text{minimize}} \quad \mathcal{Q} = \text{tr}(\mathbf{U}^T \mathcal{L} \mathbf{U}) \\ & \text{subject to} \quad \mathbf{U}^T \mathbf{U} = \mathbf{I}. \end{aligned} \quad (4.3)$$

The Rayleigh-Ritz theorem [122] states that the solution of $\mathbf{U}$ consists of the eigenvectors corresponding to the first K smallest eigenvalues (the first K smallest eigenvectors) of $\mathcal{L}$. In spectral clustering, the entries of $\mathbf{U}$ measure the similarities between data samples and clusters, as they indicate how likely data samples belong to specific clusters. A simple k -means clustering on rows of $\mathbf{U}$ can transform the real valued cluster similarities into discrete cluster indicators [17, 121].

4.2.2 Regularized Data-Cluster Similarity Matrix

For each feature type, we follow [43] to obtain the *Data-Data Similarity Matrix* by inner product:

$$\mathbf{S}(\mathbf{U}) = \mathbf{U}\mathbf{U}^T. \quad (4.4)$$

Let $\mathbf{V} \in \mathbb{R}^{N \times K}$ be the final cluster indicator matrix agreed among multiple feature types. We define the *Regularized Data-Cluster Similarity Matrix* as the projection of $\mathbf{S}$ onto $\mathbf{V}$:

$$\mathbf{P}_V(\mathbf{U}) = \mathbf{S}(\mathbf{U})\mathbf{V} = \mathbf{U}\mathbf{U}^T\mathbf{V}. \quad (4.5)$$

Compared to the original data-cluster similarity matrix $\mathbf{U}$, the regularized data-cluster similarity matrix $\mathbf{P}_V(\mathbf{U})$ measures the data-cluster similarity of each data sample with the final clustering solution $\mathbf{V}$. In the following, we will relax the final clustering solution $\mathbf{V}$ to be a real-valued *universal feature embedding with orthonormal constraints*: $\mathbf{V}^T\mathbf{V} = \mathbf{I}$. As a result, Eq. 4.5 enables self projection to be invariant:

$$\mathbf{P}_V(\mathbf{V}) = \mathbf{V}\mathbf{V}^T\mathbf{V} = \mathbf{V}. \quad (4.6)$$

4.2.3 Towards Agreement among Regularized Data-Cluster Similarity Matrices

Suppose we have M different types of features in total. Our motivation is to encourage the regularized data-cluster similarity matrices to be similar between any two feature types, *e.g.*, type i and type j . Therefore, we propose to minimize the following disagreement measure:

$$\mathcal{D}_V(\mathbf{U}_i, \mathbf{U}_j) = \|\mathbf{P}_V(\mathbf{U}_i) - \mathbf{P}_V(\mathbf{U}_j)\|_F^2. \quad (4.7)$$

Instead of forcing pairwise data-data similarity matrices to agree between two feature types in [43], we relax the constraint to data-cluster similarity matrices for noise suppression. Besides that, we propose an additional requirement that the two feature embeddings $\mathbf{U}_i$ and $\mathbf{U}_j$ in Eq. 4.7 should accommodate the universal feature embedding $\mathbf{V}$. Thus $\mathcal{D}_{\mathbf{V}}(\mathbf{U}_i, \mathbf{V})$ and $\mathcal{D}_{\mathbf{V}}(\mathbf{U}_j, \mathbf{V})$ should also be minimized. We thus further extend Eq. 4.7 into Eq. 4.8 to measure the disagreement among $\mathbf{U}_i$, $\mathbf{U}_j$ and $\mathbf{V}$:

$$\begin{aligned} \mathcal{Q}_{ij} &= \frac{1}{2} \{ \mathcal{D}_{\mathbf{V}}(\mathbf{U}_i, \mathbf{U}_j) + \mathcal{D}_{\mathbf{V}}(\mathbf{U}_i, \mathbf{V}) + \mathcal{D}_{\mathbf{V}}(\mathbf{U}_j, \mathbf{V}) \} \\ &= \text{tr} \{ \mathbf{V}^T [\mathbf{I} - \text{sym}(\mathbf{U}_i \mathbf{U}_i^T \mathbf{U}_j \mathbf{U}_j^T)] \mathbf{V} \}, \end{aligned} \quad (4.8)$$

where $\text{sym}(\mathbf{A}) = (\mathbf{A} + \mathbf{A}^T)/2$ for any square matrix $\mathbf{A}$. To derive Eq. 4.8, we use the trace expansion of the Frobenius norm, as well as the linearity and cyclicity properties of matrix trace. Now Let

$$\mathcal{L}_{ij} = \mathbf{I} - \text{sym}(\mathbf{U}_i \mathbf{U}_i^T \mathbf{U}_j \mathbf{U}_j^T), \quad (4.9)$$

then Eq. 4.8 becomes

$$\mathcal{Q}_{ij} = \text{tr}(\mathbf{V}^T \mathcal{L}_{ij} \mathbf{V}). \quad (4.10)$$

In addition, according to Eq. 4.3, we also need to minimize the unary cost of spectral embedding in each feature type for $1 \leq i \leq M$:

$$\mathcal{Q}_{ii} = \text{tr}(\mathbf{U}_i^T \mathcal{L}_i \mathbf{U}_i), \quad (4.11)$$

where $\mathcal{L}_i$ denotes the normalized Laplacian matrix of a specific feature type; $\mathbf{U}_i$ corresponds to Laplacian embedding.

Therefore, $\forall 1 \leq i \leq j \leq M$, we need to minimize both the pairwise disagreement cost defined by Eq. 4.10, as well as the unary spectral embedding cost defined by Eq. 4.11: $\sum_{j=i}^M \sum_{i=1}^M \mathcal{Q}_{ij}$. However, as the pairwise costs $\{\mathcal{Q}_{ij}\}_{i < j}$ and the unary costs $\{\mathcal{Q}_{ii}\}$ have different properties, they cannot be simply fused using the same weight. Moreover,

even for the same type of costs, assigning equal weights may not be the optimal choice either, as a poor feature modality or two opposing feature modalities may introduce a larger cost of embedding or disagreement. Instead of assigning equal weights, we prefer to assign a larger penalty weight to Q_{ij} of higher cost, which enables us to concentrate more on minimizing Q_{ij} of higher cost, so that not only the overall cost can be reduced, but also the consensus can be reached by suppressing high values of individual cost Q_{ij} . To achieve our goals, we propose the following optimization problem:

$$\begin{aligned}
& \min_{\{\mathbf{U}_m\}_{m=1}^M, \mathbf{V}} \quad \max_{\{\alpha_{ij}\}_{j \geq i}^M} \sum_{j=i}^M \sum_{i=1}^M \alpha_{ij}^\gamma Q_{ij} \\
& \text{subject to} \quad \alpha_{ij} \in \mathbb{R}^+, \sum_{j=i}^M \sum_{i=1}^M \alpha_{ij} = 1, \\
& \quad \mathbf{U}_m \in \mathbb{R}^{N \times K}, \mathbf{U}_m^T \mathbf{U}_m = \mathbf{I}, \\
& \quad \mathbf{V} \in \mathbb{R}^{N \times K}, \mathbf{V}^T \mathbf{V} = \mathbf{I},
\end{aligned} \tag{4.12}$$

where $\gamma \in [0, 1)$ is a parameter to control the distribution of weights α_{ij}^γ . When $\gamma = 0$, it is a special case with equal weights.

This optimization in Eq. 4.12 aims to achieve multi-feature fusion via minimizing the maximum weighted disagreement costs. For one thing, maximizing the overall cost w.r.t. weight variables will highlight Q_{ij} of high costs, *i.e.*, large disagreement or high embedding cost. For another, minimizing the overall cost w.r.t. embeddings can further reduce the highlighted costs. Moreover, it is worth noting that our objective function has only one parameter γ . Instead of manually selecting weights α_{ij} for Q_{ij} , our objective function can optimize the fusion weights too.

4.2.4 Optimization

It is infeasible to simultaneously achieve the optimal values of all variables in Eq. 4.12, because they depend on each other. However, the objective function $\Omega = \sum_{j=i}^M \sum_{i=1}^M \alpha_{ij}^\gamma Q_{ij}$ is concave w.r.t. each of $\{\alpha_{ij}\}_{j \geq i}^M$, and is convex w.r.t. each of $\{\mathbf{U}_m\}_{m=1}^M$ and $\mathbf{V}$. Since Ω is differentiable, it is easy to reach local optimum of the objective w.r.t. one variable

---

Algorithm 2: MULTI-FEATURE SPECTRAL CLUSTERING WITH MINIMAX OPTIMIZATION

Input: data $\{\mathcal{X}_i\}_{i=1}^N$; M types of features $\{\mathcal{F}^{(m)}\}_{m=1}^M$; number of clusters K ;
 parameter γ
Output: data clustering assignment indexes $\mathcal{Y} \in \mathbb{R}^N$
 // Initialization
 1: $\alpha_{ij} \leftarrow 1 / \sum_{q=p}^M \sum_{p=1}^M 1, \forall 1 \leq i, j \leq M$
 2: **for** $i \in [1, M]$ **do**
 3:    $\mathcal{L}_{\text{reg},i} \leftarrow \mathcal{L}_i$ (Eq. 4.2)
 4:    $\mathbf{U}_i \leftarrow$ first K smallest eigenvectors of $\mathcal{L}_{\text{reg},i}$
 5: **end for**
 // Main loop
 6: **repeat**
 7:    $\mathcal{L}_{ij} \leftarrow \mathbf{I} - \text{sym}(\mathbf{U}_i \mathbf{U}_i^T \mathbf{U}_j \mathbf{U}_j^T), \forall 1 \leq i < j \leq M$ (Eq. 4.9)
 8:    $\mathcal{L}_{\mathbf{V}} \leftarrow \sum_{j=i+1}^M \sum_{i=1}^M \alpha_{ij}^{\gamma} \mathcal{L}_{ij}$ (Eq. 4.14)
 9:    $\mathbf{V} \leftarrow$ first K smallest eigenvectors of $\mathcal{L}_{\mathbf{V}}$
 10:    $\alpha_{ij}^{\gamma} \leftarrow Q_{ij}^{\frac{\gamma}{1-\gamma}} / (\sum_{q=p}^M \sum_{p=1}^M Q_{pq}^{\frac{1}{1-\gamma}})^{\gamma}, \forall 1 \leq i, j \leq M$ (Eq. 4.18)
 11:   **for** $i \in [1, M]$ **do**
 12:      $\mathcal{L}_{\text{reg},i} = \alpha_{ii}^{\gamma} \mathcal{L}_i - \sum_{j \neq i} \alpha_{ij}^{\gamma} \text{sym}(\mathbf{U}_j \mathbf{U}_j^T \mathbf{V} \mathbf{V}^T)$ (Eq. 4.16)
 13:      $\mathbf{U}_i \leftarrow$ first K smallest eigenvectors of $\mathcal{L}_{\text{reg},i}$
 14:   **end for**
 15: **until** Ω (Eq. 4.12) is not changed or maximum number of iterations is reached
 // Discrete solution
 16: **return** $\mathcal{Y} \leftarrow k$ -means clustering on rows of $\mathbf{V}$

---

when fixing others. Therefore, we propose to alternatively update $\mathbf{V}$, $\{\mathbf{U}_m\}_{m=1}^M$ and $\{\alpha_{ij}\}_{j \geq i}^M$. We will justify the convergence of our approach in Section 4.3.4 experimentally.

Initialization. We initialize each $\mathbf{U}_i$ using Laplacian embedding of the corresponding feature type, and assign equal weights to different costs.

Minimization: optimizing $\mathbf{V}$. To minimize the objective function Ω , it can be transformed to Eq. 4.13 using the linearity of matrix trace:

$$\Omega = \text{tr}(\mathbf{V}^T \mathcal{L}_{\mathbf{V}} \mathbf{V}) + \sum_{i=1}^M \alpha_{ii}^{\gamma} Q_{ii}, \quad (4.13)$$

where

$$\mathcal{L}_{\mathbf{V}} = \sum_{j=i+1}^M \sum_{i=1}^M \alpha_{ij}^{\gamma} \mathcal{L}_{ij}, \quad (4.14)$$

and only the first term is related to $\mathbf{V}$. Under the orthonormal constraints of $\mathbf{V}$ (Eq. 4.12), $\mathbf{V}$ can be updated by performing spectral embedding of $\mathcal{L}_{\mathbf{V}}$, *i.e.*, by seeking the first K smallest eigenvectors of $\mathcal{L}_{\mathbf{V}}$.

Minimization: optimizing $\mathbf{U}_i$. To minimize the objective function Ω , it can be transformed to Eq. 4.15 using the linearity and cyclicity of matrix trace, where we let $\alpha_{ij} = \alpha_{ji}, \forall j < i$:

$$\Omega = \text{tr}(\mathbf{U}_i^T \mathcal{L}_{\text{reg},i} \mathbf{U}_i) + \mathcal{C}_i, \quad (4.15)$$

where

$$\mathcal{L}_{\text{reg},i} = \alpha_{ii}^{\gamma} \mathcal{L}_i - \sum_{j \neq i} \alpha_{ij}^{\gamma} \text{sym}(\mathbf{U}_j \mathbf{U}_j^T \mathbf{V} \mathbf{V}^T), \quad (4.16)$$

and

$$\mathcal{C}_i = \sum_{j \geq h, j \neq i} \sum_{h \neq i} \alpha_{hj}^{\gamma} \mathcal{Q}_{hj} + K \sum_{j \neq i} \alpha_{ij}^{\gamma}. \quad (4.17)$$

Since $\mathcal{C}_i$ is not related to $\mathbf{U}_i$, under the orthonormal constraints of $\mathbf{U}_i$ (Eq. 4.12), $\mathbf{U}_i$ can be updated by performing spectral embedding of $\mathcal{L}_{\text{reg},i}$, *i.e.*, by seeking the first K smallest eigenvectors.

Maximization: optimizing α_{ij} . It becomes a maximization problem w.r.t. α_{ij} . Applying the Lagrange multiplier method, we can obtain the closed-form of α_{ij}^{γ} as

$$\alpha_{ij}^{\gamma} = \frac{\mathcal{Q}_{ij}^{\frac{\gamma}{1-\gamma}}}{\left(\sum_{q=p}^M \sum_{p=1}^M \mathcal{Q}_{pq}^{\frac{1}{1-\gamma}} \right)^{\gamma}}. \quad (4.18)$$

Because $\gamma \in [0, 1)$, Eq. 4.18 shows that larger costs will be assigned with larger weights. As a result, larger disagreements will be suppressed across heterogeneous features in the process of total cost minimization. Further analyzing Eq. 4.18, we can see that, when $\gamma \rightarrow 0$, different weights will come close to each other; when $\gamma \rightarrow 1$, the weight

of the largest cost will tend to be 1, while other weights will approach 0; $0 < \gamma < 1$ achieves a trade-off weighting. We will also discuss the influence of parameter γ in the experiments.

We show our complete solution in Algorithm 2. As can be seen, the computational complexity within each iteration of our method mainly relies on $M + 1$ times of eigen-decomposition (Lines 9 and 13), which can be solved in time $O(N^3)$ [123]. To obtain further speed up for large scale applications, we can consider to apply some state-of-the-art efficient approximate approaches such as supernodes method [124] or bilateral random projections [125].

4.3 Experiments

4.3.1 Datasets and Experimental Setting

To evaluate our multi-feature clustering, we conduct experiments on three image datasets: UCI Digits [126], Oxford Flowers [127], UC Merced Land Uses [128], and a video dataset: Body Motions [129].

UCI Digits. We integrate multiple feature types for handwritten digit recognition on the UCI Digit Dataset, which consists of features of handwritten numerals ('0'–'9') extracted from 2,000 Dutch utility maps [126]. Each digit category contains 200 samples. These digits are represented by six types of features: (1) Fourier coefficients of the character shapes (FOU); (2) profile correlations (FAC); (3) Karhunen-Loeve coefficients (KAR); (3) pixel averages in 2×3 windows (PIX); (5) Zernike moments (ZER); and (6) morphological features (MOR). As the source image dataset is not available [126], we show the sampled images by the 240-dimensional pixel features in Figure 4.1.

Oxford Flowers. The Oxford Flower Dataset is composed of 17 flower categories, with 80 images for each category [127]. Each image is described by different visual features

Feature Type	Body Motions		Oxford Flowers		UC Merced Land Uses		UCI Digits	
	Feature	Dimension	Feature	Dimension	Feature	Dimension	Feature	Dimension
1	HOG	4000	Color	500	LLC 1×1	1024	FOU	76
2	MBH	4000	Shape	1000	LLC 2×2	4096	FAC	216
3	—	—	Texture	700	LLC 4×4	16384	KAR	64
4	—	—	—	—	pHOG	680	PIX	240
5	—	—	—	—	GIST	512	ZER	47
6	—	—	—	—	Color Histograms	784	MOR	6

TABLE 4.1. The image/video datasets used in our experiment and their feature descriptors.

FIGURE 4.1. Visualization of some sampled images of UCI Handwritten Digits. Each row shows 30 images from the same class of digits.

FIGURE 4.2. Sample images from Oxford 17-Category Flower Dataset. Five images are shown for each category. Each category contains instances of pose variations, scale changes, illumination variations, large intra-class variations and self-occlusion.

using color, shape, and texture. We show 5 representative flowers for each class in Figure 4.2.

UC Merced Land Uses. The UC Merced Land Use dataset [128] contains 21 classes of aerial orthoimagery, with 100 images each category. For local visual features, we represent each image as three pools of LLCs (locality-constrained linear codes) over dense SIFTs with 1×1 , 2×2 , and 4×4 partitions [130]. For global visual features, we extract pHOG [118], GIST [131] and Color Histograms [132]. We show 5 sample

FIGURE 4.3. Sample images from UC Merced 21-Category Land Use Dataset. Five samples from each category are shown above.

images for each class in Figure 4.3.

Body Motions. Appearance and motion features complement each other for body motion description and recognition in video data. Therefore, we combine such two feature types for action clustering. The human body motion dataset, which is introduced by UCF101 [129], contains 16 categories of human body actions and 1910 videos in total. For appearance features, each video is described by dense appearance trajectories based on Histogram of Oriented Gradients (HOG); while for motion features, each video is represented as dense motion trajectories based on Motion Boundary Histograms (MBH) [133]. For each category, one sample action is shown in Figure 4.4.

We summarize the feature descriptors in Table. 4.1 for the four datasets, including feature type IDs and feature dimensions. In all experiments, we compute pairwise image similarities using Gaussian kernel (Eq. 5.1). For UCI Digits [126], UC Merced Land Uses [128] and Body Motions [129], we set the bandwidth parameter σ to the median of the pair-wise Euclidean distances of corresponding feature descriptors. For Oxford Flowers [127], we use a fixed bandwidth $\sigma = 0.851$ as well as the online χ^2 distance

FIGURE 4.4. Sample videos from Human Body Motion Dataset. One sample from each category is shown above.

matrices of individual visual features¹ to compute the similarity matrices.

Except for the parameter sensitivity experiment, we fix the parameter γ to 0.33 for all experiments. As adopted in [42, 43, 70], we evaluate clustering performance using two standard measures: Clustering accuracy and normalized mutual information (NMI) from 10 random runs. Generally, the higher the measures, the better the performance.

4.3.2 Baseline Algorithms

To validate the performance of the proposed multi-feature clustering approach, we compare it with various baselines:

- **Single Feature Type Spectral Clustering (SC(#))**: running spectral clustering [121] with graph Laplacian derived from a single feature type.
- **Kernel Averaging Spectral Clustering (KASC)**: averaging normalized kernel matrix derived from individual feature types, followed by applying spectral clustering [121]

¹<http://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html>

Method	Body Motions		Oxford Flowers		UC Merced Land Uses		UCI digits	
	Accuracy	NMI	Accuracy	NMI	Accuracy	NMI	Accuracy	NMI
SC (1)	0.273 $\pm$ 0.009	0.263 $\pm$ 0.005	0.343 $\pm$ 0.013	0.371 $\pm$ 0.009	0.381 $\pm$ 0.011	0.459 $\pm$ 0.012	0.679 $\pm$ 0.044	0.649 $\pm$ 0.015
SC (2)	0.312 $\pm$ 0.010	0.342 $\pm$ 0.004	0.404 $\pm$ 0.016	0.425 $\pm$ 0.007	0.364 $\pm$ 0.020	0.440 $\pm$ 0.017	0.631 $\pm$ 0.048	0.622 $\pm$ 0.016
SC (3)	-	-	0.257 $\pm$ 0.010	0.239 $\pm$ 0.009	0.387 $\pm$ 0.010	0.449 $\pm$ 0.006	0.692 $\pm$ 0.087	0.652 $\pm$ 0.042
SC (4)	-	-	-	-	0.093 $\pm$ 0.004	0.242 $\pm$ 0.003	0.710 $\pm$ 0.053	0.660 $\pm$ 0.027
SC (5)	-	-	-	-	0.336 $\pm$ 0.007	0.397 $\pm$ 0.005	0.569 $\pm$ 0.021	0.500 $\pm$ 0.009
SC (6)	-	-	-	-	0.234 $\pm$ 0.006	0.269 $\pm$ 0.006	0.420 $\pm$ 0.028	0.469 $\pm$ 0.008
KASC	0.301 $\pm$ 0.013	0.328 $\pm$ 0.009	0.370 $\pm$ 0.012	0.403 $\pm$ 0.009	0.294 $\pm$ 0.010	0.349 $\pm$ 0.009	0.709 $\pm$ 0.042	0.668 $\pm$ 0.016
PRSC [43]	0.275 $\pm$ 0.007	0.263 $\pm$ 0.005	0.419 $\pm$ 0.013	0.435 $\pm$ 0.008	0.368 $\pm$ 0.022	0.447 $\pm$ 0.010	0.769 $\pm$ 0.049	0.728 $\pm$ 0.020
CRSC [43]	0.317 $\pm$ 0.020	0.335 $\pm$ 0.012	0.449 $\pm$ 0.019	0.461 $\pm$ 0.009	0.395 $\pm$ 0.011	0.468 $\pm$ 0.006	0.770 $\pm$ 0.036	0.713 $\pm$ 0.011
MMSC [42]	0.266 $\pm$ 0.008	0.305 $\pm$ 0.010	0.416 $\pm$ 0.015	0.427 $\pm$ 0.012	0.123 $\pm$ 0.012	0.265 $\pm$ 0.007	0.731 $\pm$ 0.011	0.675 $\pm$ 0.008
AASC [70]	0.250 $\pm$ 0.011	0.292 $\pm$ 0.008	0.410 $\pm$ 0.028	0.422 $\pm$ 0.014	0.226 $\pm$ 0.009	0.291 $\pm$ 0.007	0.683 $\pm$ 0.047	0.649 $\pm$ 0.018
Ours	0.322 $\pm$ 0.015	0.352 $\pm$ 0.010	0.493 $\pm$ 0.039	0.484 $\pm$ 0.022	0.404 $\pm$ 0.021	0.482 $\pm$ 0.015	0.800 $\pm$ 0.102	0.785 $\pm$ 0.049

TABLE 4.2. Comparisons of various baselines with the proposed approach.

with corresponding Laplacian. The kernel normalization is obtained by $(ker)^{\frac{1}{dim}}$, where ker denotes a feature kernel matrix, and dim denotes the feature dimension.

- **Centroid Co-regularized Spectral Clustering (CRSC)**: pushing all spectral embeddings of different feature types close to a centroid embedding using data-data similarity matrices (Eq. 4.4) [43], followed by k -means with the centroid embedding. We set the parameters in this algorithm to be 0.01 as suggested.
- **Pairwise Co-regularized Spectral Clustering (PRSC)**: pushing pairwise spectral embeddings of different feature types close to each other using data-data similarity matrices (Eq. 4.4) [43], followed by k -means clustering with embedding concatenation. We set the parameter in this algorithm to be 0.01 as suggested .
- **Multi-Modal Spectral Clustering (MMSC)**: learning a shared graph Laplacian from different feature types [42], followed by NMF-based spectral clustering [134]. We report the best results by tuning the parameter of this algorithm in the range from 10^{-2} to 10^2 with incremental step $10^{0.2}$ as suggested.
- **Affinity Aggregation Spectral Clustering (AASC)**: aggregating affinities of different feature types with optimized weights [70], followed by applying normalized cut [17] with corresponding Laplacian.

4.3.3 Experimental Results

We compare our method with baseline methods in Table 4.2. For the experiment of Body Motion clustering, there are only two types of features: motion features (MBH) and appearance features (HOG). Spectral clustering results show that motion features perform better than appearance features in human body motion clustering. Although most compared algorithms perform better than spectral clustering on the poorer feature type, they are difficult to beat spectral clustering on the better feature type. For example, only CRSC performs better than the better feature type. In contrast, thanks to our new formulation for multi-feature fusion in Eq. 4.12, our approach can enhance the poorer

FIGURE 4.5. Iteration comparisons of PRSC, CRSC and our method w.r.t. NMI performance of each feature embedding and the fusion result on the body motion and Oxford flower datasets. Best viewed in color.

feature embedding with iterations (Algorithm 2), thus generating a good clustering result. As can be seen in Table 4.2, our approach is not only better than the compared algorithms, but also can compete against the result of the better feature type.

Similarly, our approach also outperforms all compared methods on the other benchmark datasets, which further verifies the effectiveness of the proposed multi-feature clustering. Regarding flower clustering, all the compared methods perform better than spectral clustering on a single feature type. However, our approach can achieve much higher accuracy and NMI values than the other methods. For scene clustering, spectral clustering performs poorly on pHOG and color histograms. Despite such poor features, our approach still achieves a noticeable performance gain thanks to heterogeneous feature fusion. In handwritten digit grouping, since each single feature type reaches a good spectral clustering, the compared multi-feature clustering methods all obtain commendable results. Again, our approach achieves the most significant improvement than other compared methods. All the results obtained with our approach can benefit from effectively unveiling and fusing complementary information from heterogeneous features by optimizing Eq. 4.12.

To further illustrate the advantage of our approach, we study how different regularization methods influence the performance of each feature embedding, as well as the performance of the fusion result. As shown in Figure 4.5, we compare the most related work PRSC and CRSC with our approach on the body motion and Oxford flower datasets.

PRSC adopts pairwise regularization among different modality-specific Laplacian embeddings using data-data similarity matrices. From Figure 4.5 (a) and (d), we can see that, PRSC is sensitive to the poorer feature types, *e.g.*, HOG features in body motion dataset and color/texture features in Oxford flower dataset. In the initialization, *i.e.*, the first iteration, the fusion result approaches/exceeds the performance of the best feature type. However, with more iterations, the regularization may lead to a worse result. In such a case, it is not a good choice for multi-feature clustering. Similarly, CRSC also

FIGURE 4.6. Convergence study of our algorithm: NMI and objective values with iterations.

FIGURE 4.7. Parameter study of our algorithm: NMI values with different parameters.

leverages data-data similarity matrices to perform regularization. But it aims to force each modality-specific Laplacian embedding towards a consensus embedding. From Figure 4.5 (b) and (e), we can see that, CRSC is not very sensitive to poor feature types. However, it cannot effectively enhance the poor feature types. As shown in Figure 4.5 (e), successive regularization does not improve the bad performances of color and texture features. In such a case, CRSC is unable to bring different modality-specific Laplacian embeddings close enough, which finally influences the consensus result. On the contrary, we do not directly use data-data similarity matrices for regularization. By projecting each data-data similarity matrix onto the universal feature embedding, we relax the data-data similarity matrix from each feature type to obtain the regularized data-cluster similarity matrix (using Eq. 4.5) for our regularization framework. As shown in Figure 4.5 (c) and (f), although our method may not perform better than the best individual feature initially (Figure 4.5 (c)), the regularization can gradually reduce the disagreements among different feature embeddings to refine the performance, and finally enhance the fusion result.

4.3.4 Convergence Analysis

To evaluate the convergence of the proposed solution in Algorithm 2, Figure 4.6 shows the objective function value together with its NMI performance indicator over iterations, with $\gamma = 0.33$. The dashed line shows the objective values, while the solid line shows the NMI values. As can be seen, the objective function first moves down then upwards and flattens. After less than 20 iterations, the algorithm will converge to a saddle, which meets minimax optimization of the objective function in Eq. 4.12. Besides, it is interesting to notice that, although the objective function value has almost no change after several iterations, the performance can still benefit from the min-max iteration, *e.g.*, the results shown in Figure 4.6 (b) and (e). This further verifies the effectiveness of our proposed algorithm. This proposed min-max optimization cannot theoretically guarantee convergence. We notice that even if we cannot obtain a converged solution, the objective values of the alternating optimization can still be bounded in an advisable interval. Thus we can terminate the algorithm if such a situation happens.

4.3.5 Sensitivity of Parameters

In our objective function in Eq. 4.12, we have only one parameter γ to control the weights. As another factor, our proposed method relies on the input order of different features as shown in lines 10-13 of Algorithm 2. The sensitivity of the related factors will be studied in this section.

Parameter γ . As formulated in the objective function of Eq. 4.12, our proposed approach only has one parameter γ to balance different cost terms. γ is in the interval $[0, 1)$. We plot NMI performance curve w.r.t. parameter γ in Figure 4.7. As can be seen, different values of γ do not influence much for appearance and motion feature fusion on the body motion dataset. The reason may be that these two types of features can regularize each other, and the weight assignments are not so critical in such a case. For each of the other datasets, the result are not very sensitive to γ .

Feature Order	Accuracy	NMI
1 2 3	0.493 ± 0.039	0.484 ± 0.022
3 2 1	0.481 ± 0.034	0.482 ± 0.017
3 1 2	0.498 ± 0.034	0.498 ± 0.023
2 3 1	0.458 ± 0.026	0.468 ± 0.013
2 1 3	0.458 ± 0.017	0.471 ± 0.009
1 3 2	0.506 ± 0.028	0.497 ± 0.019

TABLE 4.3. Performance of flower clustering with different feature input orders using our algorithm.

Feature Order. To study how the input order of feature types influences clustering performance, we test our algorithm using different feature input orders on the Oxford flower dataset. We enumerate all six possible permutations of feature types, as shown in the first column of Table 4.3. The corresponding feature IDs are given in Table 4.1. All the results (in Table 4.3) perform better than the baseline methods (as shown in Table 4.2), and the result is not sensitive to the feature input order.

4.4 Conclusion

Multi-feature clustering is a challenging problem as it is difficult to find a clustering result agreeable to all feature modalities. To find the consensus, we explore an loss function consisting of both the unary term based on the cost of the Laplacian embedding of each individual feature modality and the pairwise disagreement term between any pair of feature modalities. To optimize the objective function, we propose a minimax formulation by minimizing the maximum loss. Our multi-feature clustering approach has only one parameter and does not need to specify the fusion weights. Our multi-feature clustering results on four image and video datasets show superior performance when compared with the state-of-the-art methods.

Chapter 5

Collaborative Multi-feature Fusion for Transductive Spectral Learning

5.1 Introduction

As noted in Chapters 3 and 4, in many pattern classification problems, the target data, *e.g.*, an image, can be naturally represented using different types (modalities) of features, *e.g.*, color, shape, and texture features. In Chapters 3 and 4, we have studied how to combine multiple complementary features for a better clustering result. Furthermore, this chapter presents a novel semi-supervised method for visual classification and pattern discovery. With semi-supervised visual pattern discovery, we target at leveraging a small amount of labeled data to transfer class labels to unlabeled data in individual groups of visual patterns. Much previous work [44] has studied how to leverage multiple feature types to improve the classification performance, such as co-training [82, 104], canonical correlation analysis [80, 81], and multiple kernel learning [77–79]. Despite previous successes, most existing multi-feature learning approaches rely on the agreement among different feature types to improve the performance: the decision of a data sample is preferred to be consistent across different feature types. However, as different feature types may have different data characteristics

and distributions, a forced agreement among different feature types may not bring a satisfying result.

To handle the different data characteristics among multiple feature types, we propose to respect the data distribution and allow different feature types to have its own clustering results. This can faithfully reflect the data characteristics in different feature types, *e.g.*, color feature space can be categorized into a number of typical colors, while texture feature space categorized into a different number of texture patterns. To integrate the clustering results from different feature types, we represent each data sample by a co-occurrence of feature patterns, *e.g.*, a composition of typical color and texture patterns. Unlike much previous work on co-occurrence pattern discovery [32] in spatial domain, *e.g.*, [35, 36] and [37], we aim to capture co-occurrence patterns across multiple feature modalities. Such a treatment has two advantages. First, instead of forcing different feature types to agree with each other, we compose multiple feature types to reveal the compositional pattern across different feature types, and thus it can naturally combine multiple features. Comparing with a direct concatenation of multiple types of features, the feature co-occurrence patterns encode the latent compositional structure among multiple feature types, thus having a better representation power. Moreover, as it allows different clustering results in different feature types, the feature co-occurrence patterns can be more flexible. Second, the new feature co-occurrence representations of the data samples enable us to propagate labels in the same group of feature co-occurrence patterns. Our new feature co-occurrence representation does not need to optimize individual feature type iteratively like in co-training, thus is less sensitive to noisy feature types.

We study the collaborative multi-feature fusion in a transductive learning framework, where the labeled data can transfer the labels to the unlabeled data. To enable transductive spectral learning, we formulate a new objective function with three objectives, namely the good quality of spectral clustering in individual feature types, the label smoothness of data samples in terms of their feature co-occurrence representations, and

FIGURE 5.1. Label propagation of unlabeled data by the discovery of the co-occurrence patterns among different types of clusters. See text for details and best viewed in color.

the fitness to the labels provided by the training data. The optimization of this objective function is complicated as the spectral clustering results in different feature types and the formed co-occurrence patterns influence each other under the transductive learning formulation. We thus propose an iterative optimization approach that can decouple these factors. During the iterations, the clustering results of individual feature types and the smoothness of the labeling of data samples will help each other, leading to a better transductive learning. To evaluate our method, we conduct experiments on a synthetic

dataset, as well as object and action recognition datasets. The comparison with related methods such as [5],[135] and [4] show promising results that our proposed method can well handle the different data characteristics of multiple feature types and is robust to noisy feature types.

We explain our proposed transduction spectral learning using multi-feature fusion in Figure 5.1. There are four data classes represented by two feature modalities, *i.e.*, texture and color. The texture modality forms two texture patterns, chessboard and brick; while the color modality forms two color patterns, green and blue. All data samples belong to one of the four compositional patterns: green brick (Hexagon), blue chessboard (Triangle), green chessboard (Square), and blue brick (Circle). Clearly, the four data classes cannot be distinguished in either the texture or the color feature space alone. For example, the two classes Square and Triangle share the same texture attribute, but different in color, while the Hexagon and Square classes share the same color but different in texture. However, each class can be easily distinguished by a co-occurrence of the texture and color pattern, *e.g.*, the Hexagon class composes “brick” texture and “green” color. As a result, the unlabeled data samples of the same co-occurrence feature pattern can be labeled as the same class as the labeled data sample.

5.2 Proposed Method

We study the collaborative multi-feature fusion in a transductive learning framework, where the labeled data can transfer the labels to the unlabeled data. Consider a collection of partially labeled multi-class dataset $\mathcal{X} = (\mathcal{X}_l, \mathcal{X}_u)$. The labeled inputs $\mathcal{X}_l = \{x_i\}_{i=1}^l$ are associated with known labels $\mathcal{Y}_l = \{y_i\}_{i=1}^l$, where $y_i \in \mathcal{L} = \{1, 2, \dots, M\}$. The unlabeled data $\mathcal{X}_u = \{x_i\}_{i=l+1}^N$ are with missing labels $\mathcal{Y}_u = \{y_i\}_{i=l+1}^N$, where $y_i \in \mathcal{L}$ and the task is to infer $\mathcal{Y}_u$. A binary matrix $\mathbf{Y} \in \{1, 0\}^{N \times M}$ encodes the label information of $\mathcal{X}$, where $\mathbf{Y}_{ij} = 1$ if x_i has a label $y_i = j$ and $\mathbf{Y}_{ij} = 0$ otherwise. We set $\mathbf{Y}_{ij} = 0$ initially for unlabeled data $y_i \in \mathcal{Y}_u$. We assume each $x_i \in \mathcal{X}$ is represented as

K types/modalities of features as $\{\mathbf{f}_i^{(k)}\}_{k=1}^K$, where $\mathbf{f}_i^{(k)} \in \mathbb{R}^{d_k}$. To enable multi-feature collaboration in label propagation, we propose our methods in the following.

5.2.1 Spectral Embedding of Multi-feature Data

To handle the different data characteristics among multiple feature types, we propose to respect the data distribution and allow different feature types to have its own clustering results. As spectral embedding can effectively capture the data clustering structure [136], we leverage it to study the data distribution in each feature type.

At first, each feature type $\{\mathcal{F}^{(k)}\} = \{\mathbf{f}_i^{(k)}\}_{i=1}^N$ of $\mathcal{X}$ defines an undirected graph $G_k = (\mathcal{X}, \mathcal{E}, \mathbf{W}_k)$ in which the set of vertices is $\mathcal{X}$ and the set of edges connecting pairs of vertices is $\mathcal{E} = \{e_{ij}\}$. Each edge e_{ij} is assigned a weight $w_{ij}^{(k)} = \kappa(x_i, x_j)$ to represent the similarity between x_i and x_j . The matrix $\mathbf{W}_k = (w_{ij}^{(k)}) \in \mathbb{R}^{N \times N}$ denote the similarity or kernel matrix of $\mathcal{X}$ in this feature type. Following spectral clustering, we use the following function to compute the graph similarities:

$$w_{ij} = \exp\left\{-\frac{\text{dist}^2(\mathbf{f}_i^{(k)}, \mathbf{f}_j^{(k)})}{2\sigma^2}\right\}, \quad (5.1)$$

where $\text{dist}(\mathbf{f}_i^{(k)}, \mathbf{f}_j^{(k)})$ denotes the distance between a pair of features; σ is the bandwidth parameter to control how fast the similarity decreases. By summing the weights of edges being connected to x_i , we can obtain the degree of this vertex $d_i^{(k)} = \sum_{j=1}^N w_{ij}^{(k)}$. Let $\mathbf{D}_k \in \mathbb{R}^{N \times N}$ be the vertex degree matrix by placing $\{d_i^{(k)}\}_{i=1}^N$ on the diagonal. Then we can write the graph Laplacian $\Delta_k \in \mathbb{R}^{N \times N}$ as

$$\Delta_k = \mathbf{D}_k - \mathbf{W}_k \quad (5.2)$$

and the normalized graph Laplacian $\mathbf{L}_k \in \mathbb{R}^{N \times N}$ as $\mathbf{L}_k = \mathbf{D}_k^{-1/2} \Delta_k \mathbf{D}_k^{-1/2} = \mathbf{I}_N - \mathbf{D}_k^{-1/2} \mathbf{W}_k \mathbf{D}_k^{-1/2}$, where $\mathbf{I}_N$ is an identify matrix of order N .

After the above preprocessing to each feature type, we perform spectral clustering to group the feature points of both labeled and unlabeled data into clusters. Assume there are M_k clusters in the k_{th} feature type. The spectral clustering on this feature type is to minimize the spectral embedding cost [137]:

$$\Omega_{\text{type}}(\mathbf{R}_k) = \text{tr}(\mathbf{R}_k^T \mathbf{L}_k \mathbf{R}_k), \quad (5.3)$$

subject to $\mathbf{R}_k^T \mathbf{R}_k = \mathbf{I}_{M_k}$, where $\text{tr}(\cdot)$ denotes the matrix trace; $\mathbf{R}_k \in \mathbb{R}^{N \times M_k}$ is the real-valued cluster indicators of the M_k clusters [136]; $\mathbf{I}_{M_k}$ is an identify matrix of order M_k . By using the Rayleigh-Ritz theorem [122], we can obtain the solution of $\mathbf{R}_k$, which consists of the first M_k eigenvectors corresponding to the M_k smallest eigenvalues of $\mathbf{L}_k$, *i.e.*, $\mathbf{r}_i^{(k)}, i = 1, 2, \dots, M_k$, denoting as:

$$\mathbf{R}_k = [\mathbf{r}_1^{(k)}, \mathbf{r}_2^{(k)}, \dots, \mathbf{r}_{M_k}^{(k)}] \triangleq \text{eig}(\mathbf{L}_k, M_k). \quad (5.4)$$

By using Eq. 5.4, we can independently perform spectral embedding in different feature types. In other words, we do not have to force the clustering in different feature spaces to agree with each other.

5.2.2 Building Feature Co-occurrence Patterns and Multi-feature Similarity Graph

We have obtained K label indicator matrices $\{\mathbf{R}_k\}_{k=1}^K$ obtained from the K types of features by Eq. 5.4 in the above section. To integrate them, we build a matrix $\mathbf{T}_v \in \mathbb{R}^{\sum_{k=1}^K M_k \times N}$ as:

$$\mathbf{T}_v = [\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_K]^T. \quad (5.5)$$

The n_{th} column of $\mathbf{T}_v$ is the multi-feature representation of x_n , which conveys the complementary information across multiple types of feature clusters without forcing

clustering agreement among different feature types. Additionally, $\mathbf{T}_v$ stores soft feature co-occurrence patterns since $\{\mathbf{R}_k\}_{k=1}^K$ are soft cluster indicators of multiple feature types. Comparing to hard clustering indicators used in [97], the spectral soft relaxation can more effectively capture the feature clustering structures of individual feature types, and tolerate noisy features [136].

With the multi-feature representations of the samples in $\mathcal{X}$, *i.e.*, the feature co-occurrence patterns $\mathbf{T}_v$ in Eq. 5.5, we introduce the multi-feature similarity graph $G_v = (\mathcal{X}, \mathcal{E}, \mathbf{W}_v)$ based on $\mathbf{T}_v$. By Laplacian embedding, the resulting soft cluster indicators $\{\mathbf{R}_k\}_{k=1}^K$ can be considered to obey linear similarities [43], so are the concatenation of them, *i.e.*, $\mathbf{T}_v$. Therefore we define the similarity matrix $\mathbf{W}_v \in \mathbb{R}^{N \times N}$ as a linear kernel:

$$\mathbf{W}_v = \mathbf{T}_v^T \mathbf{T}_v = \sum_{k=1}^K \mathbf{R}_k \mathbf{R}_k^T. \quad (5.6)$$

Regarding the weighting coefficients, $\mathbf{W}_v$ can be considered as an average of the linear kernels of the soft cluster indicators in multiple feature types. Therefore it will be less sensitive to poor individual feature types. What needs to be noted is that although the entries of the matrix $\mathbf{W}_v$ are not necessary all non-negative, $\mathbf{W}_v$ is semi-positive. One can also add $\mathbf{W}_v$ with a rank-1 matrix whose entries are all equal to the minimum negative entry of $\mathbf{W}_v$ to make sure each entry of $\mathbf{W}_v$ non-negative. We will omit this manipulation in the following statement and derivation as it does not affect the solution (Section 5.2.4) to the problem Eq. 5.9.

According to the similarity matrix, we can obtain the degree matrix $\mathbf{D}_v \in \mathbb{R}^{N \times N}$ by

$$\mathbf{D}_v = \text{diag}(\mathbf{W}_v \mathbf{1}), \quad (5.7)$$

where $\mathbf{1} \in \mathbb{R}^N$ is an all one vector. We define the normalized Laplacian as:

$$\mathbf{L}_v = \mathbf{I}_N - \mathbf{D}_v^{-1/2} \mathbf{W}_v \mathbf{D}_v^{-1/2}. \quad (5.8)$$

With $\mathbf{L}_v$, we encode the smoothness of the multi-feature similarity graph. It will help us to assign the same label to data samples of similar feature co-occurrence patterns.

5.2.3 Multi-feature Fusion with Transductive Learning

After we construct the multi-feature similarity graph G_v by the co-occurrence patterns based on the feature clusters of multiple feature types, it is still a non-trivial task to build a smooth connection between the feature clustering structures of multiple feature types and the label predictions of unlabeled data. In order to address the problem, we introduce a soft class label matrix $\mathbf{R}_v \in \mathbb{R}^{N \times M}$ to assist the transition. Different from the hard class labels $\mathbf{Y} \in \{0, 1\}^{N \times M}$, $\mathbf{R}_v$ is a relaxed real matrix. All taken into account, we propose to minimize the spectral clustering costs of individual feature types, the labeling smoothness regularization of unlabeled data samples, and the fitting penalty of hard class labels $\mathbf{Y}$ and soft class labels $\mathbf{R}_v$ together in the following objective function:

$$\begin{aligned}
& \Omega(\{\mathbf{R}_i\}_{i=1}^K, \mathbf{R}_v, \mathbf{Y}) \\
&= \sum_{i=1}^K \Omega_{\text{type}}(\mathbf{R}_i) + \alpha \Omega_{\text{smooth}}(\mathbf{R}_v, \{\mathbf{R}_j\}_{j=1}^K) \\
&\quad + \beta \Omega_{\text{fit}}(\mathbf{R}_v, \mathbf{Y}) \\
&= \sum_{i=1}^K \text{tr}(\mathbf{R}_i^T \mathbf{L}_i \mathbf{R}_i) + \alpha \text{tr}(\mathbf{R}_v^T \mathbf{L}_v \mathbf{R}_v) \\
&\quad + \beta \text{tr}\{(\mathbf{R}_v - \mathbf{S}\mathbf{Y})^T (\mathbf{R}_v - \mathbf{S}\mathbf{Y})\},
\end{aligned} \tag{5.9}$$

subject to $\mathbf{R}_k^T \mathbf{R}_k = \mathbf{I}_{M_k}$, $\forall k = 1, 2, \dots, K$; $\mathbf{R}_v \in \mathbb{R}^{N \times M}$; $\mathbf{Y} \in \{1, 0\}^{N \times M}$ and $\sum_{j=1}^M \mathbf{Y}_{ij} = 1$ with balance parameters α and β . In our objective, $\mathbf{R}_k^T \mathbf{R}_k = \mathbf{I}_{M_k}$ is the requirement of unique embedding; $\sum_{j=1}^M \mathbf{Y}_{ij} = 1$ is to make a unique label assignment for each vertex; and $\mathbf{S} \in \mathbb{R}^{N \times N}$ is a normalized term to weaken the influence of noisy labels and balance class biases. Similar to [5], the diagonal elements of $\mathbf{S}$ are filled by the class-normalized node degrees: $\mathbf{s} = \sum_{j=1}^M \frac{\mathbf{Y}_{.j} \odot \mathbf{D}_v \mathbf{1}}{\mathbf{Y}_{.j}^T \mathbf{D}_v \mathbf{1}}$, where $\odot$ denotes Hadamard product; $\mathbf{Y}_{.j}$ denotes the j_{th} column of $\mathbf{Y}$; $\mathbf{1} \in \mathbb{R}^N$ is an all one vector.

More specifically, as discussed in Section 5.2.1, the spectral clustering objective of multiple feature types $\sum_{i=1}^K \Omega_{\text{type}}(\mathbf{R}_i)$ is to reveal the data distributions in multiple feature types without forcing clustering agreement. In addition, to allow the soft class labels $\mathbf{R}_v$ for $\mathcal{X}$ to be consistent on closely connected vertices in the multi-feature similarity graph G_v , we regularize our objective with the following smoothing function:

$$\Omega_{\text{smooth}}(\mathbf{R}_v, \{\mathbf{R}_j\}_{j=1}^K) = \text{tr}(\mathbf{R}_v^T \mathbf{L}_v \mathbf{R}_v), \quad (5.10)$$

where $\mathbf{L}_v$ is defined by Eq. 5.8 which is related to $\{\mathbf{R}_j\}_{j=1}^K$. Furthermore, to prevent overfitting, it should allow occasional disagreement between the soft class labels $\mathbf{R}_v$ and the hard class labels $\mathbf{Y}$ on the dataset $\mathcal{X}$. Thus, we minimize the fitting penalty:

$$\Omega_{\text{fit}}(\mathbf{R}_v, \mathbf{Y}) = \text{tr}\{(\mathbf{R}_v - \mathbf{S}\mathbf{Y})^T (\mathbf{R}_v - \mathbf{S}\mathbf{Y})\}. \quad (5.11)$$

Regarding our objective of Eq. 5.9, it is worth noting that the three terms of this function are correlated among each other. We thus cannot minimize Ω by minimizing the three terms separately. Moreover, the binary integer constraint on $\mathbf{Y}$ also challenges the optimization. We will in Section 5.2.4 show how to decouple the dependencies among them and propose our algorithm to solve this optimization function.

5.2.4 Optimization: Collaboration between Clustering and Classification

In this section we decouple the dependencies among the terms of Eq. 5.9 to solve the objective function. More specifically, we fix the soft feature clustering results $\{\mathbf{R}_k\}_{k=1}^K$ in individual feature types to optimize Ω over the class labeling results with soft class labels $\mathbf{R}_v$ and hard class labels $\mathbf{Y}$ together. And similarly, we fix the class labeling results with soft class labels $\mathbf{R}_v$ and hard class labels $\mathbf{Y}$ simultaneously to optimize Ω over the soft feature clustering results $\{\mathbf{R}_k\}_{k=1}^K$ in individual feature types. In the class

---

Algorithm 3: COLLABORATIVE MULTI-FEATURE FUSION FOR TRANSDUCTIVE SPECTRAL LEARNING

Input: labeled data $\{\mathcal{X}_l, \mathcal{Y}_l\}$; unlabeled data $\mathcal{X}_u$; K types of features $\{\mathcal{F}^{(k)}\}_{k=1}^K$; cluster numbers of individual feature types $\{M_k\}_{k=1}^K$; class number M ; parameters α and β

Output: labels on unlabeled data $\mathcal{Y}_u$

- 1: **Initialization:** initial label matrix $\mathbf{Y}$; normalized graph Laplacians of individual feature types $\mathbf{L}'_k \leftarrow \mathbf{L}_k, k = 1, 2 \cdots K$
  - 2: **repeat**
    - // Spectral embedding
    - 3: $\mathbf{R}_k \leftarrow \text{eig}(\mathbf{L}'_k, M_k), k = 1, 2 \cdots K$ (Eq. 5.4)
    - // Generate feature co-occurrence patterns
    - 4: $\mathbf{T}_v = [\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_K]^T$ (Eq. 5.5)
    - // Build multi-feature similarity graph Laplacian
    - 5: $\mathbf{W}_v \leftarrow \mathbf{T}_v^T \mathbf{T}_v$ (Eq. 5.6)
    - 6: $\mathbf{L}_v \leftarrow \mathbf{I}_N - \mathbf{D}_v^{-1/2} \mathbf{W}_v \mathbf{D}_v^{-1/2}$ (Eq. 5.8)
    - // Compute gradient w.r.t. class-normalized labels
    - 7: $\nabla_{(\mathbf{S}\mathbf{Y})} \Omega \leftarrow 2[\alpha \mathbf{P} \mathbf{L}_v \mathbf{P} + \beta (\mathbf{P} - \mathbf{I}_N)^2] \mathbf{S}\mathbf{Y}$ (Eq. 5.15)
    - // Reset unlabeled data
    - 8: $\mathcal{X}'_u \leftarrow \mathcal{X}_u$
    - // Gradient search for unlabeled data labeling
    - 9: **repeat**
      - 10: $(\tilde{i}, \tilde{j}) \leftarrow \arg \min_{(i,j): x_i \in \mathcal{X}_u, j \in \{1,2,\dots,M\}} \nabla_{(\mathbf{S}\mathbf{Y})} \Omega;$
      - 11: $\mathbf{Y}_{\tilde{i}, \tilde{j}} \leftarrow 1;$
      - 12: $y_{\tilde{i}} \leftarrow \tilde{j};$
      - 13: **until** $\mathcal{X}'_u \leftarrow \mathcal{X}'_u \setminus \tilde{x}_i = \emptyset$
      - // Update soft class labels of unlabeled data
      - 14: $\mathbf{R}_v \leftarrow \mathbf{P} \mathbf{S}\mathbf{Y}$ (Eq. 5.13)
      - // Regularize graph Laplacians for each feature types
      - 15: $\mathbf{L}'_k \leftarrow \mathbf{L}_k - \alpha \sum_{k=1}^K \mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_v \mathbf{R}_v^T \mathbf{D}_v^{-\frac{1}{2}}, k = 1, 2 \cdots K$ (Eq. 5.18)
      - 16: **until** Ω is not decreasing
-

labeling update step, we solve $\mathbf{R}_v$ by an analytical form, and then optimize Ω over $\mathbf{Y}$ using a gradient based greedy search approach. In the feature clustering update step, we optimize Ω over $\mathbf{R}_k, k = 1, 2, \dots, K$ separately.

The closed form of $\mathbf{R}_v$. Since Ω is quadratic w.r.t. $\mathbf{R}_v$, similar to [5], we are allowed to zero the partial derivative to obtain the analytical solution of $\mathbf{R}_v$ w.r.t. $\mathbf{Y}$ and $\{\mathbf{R}_k\}_{k=1}^K$.

We then have:

$$\frac{\partial \Omega}{\partial \mathbf{R}_v} = \alpha \mathbf{L}_v \mathbf{R}_v + \beta (\mathbf{R}_v - \mathbf{S}\mathbf{Y}) = 0, \quad (5.12)$$

which implies

$$\mathbf{R}_v = \left(\frac{\alpha}{\beta}\mathbf{L}_v + \mathbf{I}_N\right)^{-1}\mathbf{S}\mathbf{Y} = \mathbf{P}\mathbf{S}\mathbf{Y}, \quad (5.13)$$

where $\mathbf{P} = \left(\frac{\alpha}{\beta}\mathbf{L}_v + \mathbf{I}_N\right)^{-1}$, which is related to $\{\mathbf{R}_k\}_{k=1}^K$ according to Eq 5.8.

The soft class labels $\mathbf{R}_v$ make the transition smooth from feature clustering results of multiple feature types $\{\mathbf{R}_k\}_{k=1}^K$ to the prediction of hard class labels $\mathbf{Y}$ for the dataset $\mathcal{X}$. Then we can substitute the analytical solution of $\mathbf{R}_v$ in Eq. 5.13 to Eq. 5.9, and optimize Ω over $\mathbf{Y}$.

Optimize Ω over $\mathbf{Y}$. Given $\{\mathbf{R}_k\}_{k=1}^K$, we use the gradient based greedy search approach [5] to optimize the binary integer optimization. It is worth noting that searching along the gradient of hard class labels $\mathbf{Y}$ and class-normalized labels $\mathbf{S}\mathbf{Y}$ is in fact equivalent. Therefore,

$$\mathbf{Y}^{\text{update}}(\{\mathbf{R}_k\}_{k=1}^K) = \arg \min_{\mathbf{Y}} \nabla_{\mathbf{Y}} \Omega = \arg \min_{\mathbf{Y}} \nabla_{(\mathbf{S}\mathbf{Y})} \Omega, \quad (5.14)$$

where the gradient of Ω over $\mathbf{S}\mathbf{Y}$ is:

$$\nabla_{(\mathbf{S}\mathbf{Y})} \Omega = 2[\alpha\mathbf{P}\mathbf{L}_v\mathbf{P} + \beta(\mathbf{P} - \mathbf{I}_N)^2]\mathbf{S}\mathbf{Y}. \quad (5.15)$$

Eq. 5.14 shows how to leverage the feature clustering structures in multiple types of features $\{\mathbf{R}_k\}_{k=1}^K$ and the labeled data to predict the labels of unlabeled data.

Optimize Ω over $\mathbf{R}_k$, $\forall k = 1, 2, \dots, K$. We propose to update data clustering results by data class labeling results, which have not been studied before to the best of our knowledge. To this end, we fix $\{\mathbf{R}_i\}_{i \neq k}$, $\mathbf{R}_v$ and $\mathbf{Y}$, and obtain an equivalent minimization function J to minimize Ω (Eq. 5.9), where¹

$$\begin{aligned} J(\mathbf{R}_k, \mathbf{R}_v, \mathbf{Y}, \{\mathbf{R}_i\}_{i \neq k}) \\ = \sum_{i=1}^K \text{tr}\{\mathbf{R}_i^T (\mathbf{L}_i - \alpha \mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_v \mathbf{R}_v^T \mathbf{D}_v^{-\frac{1}{2}}) \mathbf{R}_i\}, \end{aligned} \quad (5.16)$$

¹The detailed derivation is shown in Appendix.

subject to $\mathbf{R}_k^T \mathbf{R}_k = \mathbf{I}_{M_k}$. However, the partial derivative of $\mathbf{D}_v$ w.r.t. $\mathbf{R}_k$ is intractable since there is a diagonalization operation in Eq. 5.7. We therefore use the values of $\{\mathbf{R}_i\}_{i=1}^K$ at the current iteration to initialize $\mathbf{D}_v$. Then given $\mathbf{D}_v$, the optimization turns out to minimize the following objective:

$$\begin{aligned} \Omega_{\text{type}}^{\text{new}}(\mathbf{R}_k, \mathbf{Y}, \{\mathbf{R}_j\}_{j \neq k}) \\ = \text{tr}\{\mathbf{R}_k^T (\mathbf{L}_k - \alpha \mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_v \mathbf{R}_v^T \mathbf{D}_v^{-\frac{1}{2}}) \mathbf{R}_k\}, \end{aligned} \quad (5.17)$$

subject to $\mathbf{R}_k^T \mathbf{R}_k = \mathbf{I}_{M_k}$. It becomes a spectral clustering with a regularized graph Laplacian:

$$\mathbf{L}_k^{\text{new}} = \mathbf{L}_k - \alpha \sum_{k=1}^K \mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_v \mathbf{R}_v^T \mathbf{D}_v^{-\frac{1}{2}}. \quad (5.18)$$

By using the Rayleigh-Ritz theorem [122], we can update $\mathbf{R}_k$ as the first M_k eigenvectors corresponding to the M_k smallest eigenvalues of $\mathbf{L}_k^{\text{new}}$:

$$\mathbf{R}_k^{\text{update}}(\mathbf{R}_k, \mathbf{Y}, \{\mathbf{R}_j\}_{j \neq k}) = \text{eig}(\mathbf{L}_k^{\text{new}}, M_k). \quad (5.19)$$

Eq. 5.19 shows how to tune the feature clustering result of each feature type $\mathbf{R}_k, \forall k = 1, 2, \dots, K$ by learning from the known data class labels and the feature clustering results of the other feature types. It is worth noting that, at the beginning, our method does not require the clustering agreement among different feature types. However, by further optimizing the objective, individual feature types will be regularized by known data class labels, and each individual feature type will be influenced by other feature types. In fact, the regularized graph Laplacian (Eq. 5.18) in each feature type has become a multi-feature Laplacian representation. Such multi-feature Laplacian representations should gradually agree with each other.

We also notice that the adjustment of $\mathbf{R}_k$ is related to its value in the previous iteration. This leads to a gradual change of $\mathbf{R}_k$. Strictly, because of this manipulation, it is difficult to theoretically compare the values of Eq. 5.17 before and after one iteration for convergence analysis. Nevertheless in our observation our method usually

converges in few steps. We show our complete solution in Algorithm 3. The complexity within each iteration of our method mainly relies on M times of eigen-decomposition ($O(N^3)$ [123]) and one update of soft class label matrix $\mathbf{R}_v$ and hard class matrix $\mathbf{Y}$ ($O(N^2)$ [138]). As mentioned in Chapter 4, we can further consider to apply supernodes method [124] or bilateral random projections [125] to speed up the computation of eigen-decomposition.

5.3 Experiments

5.3.1 Experiment Setting

In the experiments, the regularized parameters are both set to 1. Specifically, in our algorithm, we set $\alpha = 1$, and $\beta = 1$ as we observe they are not very sensitive. The observation on our extension of GTAM is consistent with GTAM in [5], which is also robust to the parameter setting. For a fair comparison, we set $C = 1$ in RWMV [4], and set $\mu = 1$ in GTAM [5]. As suggested in [4], the graph combination parameters in RWMV are set equally, *i.e.*, $\alpha_i = 1/M, i = 1, 2, \dots, K$. Besides, we use $\sigma = 0.3$ and Euclidean distance measure to build graph similarities for the simulation data (Section 5.3.2). In the real datasets, the bandwidth parameter σ equals to the median of the pairwise distances. We measure $\text{dist}^2(\cdot, \cdot)$ as χ^2 distance in the Oxford 17-Category Flower Dataset as provided in [127, 139] (Section 5.3.4). Euclidean distance measure is used in the UCI Handwritten Digit Dataset (Section 5.3.3), Human Body Motion Dataset (Section 5.3.5) and UC Merced Land Use Dataset (Section 5.3.6). Moreover, in each real dataset experiment, we randomly pick labeled samples and run 10 rounds for performance evaluation.

FIGURE 5.2. Classification on synthetic toy data with two feature types. Different markers, *i.e.*, “1”, “2”, “3” and “4”, indicate four different classes. Shading markers highlight the labeled data. The first column shows the synthetic toy data. The last three columns show the classifying results of RWMV [4], GTAM [5] and our proposed approach. Best viewed in color.

5.3.2 Synthetic Data

We synthesize a toy dataset with two types of features in Figure 5.2. Each type of features is described by a 2-dimensional feature space. The dataset has four classes labeled by “1”, “2”, “3” and “4”, respectively. The labeled data are highlighted using different colors. Each class has 200 samples. Feature type #1 has two clusters: Above moon and Below moon. Feature type #2 also has two clusters: Left moon and Right moon.

It is worth noting that the feature clusters are mixed across different classes. In feature type #1, both classes #1 and #2 share cluster A; and both classes #3 and #4 share cluster B. In feature type #2, both classes #2 and #4 share cluster L; and both classes #1 and #3 share cluster R. Therefore it is infeasible to classify the data by using a single feature type. In addition, a direct concatenation of features from multiple feature types will diminish the differences among samples, thus cannot distinguish all samples from different classes. For example, by using the GTAM [5], the concatenated features obtain 92.81% accuracy, but cannot disambiguate among several samples. In terms of general multi-feature fusion approaches, *e.g.*, RWMV [4], the requirement that the data categorization results in individual feature types should agree with each other does not hold, *e.g.*, the toy data. Hence the accuracy of RWMV just reaches 48.11%.

In contrast, by utilizing the feature co-occurrence patterns among multiple feature types, our approach can learn a favourable clustering, and the accuracy is 100%. Specifically, class #1 exhibits the co-occurrence of cluster A in feature type #1 and cluster R in feature type #2; class #2 exhibits the co-occurrence of cluster A in feature type #1 and cluster L in feature type #2; class #3 exhibits the co-occurrence of cluster B in feature type #1 and cluster R in feature type #2; and class #4 exhibits the co-occurrence of cluster B in feature type #1 and cluster L in feature type #2.

5.3.3 UCI Handwritten Digit Dataset

To evaluate how multiple feature types influence handwritten digit recognition, we test the multi-feature digit dataset [140] from the UCI Machine Learning Repository [126]. It consists of features of handwritten numerals ('0'–'9') extracted from a collection of Dutch utility maps. There are 200 samples in each class. So the data set has a total of 2,000 samples. These digits are represented by six types of features: (1) 76-dimensional Fourier coefficients of the character shapes (fou); (2) 64-dimensional Karhunen-Loeve coefficients (kar); (3) 240-dimensional pixel averages in 2×3 windows (pix); (4) 216-dimensional profile correlations (fac); (5) 47-dimensional Zernike moments (zer); and (6) 6-dimensional morphological features (mor). All features are concatenated to generate the 649-dimensional features. As the source image dataset is not available [126], we have shown the sampled images by the 240-dimensional pixel features in Figure 4.1.

In this experiment, the first 50 samples from each digit class are labeled for transductive learning. The classification results on the remaining 1500 unlabeled samples are used for evaluation. For each class, we randomly pick labeled data from the 50 labeled candidates and vary the size from 2 to 20. The accuracy comparison results are shown in Figure 5.3, including our approach, GTAM [5] (on the best single feature type, the worst single feature type and the concatenations of all feature types) and RWMV [4] (on all feature types).

The various performances of individual feature types show there is a substantial disagreement among feature types in this dataset. The concatenation of all the six feature types performs better than the worst single feature but worse than the best single feature when using GTAM. This also shows that feature concatenation can be easily affected by the bad feature types, thus not the best choice for multi-feature transductive learning. By a linear combination of similarity matrices of the six feature types [4], the performance of RWMV can be close to that of GTAM on the best single feature type, but is still affected by the poor feature types. The best performance is achieved by our approach, which benefits from learning the feature co-occurrence patterns. In Figure 5.3,

FIGURE 5.3. Performance comparison on UCI handwritten digits.

#cluster per feature type	Accuracy	#cluster per feature type	Accuracy
5	0.870 ± 0.012	50	0.970 ± 0.001
10	0.925 ± 0.002	100	0.966 ± 0.013
20	0.958 ± 0.001	200	0.936 ± 0.035

TABLE 5.1. Performance of our approach on UCI handwritten digits under different cluster numbers per feature type. The size of labeled data is 20.

we show the results of our approach with 100 clusters per feature type. Because we do not force individual feature types to have the same clustering structure, the feature co-occurrence patterns faithfully reflect the data distribution characteristics. Moreover, as discussed in Section 5.2.3, the feature co-occurrence patterns are less sensitive to poor feature types when performing graph transduction. Therefore, our approach achieves a noticeable performance improvement by combining all the individual feature types, despite some poor feature types and the disagreement among different feature types.

We also study the impact of the cluster number in each feature type. The performance comparison is shown in Table 5.1, in which the number of clusters per feature type varies from 5 to 200, with the size of labeled samples per class equal to 20. With the increase of cluster number per feature type, the accuracy increases first then decreases. This is because either under-clustering or over-clustering will discourage the investigation of data distributions in multiple feature types. Despite that, there still exists a large number of effective over-clustering which can produce informative feature clusters, boosting the performance of graph transduction. For example, when the cluster number per feature type is between 10 to 200, the labeling accuracies of unlabeled data all reach more than 90%.

FIGURE 5.4. Performance comparison on Oxford 17-category flowers.

FIGURE 5.5. Confusion matrix comparison on Oxford 17-category flowers.

5.3.4 Oxford Flower Dataset

Our approach can also combine different visual features for object recognition. The Oxford Flower Dataset is used for experiment, which is composed of 17 flower categories, including *Buttercup*, *Coltsfoot*, *Daffodil*, *Daisy*, *Dandelion*, *Fritillary*, *Iris*, *Pansy*, *Sunflower*, *Windflower*, *Snowdrop*, *LilyValley*, *Bluebell*, *Crocus*, *Tigerlily*, *Tulip*, *Cowslip*. Each category is with 80 images. We have shown 5 representative flowers for each class in Figure 4.2. In the experiment, we use seven pairwise distance matrices provided by the dataset. These matrices are precomputed respectively from seven types of image appearance features [127, 139]. Using these pairwise distances, we compute the similarities between pairs of features according to Eq. 5.1.

We label the first 30 samples per class and use them for transductive learning. The classification performance on the remaining 850 unlabeled samples is used for evaluation. We compare our approach with GTAM [5] (on the best single feature type, the worst single feature type) and RWMV [4] (on all feature types) w.r.t. mean value and standard deviation of classification accuracies in Figure 5.4. For each class, we randomly pick labeled data from the 30 labeled candidates and vary the size from 2 to 20. In Figure 5.5, we show the confusion matrices of compared methods when there are 20 labeled data samples for each class. Because we do not have the original features, we do not compare the results of feature concatenation.

As shown in Figure 5.4, the individual types of features all show poor performances. Moreover, the best and worst single feature types confuse in different flower classes (Figure 5.5 (a),(b)), resulting in a large performance gap. Therefore there are serious disagreements among different feature types. In this case, the effectiveness of the linear combination of similarity matrices is limited to reduce the classification confusion caused by different feature types. By comparing Figure 5.5 (c) and Figure 5.5 (a),(b), we can see that the confusion matrix generated by RWMV is only a slight smooth over different feature types. Hence RWMV only brings a little gain compared with the best single feature type (Figure 5.4). In contrast, the confusion matrices in Figure 5.5 (d)

and (e) show that our approach can adequately alleviate classification confusion either using 17 clusters or 100 clusters per feature type. The performances consequently show significant improvements over GTAM on individual types of features and RWMV on all feature types. As mentioned in Section 5.3.3, because of better exploring the feature clustering structures of individual feature types, our method using 100 clusters per feature type performs better than that of using 17 clusters per feature type.

5.3.5 Human Body Motion Dataset

In video data, appearance and motion features complement each other for body motion description and recognition. Therefore, in this section, we combine such two feature types for video recognition. We experiment on the recent Body Motion Dataset, which is included in UCF101 [129] and contains 1910 videos in total, with 16 categories of human body motion actions: *Baby Crawling*, *Blowing Candles*, *Body Weight Squats*, *Handstand Pushups*, *Handstand Walking*, *Jumping Jack*, *Lunges*, *Pull Ups*, *Push Ups*, *Rock Climbing Indoor*, *Rope Climbing*, *Swing*, *Tai Chi*, *Trampoline Jumping*, *Walking with a Dog*, *Wall Pushups*. For each category, one sample action has been shown in Figure 4.4. Each video is represented as dense appearance trajectories based on Histogram of Oriented Gradients (HOG) and dense motion trajectories based on Motion Boundary Histograms (MBH) [133].

We label the first 50 samples per class for transductive learning. For each class, we randomly pick the labeled data from the 50 candidates and vary the size from 2 to 20. The classification performance on the remaining 1110 unlabeled samples are used for evaluation. Again, we compare our approach with GTAM [5] (on individual feature types and feature concatenation) and RWMV [4] (on all feature types) in Table 5.2.

Comparing the first two columns of Table 5.2, we can see that motion features perform better than appearance features in human body motion classification. The 3rd and 4th columns show that the approaches of GTAM on feature concatenation and RWMV that

# labeled per class	GTAM with HOG	GTAM with MBH	GTAM with feature concat	RWMV with all feature types	Ours 16 clusters per feature type	Ours 50 clusters per feature type	Ours 100 clusters per feature type
20	0.088 ± 0.004	0.140 ± 0.007	0.104 ± 0.007	0.078 ± 0.007	0.340 ± 0.042	0.464 ± 0.040	0.511 ± 0.026
17	0.087 ± 0.003	0.135 ± 0.008	0.101 ± 0.008	0.080 ± 0.011	0.332 ± 0.040	0.465 ± 0.032	0.509 ± 0.022
14	0.088 ± 0.004	0.133 ± 0.013	0.103 ± 0.009	0.082 ± 0.012	0.320 ± 0.029	0.439 ± 0.046	0.488 ± 0.031
11	0.090 ± 0.004	0.135 ± 0.013	0.107 ± 0.007	0.097 ± 0.024	0.301 ± 0.039	0.416 ± 0.050	0.474 ± 0.025
8	0.089 ± 0.008	0.132 ± 0.014	0.102 ± 0.012	0.101 ± 0.030	0.261 ± 0.036	0.381 ± 0.039	0.424 ± 0.028
5	0.081 ± 0.012	0.118 ± 0.019	0.099 ± 0.019	0.089 ± 0.037	0.234 ± 0.034	0.353 ± 0.026	0.395 ± 0.037
2	0.081 ± 0.012	0.132 ± 0.029	0.103 ± 0.023	0.075 ± 0.019	0.197 ± 0.047	0.302 ± 0.038	0.317 ± 0.034

TABLE 5.2. Performance comparison on Human Body Motion videos.

uses all feature types usually perform better than GTAM on the poorer feature type, but still cannot compete against GTAM on the better feature type. Therefore they are not suitable to handle appearance and motion feature fusion. In contrast, our approach using 16 clusters per feature type (as shown in the 5th column) improves GTAM on the best single feature type. To further investigate clustering structures of individual feature types sufficiently, we over-cluster individual types of features and obtain 50 or 100 clusters per feature type. The results are shown in the last two columns of Table 5.2. This process brings a significantly improved performance in all labeled data sizes, which further verifies the effectiveness of our approach in fusing appearance and motion features.

5.3.6 UC Merced Land Use Dataset

To further evaluate our method, we conduct scene recognition experiment on UC Merced Land Use Dataset [128] and compare one more recent method [135] except for GTAM and RWMV. This dataset contains 21 classes of aerial orthoimagery: *agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium density residential, mobile home park, overpass, parking lot, river, runway, sparse residential, storage tanks, and tennis courts*. Each class has 100 images with resolution 256×256 . We have shown 5 sample images for each class in Figure 4.3. For each image, we extract SIFT features over the 16×16 patches with spacing of 6 pixels. By applying the locality-constrained linear coding (LLC) [130] on all SIFT features extracted from this dataset, and running spatial pyramid max pooling on images with 1×1 , 2×2 , and 4×4 sub-regions, we generate 3 scales of image representations with dimensionalities of 1×1024 , $2 \times 2 \times 1024$, and $4 \times 4 \times 1024$ as three feature types. The image representations with different scales result in different types of features.

We select the first 40 samples per class as the labeled data pool and vary the number (from 2 to 20) of labeled samples from the pool. The classification performance on the

# labeled per class	GTAM [5]	GTG [135]	RWMV [4]	Ours 21 clusters per feature type	Ours 50 clusters per feature type	Ours 100 clusters per feature type
20	0.334 ± 0.018	0.379 ± 0.012	0.304 ± 0.010	0.357 ± 0.020	0.485 ± 0.023	0.554 ± 0.023
17	0.331 ± 0.019	0.373 ± 0.018	0.298 ± 0.016	0.337 ± 0.028	0.484 ± 0.020	0.527 ± 0.023
14	0.340 ± 0.017	0.380 ± 0.018	0.293 ± 0.017	0.325 ± 0.029	0.458 ± 0.028	0.511 ± 0.035
11	0.334 ± 0.026	0.371 ± 0.020	0.290 ± 0.017	0.315 ± 0.028	0.452 ± 0.018	0.488 ± 0.025
8	0.333 ± 0.031	0.368 ± 0.022	0.291 ± 0.026	0.293 ± 0.026	0.409 ± 0.039	0.463 ± 0.037
5	0.320 ± 0.022	0.350 ± 0.018	0.276 ± 0.021	0.274 ± 0.027	0.372 ± 0.044	0.400 ± 0.036
2	0.310 ± 0.038	0.314 ± 0.021	0.243 ± 0.034	0.270 ± 0.043	0.314 ± 0.031	0.343 ± 0.067

TABLE 5.3. Performance comparison on UC Merced Land Use images.

remaining 1260 unlabeled samples is reported for evaluation. Besides GTAM [5] and RWMV [4], we also compare with graph transduction game (GTG) [135] in Table 5.3. For GTAM or GTG, we separately perform it on each single feature type or feature concatenation, and report the best performance it obtains. For RWMV and our method, we report the results of multi-feature fusion. As can be seen from the 1st to the 4th columns, GTG generally outperforms GTAM, RWMV, and performs better than our method with 21 clusters per feature type. However, by appropriately increasing the number of clusters per feature type, the classification performance of our method can be considerably enhanced as shown in the last two columns of Table 5.3. The results further justify the benefit of our method and the effectiveness of collaboration between clustering and classification. Overall, the performance gain depends on the spectral clustering results of using individual features, as well as the complementary among the multiple features.

5.4 Conclusion

The different data characteristics and distributions among multiple feature types challenge many existing multi-feature learning methods. Instead of iteratively updating individual feature type and forcing different feature types to agree with each other, we allow each feature type to perform data clustering by its own and then represent each data sample by a co-occurrence of feature patterns across different feature types. Relying on these feature co-occurrence representations of the data samples, we propose a transductive spectral learning approach, so that data labels can be transferred based on similar feature co-occurrence patterns. To transfer the labels from the labeled data to unlabeled data under our transductive learning formulation, we develop an algorithm that can iteratively refine the spectral clustering results of individual feature types and the labeling results of unlabeled data. The experiments on both synthetic and real-world

image/video datasets highlight the advantages of the proposed method to handle multi-feature fusion in transductive learning.

Chapter 6

Conclusion and Future Work

Over the past decade, visual pattern discovery has received increasing attention, especially in the communities of computer vision and data mining. In spite of recent progress, there are still some problems crying out for solutions. In this thesis, we mainly studied three problems of discovering visual patterns in image data:

- Combining feature context and spatial context for visual co-occurrence pattern discovery (Chapter 3). We investigate the spatial co-occurrence patterns among visual primitives and feature co-occurrence patterns among different types of features, and proposed to combine spatial context and feature context for image pattern discovery. We formulate our method as a regularized k -means clustering. An iterative self-learning procedure is developed. Such a procedure can iteratively refine the pattern discovery results and guarantees to converge.
- Multi-feature clustering for visual pattern clustering (Chapter 4). We explore an loss function consisting of both the unary term based on the cost of each individual feature modality and the pairwise disagreement term between any pair of feature modalities. After that, we propose a minimax optimization to minimize the maximum loss. This proposed approach has only one parameter and does not need to specify the fusion weights.

- Multi-feature transductive learning for visual pattern classification (Chapter 5).

To transfer the labels from the labeled data to unlabeled data under transductive learning formulation, we develop an algorithm that can iteratively refine the spectral clustering results of individual feature types and the labeling results of unlabeled data. This proposed approach allows each feature type to perform data clustering by its own and then represent each data sample by a co-occurrence of feature patterns across different feature types. As a result, the data samples of similar feature co-occurrence pattern will share the same label.

Regarding our methods, the use of the state-of-the-art features such as Fisher Vector [25–27], Vector of Locally Aggregated Descriptors (VLAD) [28], and Convolutional Neural Networks (CNN) based features [29–31] is possible to enhance the discovery of visual patterns. What matters is, with these features, we are able to obtain a favourable k -means clustering (Chapter 3) or spectral clustering (Chapters 4 and 5). Meanwhile, when performing multi-feature transduction (Chapter 5), enough labeled data will be helpful.

There are also a few potential applications to study in the future work: (i) In many computer vision applications, it is important to appropriately fuse different features to perform better recognition or detection task, such as using appearance and motion features for action recognition [73][42], RGB and depth features for object recognition [141], *etc.* Therefore, we can investigate how the proposed method can benefit these tasks. (ii) It is possible to explore the applications of image search by semi-supervised visual pattern classification and social relationship mining [142] by visual co-occurrence discovery. (iii) Besides the image domain and feature domain, our methods also have the potential to mine recurring patterns among any other multiple domains. Especially we are able to mine the hidden but massive links (*i.e.*, patterns) in heterogeneous information networks such as the Conference-Author Network [143], Image-Rich Information Networks formed by social image websites [144].

Besides that, there are still several open issues in visual pattern discovery that need to be addressed for further study, including:

- Interpreting visual patterns and effectively measure their quality. The interpretation and quality measure is crucial to visual pattern discovery. Despite a few successes in explaining visual patterns [8, 10, 54], we still need deeper investigation of spatial co-occurrences, geometric associations, and visual appearance of individual primitives, in order to better understand and utilize visual patterns.
- Selecting representative and discriminative patterns. Mining representative and discriminative patterns is a non-trivial problem as sometimes the two goals contradict to each other. However, depending on application, it is interesting to develop methods that can find such visual patterns, *e.g.*, local frequent histograms [51] and discriminative doublets [145].
- Effectively combining the bottom-up and top-down approaches of visual pattern discovery. Bottom-up methods capture local spatial cues of visual patterns while top-down methods model compositions of visual patterns [146, 147]. How to combine the strengths of bottom-up methods and top-down methods for visual pattern discovery will be an interesting research topic.

Appendix

This appendix contains the detailed analysis and technical proofs of some equations involved in this thesis.

A. Matrix Form of Hamming Distance

The Hamming distance between two $n \times 1$ binary vectors $\mathbf{x}_i$ and $\mathbf{y}_j$ can be represented as

$$\begin{aligned} d(\mathbf{x}_i, \mathbf{y}_j) &= n - [\mathbf{x}_i^T \mathbf{y}_j + (\mathbf{1} - \mathbf{x}_i)^T (\mathbf{1} - \mathbf{y}_j)] \\ &= n - [\mathbf{x}_i^T \mathbf{y}_j + n - \mathbf{1}^T \mathbf{y}_j - \mathbf{x}_i^T \mathbf{1} + \mathbf{x}_i^T \mathbf{y}_j] \\ &= -2\mathbf{x}_i^T \mathbf{y}_j + \mathbf{1}^T \mathbf{y}_j + \mathbf{x}_i^T \mathbf{1}, \end{aligned}$$

where $\mathbf{1}$ is the $n \times 1$ all 1 vector.

Suppose that we have $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_s]$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t]$, and the i th row and j th column element of the $s \times t$ matrix $\mathbf{D}$ denote the Hamming distortion between $\mathbf{x}_i$ and $\mathbf{y}_j$, i.e., $d(\mathbf{x}_i, \mathbf{y}_j)$. The matrix form of Hamming distortion is given by

$$\mathbf{D} = -2\mathbf{X}^T \mathbf{Y} + \mathbf{1}_Y \mathbf{Y} + \mathbf{X}^T \mathbf{1}_X,$$

where $\mathbf{1}_Y$ is $s \times n$ all 1 matrix, and $\mathbf{1}_X$ is $n \times t$ all 1 matrix.

B. Derivation of Eqs. 3.7, 3.8, 3.9

Substituting Eq. 3.6 into Eq. 3.2, and applying the three properties of matrix trace: (1) for square matrix $\mathbf{A}$ and $\mathbf{B}$, we have $\text{tr}(\mathbf{A}) = \text{tr}(\mathbf{A}^T)$; (2) for $m \times n$ matrix $\mathbf{A}$ and $n \times m$ matrix $\mathbf{B}$, we have $\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})$; (3) for constants α and β , and square matrices $\mathbf{A}$ and $\mathbf{B}$, we have $\alpha\text{tr}(\mathbf{A}) + \beta\text{tr}(\mathbf{B}) = \text{tr}(\alpha\mathbf{A} + \beta\mathbf{B})$; we have:

$$\begin{aligned}
\mathbf{J} &= \text{tr}(\mathfrak{R}^T \mathfrak{D}) + \lambda_v \text{tr}(\mathbf{R}_v^T \mathbf{D}_v) + \lambda_s \text{tr}(\mathbf{R}_s^T \mathbf{D}_s) \\
&= \text{tr}(\mathfrak{R}^T \mathfrak{D}) + \lambda_v \text{tr}(\mathbf{R}_v^T \mathbf{D}_v) \\
&\quad + \lambda_s \text{tr}[\mathbf{R}_s^T (-2\mathbf{U}_s^T \mathbf{R}_v \mathbf{Q}_s + \mathbf{1}_{T_s} \mathbf{R}_v \mathbf{Q}_s + \mathbf{U}_s^T \mathbf{1}_{U_s})] \\
&= \text{tr}(\mathfrak{R}^T \mathfrak{D}) + \lambda_v \text{tr}(\mathbf{R}_v^T \mathbf{D}_v) + \lambda_s \text{tr}[\mathbf{R}_s^T (\mathbf{U}_s^T \mathbf{1}_{U_s})] \\
&\quad + \lambda_s \text{tr}[\mathbf{R}_s^T (-2\mathbf{U}_s^T \mathbf{R}_v \mathbf{Q}_s + \mathbf{1}_{T_s} \mathbf{R}_v \mathbf{Q}_s)] \\
&= \text{tr}(\mathfrak{R}^T \mathfrak{D}) + \lambda_v \text{tr}(\mathbf{R}_v^T \mathbf{D}_v) + \lambda_s \text{tr}[\mathbf{R}_s^T (\mathbf{U}_s^T \mathbf{1}_{U_s})] \\
&\quad + \lambda_s \text{tr}[(-2\mathbf{U}_s^T \mathbf{R}_v \mathbf{Q}_s + \mathbf{1}_{T_s} \mathbf{R}_v \mathbf{Q}_s)^T \mathbf{R}_s] \\
&= \text{tr}(\mathfrak{R}^T \mathfrak{D}) + \lambda_v \text{tr}(\mathbf{R}_v^T \mathbf{D}_v) + \lambda_s \text{tr}[\mathbf{R}_s^T (\mathbf{U}_s^T \mathbf{1}_{U_s})] \\
&\quad + \lambda_s \text{tr}[\mathbf{Q}_s^T \mathbf{R}_v^T (-2\mathbf{U}_s^T + \mathbf{1}_{T_s})^T \mathbf{R}_s] \\
&= \text{tr}(\mathfrak{R}^T \mathfrak{D}) + \lambda_v \text{tr}(\mathbf{R}_v^T \mathbf{D}_v) + \lambda_s \text{tr}[\mathbf{R}_s^T (\mathbf{U}_s^T \mathbf{1}_{U_s})] \\
&\quad + \lambda_s \text{tr}[\mathbf{R}_v^T (-2\mathbf{U}_s^T + \mathbf{1}_{T_s})^T \mathbf{R}_s \mathbf{Q}_s^T] \\
&= \text{tr}\{\mathbf{R}_v^T [\lambda_v \mathbf{D}_v - \lambda_s (2\mathbf{U}_s^T - \mathbf{1}_{T_s})^T \mathbf{R}_s \mathbf{Q}_s^T]\} + \text{tr}(\mathfrak{R}^T \mathfrak{D}) \\
&\quad + \lambda_s \text{tr}[\mathbf{R}_s^T (\mathbf{U}_s^T \mathbf{1}_{U_s})].
\end{aligned}$$

Similarly, we can get Eq. 3.8 by substituting Eq. 3.5 into Eq. 3.2. Specifically, we have the derivation as follows:

$$\begin{aligned}
\mathbf{J} &= \text{tr}(\mathbf{R}^T \mathbf{D}) + \lambda_v \text{tr}(\mathbf{R}_v^T \mathbf{D}_v) + \lambda_s \text{tr}(\mathbf{R}_s^T \mathbf{D}_s) \\
&= \text{tr}(\mathbf{R}^T \mathbf{D}) + \lambda_s \text{tr}(\mathbf{R}_s^T \mathbf{D}_s) \\
&\quad + \lambda_v \text{tr}[\mathbf{R}_v^T (-2\mathbf{U}_v^T \mathbf{R} \mathbf{Q}_v + \mathbf{1}_{T_v} \mathbf{R} \mathbf{Q}_v + \mathbf{U}_v^T \mathbf{1}_{U_v})]
\end{aligned}$$

$$\begin{aligned}
&= \text{tr}(\mathbf{R}^T \mathbf{D}) + \lambda_s \text{tr}(\mathbf{R}_s^T \mathbf{D}_s) + \lambda_v \text{tr}[\mathbf{R}_v^T (\mathbf{U}_v^T \mathbf{1}_{\mathbf{U}_v})] \\
&\quad + \lambda_v \text{tr}[\mathbf{R}_v^T (-2\mathbf{U}_v^T \mathbf{R} \mathbf{Q}_v + \mathbf{1}_{\mathbf{T}_v} \mathbf{R} \mathbf{Q}_v)] \\
&= \text{tr}(\mathbf{R}^T \mathbf{D}) + \lambda_s \text{tr}(\mathbf{R}_s^T \mathbf{D}_s) + \lambda_v \text{tr}[\mathbf{R}_v^T (\mathbf{U}_v^T \mathbf{1}_{\mathbf{U}_v})] \\
&\quad + \lambda_v \text{tr}[(-2\mathbf{U}_v^T \mathbf{R} \mathbf{Q}_v + \mathbf{1}_{\mathbf{T}_v} \mathbf{R} \mathbf{Q}_v)^T \mathbf{R}_v] \\
&= \text{tr}(\mathbf{R}^T \mathbf{D}) + \lambda_s \text{tr}(\mathbf{R}_s^T \mathbf{D}_s) + \lambda_v \text{tr}[\mathbf{R}_v^T (\mathbf{U}_v^T \mathbf{1}_{\mathbf{U}_v})] \\
&\quad + \lambda_v \text{tr}[\mathbf{Q}_v^T \mathbf{R}^T (-2\mathbf{U}_v^T + \mathbf{1}_{\mathbf{T}_v})^T \mathbf{R}_v] \\
&= \text{tr}(\mathbf{R}^T \mathbf{D}) + \lambda_s \text{tr}(\mathbf{R}_s^T \mathbf{D}_s) + \lambda_v \text{tr}[\mathbf{R}_v^T (\mathbf{U}_v^T \mathbf{1}_{\mathbf{U}_v})] \\
&\quad + \lambda_v \text{tr}[\mathbf{R}^T (-2\mathbf{U}_v^T + \mathbf{1}_{\mathbf{T}_v})^T \mathbf{R}_v \mathbf{Q}_v^T] \\
&= \text{tr}\{\mathbf{R}^T [\mathbf{D} - \lambda_v (2\mathbf{U}_v^T - \mathbf{1}_{\mathbf{T}_v})^T \mathbf{R}_v \mathbf{Q}_v^T]\} \\
&\quad + \lambda_s \text{tr}(\mathbf{R}_s^T \mathbf{D}_s) + \lambda_v \text{tr}[\mathbf{R}_v^T (\mathbf{U}_v^T \mathbf{1}_{\mathbf{U}_v})].
\end{aligned}$$

Besides, note that $\lambda_s \text{tr}(\mathbf{R}_s^T \mathbf{D}_s) = \text{tr}[\mathbf{R}_s^T (\lambda_s \mathbf{D}_s)]$, this implies Eq. 3.9 from Eq. 3.2.

C. Deviation of Eq. 4.8

First, we expend Eq. 4.7 in the following:

$$\begin{aligned}
\mathcal{D}_{\mathbf{V}}(\mathbf{U}_i, \mathbf{U}_j) &= \|\mathbf{U}_i \mathbf{U}_i^T \mathbf{V} - \mathbf{U}_j \mathbf{U}_j^T \mathbf{V}\|_{\text{F}}^2 \\
&= \text{tr}\{(\mathbf{U}_i \mathbf{U}_i^T \mathbf{V} - \mathbf{U}_j \mathbf{U}_j^T \mathbf{V})^T (\mathbf{U}_i \mathbf{U}_i^T \mathbf{V} - \mathbf{U}_j \mathbf{U}_j^T \mathbf{V})\} \\
&= \text{tr}\{(\mathbf{V}^T \mathbf{U}_i \mathbf{U}_i^T - \mathbf{V}^T \mathbf{U}_j \mathbf{U}_j^T)(\mathbf{U}_i \mathbf{U}_i^T \mathbf{V} - \mathbf{U}_j \mathbf{U}_j^T \mathbf{V})\} \\
&= \text{tr}\{\mathbf{V}^T \mathbf{U}_i \mathbf{U}_i^T \mathbf{V} - \mathbf{V}^T \mathbf{U}_i \mathbf{U}_i^T \mathbf{U}_j \mathbf{U}_j^T \mathbf{V} \\
&\quad - \mathbf{V}^T \mathbf{U}_j \mathbf{U}_j^T \mathbf{U}_i \mathbf{U}_i^T \mathbf{V} + \mathbf{V}^T \mathbf{U}_j \mathbf{U}_j^T \mathbf{V}\} \\
&= \text{tr}\{\mathbf{V}^T [\mathbf{U}_i \mathbf{U}_i^T - 2\text{sym}(\mathbf{U}_i \mathbf{U}_i^T \mathbf{U}_j \mathbf{U}_j^T) + \mathbf{U}_j \mathbf{U}_j^T] \mathbf{V}\}.
\end{aligned}$$

Then, we can obtain

$$\mathcal{D}_{\mathbf{V}}(\mathbf{U}_i, \mathbf{V}) = \mathbf{V}^T [\mathbf{U}_i \mathbf{U}_i^T - 2\text{sym}(\mathbf{U}_i \mathbf{U}_i^T \mathbf{V} \mathbf{V}^T) + \mathbf{V} \mathbf{V}^T] \mathbf{V},$$

and

$$\mathcal{D}_{\mathbf{V}}(\mathbf{U}_j, \mathbf{V}) = \mathbf{V}^T[\mathbf{U}_j\mathbf{U}_j^T - 2\text{sym}(\mathbf{U}_j\mathbf{U}_j^T\mathbf{V}\mathbf{V}^T) + \mathbf{V}\mathbf{V}^T]\mathbf{V}.$$

When adding up the above equations, we apply the cyclicity property of matrix trace and leverage $\mathbf{V}^T\mathbf{V} = \mathbf{I}$. Finally, we can obtain:

$$\begin{aligned} \mathcal{Q}_{ij} &= \frac{1}{2}\{\mathcal{D}_{\mathbf{V}}(\mathbf{U}_i, \mathbf{U}_j) + \mathcal{D}_{\mathbf{V}}(\mathbf{U}_i, \mathbf{V}) + \mathcal{D}_{\mathbf{V}}(\mathbf{U}_j, \mathbf{V})\} \\ &= \text{tr}\{\mathbf{V}^T[\mathbf{I} - \text{sym}(\mathbf{U}_i\mathbf{U}_i^T\mathbf{U}_j\mathbf{U}_j^T)]\mathbf{V}\}. \end{aligned}$$

D. Derivation of Eq. 5.16

Given $\{\mathbf{R}_i\}_{i \neq k}$, $\mathbf{R}_v$ and $\mathbf{Y}$, we show how to obtain Eq. 5.16 from Ω_{smooth} (in Eq. 5.9).

By using the linearity and cyclicity property of the matrix trace, we have:

$$\begin{aligned} \Omega_{\text{smooth}}(\mathbf{R}_v, \{\mathbf{R}_j\}_{j=1}^K) &= \text{tr}(\mathbf{R}_v^T \mathbf{L}_v \mathbf{R}_v) \\ &= \text{tr}\{\mathbf{R}_v^T(\mathbf{I}_N - \mathbf{D}_v^{-\frac{1}{2}} \sum_{i=1}^K \mathbf{R}_i \mathbf{R}_i^T \mathbf{D}_v^{-\frac{1}{2}}) \mathbf{R}_v\} \\ &= \text{tr}\{\mathbf{R}_v^T \mathbf{R}_v\} - \text{tr}\{\mathbf{R}_v^T (\sum_{j=1}^K \mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_j \mathbf{R}_j^T \mathbf{D}_v^{-\frac{1}{2}}) \mathbf{R}_v\} \\ &= \text{tr}\{\mathbf{R}_v^T \mathbf{R}_v\} - \text{tr}(\sum_{j=1}^K \mathbf{R}_v^T \mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_j \mathbf{R}_j^T \mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_v) \\ &= \text{tr}\{\mathbf{R}_v^T \mathbf{R}_v\} - \sum_{j=1}^K \text{tr}(\mathbf{R}_v^T \mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_j \mathbf{R}_j^T \mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_v) \\ &= \text{tr}\{\mathbf{R}_v^T \mathbf{R}_v\} - \sum_{j=1}^K \text{tr}(\mathbf{R}_j^T \mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_v \mathbf{R}_v^T \mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_j) \\ &= \text{tr}\{\mathbf{R}_v^T \mathbf{R}_v\} - \sum_{j=1}^K \text{tr}\{\mathbf{R}_j^T (\mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_v \mathbf{R}_v^T \mathbf{D}_v^{-\frac{1}{2}}) \mathbf{R}_j\}. \end{aligned}$$

Substituting Eq. 1 into Eq. 5.9, and combining the constant terms, we obtain $\Omega = J + C$,

where $C = \alpha \text{tr}\{\mathbf{R}_v^T \mathbf{R}_v\} + \beta \text{tr}\{(\mathbf{R}_v - \mathbf{S}\mathbf{Y})^T (\mathbf{R}_v - \mathbf{S}\mathbf{Y})\}$ is unchanged since $\mathbf{R}_v$ and

$\mathbf{Y}$ are fixed. We therefore only need to minimize J , where

$$\begin{aligned}
 & J(\mathbf{R}_k, \mathbf{R}_v, \mathbf{Y}, \{\mathbf{R}_i\}_{i \neq k}) \\
 &= \sum_{i=1}^K \Omega_{\text{type}}(\mathbf{R}_i) - \alpha \sum_{j=1}^K \text{tr}\{\mathbf{R}_j^T (\mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_v \mathbf{R}_v^T \mathbf{D}_v^{-\frac{1}{2}}) \mathbf{R}_j\} \\
 &= \sum_{i=1}^K \text{tr}(\mathbf{R}_i^T \mathbf{L}_i \mathbf{R}_k) - \alpha \sum_{i=1}^K \text{tr}\{\mathbf{R}_i^T (\mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_v \mathbf{R}_v^T \mathbf{D}_v^{-\frac{1}{2}}) \mathbf{R}_i\} \\
 &= \sum_{i=1}^K \text{tr}\{\mathbf{R}_i^T (\mathbf{L}_i - \alpha \mathbf{D}_v^{-\frac{1}{2}} \mathbf{R}_v \mathbf{R}_v^T \mathbf{D}_v^{-\frac{1}{2}}) \mathbf{R}_i\},
 \end{aligned}$$

subject to $\mathbf{R}_k^T \mathbf{R}_k = \mathbf{I}_{M_k}$.

Author's Publications

Journal Publications

1. **Hongxing Wang**, Junsong Yuan, and Ying Wu. Context-Aware Discovery of Visual Co-occurrence Patterns. *IEEE Trans. Image Process. (T-IP)*, vol. 23, no. 4, pp. 1805-1819, 2014.
2. **Hongxing Wang** and Junsong Yuan. Collaborative Multifeature Fusion for Transductive Spectral Learning. *IEEE Trans. Cybern. (T-CYB)*, vol. 45 no.3, pp. 466-475, 2015.
3. **Hongxing Wang**, Gangqiang Zhao, and Junsong Yuan. Visual Pattern Discovery in Image and Video Data: a Brief Survey. *WIREs Data Min. Knowl. Discovery (WIREs-DMKD)*, vol. 4, no. 1, pp. 24-37, 2014.

Conference Publications

1. **Hongxing Wang**, Chaoqun Weng, and Junsong Yuan. Multi-feature Spectral Clustering with Minimax Optimization. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 4106–4113, 2014.
2. **Hongxing Wang**, Junsong Yuan, and Yap-Peng Tan. Combining Feature Context and Spatial Context for Image Pattern Discovery. In *Proc. IEEE Int. Conf. Data Min. (ICDM)*, pp. 764- 773, 2011.

3. Chaoqun Weng, **Hongxing Wang**, and Junsong Yuan. Hierarchical Sparse Coding based on Spatial Pooling and Multi-Feature Fusion. In *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, pp. 1-6, 2013.
4. Chaoqun Weng, **Hongxing Wang**, and Junsong Yuan. Learning Weighted Geometric Pooling for Image Classification. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, pp. 3805-3809, 2013.

Bibliography

- [1] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories,” in *CVPR Workshop on Generative-Model Based Vision*, 2004, pp. 178–178.
- [2] A. Quattoni and A. Torralba, “Recognizing indoor scenes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009.
- [3] G. Zhao and J. Yuan, “Discovering thematic patterns in videos via cohesive subgraph mining,” in *Proc. IEEE Int. Conf. Data Min.*, 2011, pp. 1260–1265.
- [4] D. Zhou and C. Burges, “Spectral clustering and transductive learning with multiple views,” in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 1159–1166.
- [5] J. Wang, T. Jebara, and S. Chang, “Graph transduction via alternating minimization,” in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 1144–1151.
- [6] T. Tuytelaars, C. Lampert, M. Blaschko, and W. Buntine, “Unsupervised object discovery: A comparison,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 284–302, 2010.
- [7] J. Yuan, *Discovering visual patterns in image and video data: concepts, algorithms, experiments*. Saarbrücken, Germany: VDM Verlag Dr. Müller, 2011.
- [8] S. Zhu, C. Guo, Y. Wang, and Z. Xu, “What are textons?” *Int. J. Comput. Vis.*, vol. 62, no. 1, pp. 121–143, 2005.

- [9] S. Fidler and A. Leonardis, “Towards scalable representations of object categories: Learning a hierarchy of parts,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
- [10] C. Li, D. Parikh, and T. Chen, “Automatic discovery of groups of objects for scene understanding,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012.
- [11] D. W. Thompson, *On growth and form*. Cambridge, UK: Cambridge University Press, 1961.
- [12] J. Han, H. Cheng, D. Xin, and X. Yan, “Frequent pattern mining: current status and future directions,” *Data Mining and Know. Disc.*, vol. 15, no. 1, pp. 55–86, 2007.
- [13] T. Tuytelaars and K. Mikolajczyk, “Local invariant feature detectors: a survey,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.
- [14] Y. Li, S. Wang, Q. Tian, and X. Ding, “A survey of recent advances in visual feature detection,” *Neurocomputing*, vol. 149, pp. 736–751, 2015.
- [15] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [16] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, “A comparison of affine region detectors,” *Int. J. Comput. Vis.*, vol. 65, no. 1-2, pp. 43–72, 2005.
- [17] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.
- [18] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.

-
- [19] C. L. Zitnick and P. Dollár, “Edge boxes: Locating object proposals from edges,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.
- [20] K. Grauman and B. Leibe, *Visual object recognition (Synthesis Lectures on Artificial Intelligence and Machine Learning)*. San Rafael, CA: Morgan & Claypool Publishers, 2011.
- [21] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.
- [22] J. Heinly, E. Dunn, and J.-M. Frahm, “Comparative evaluation of binary features,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 759–773.
- [23] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary robust independent elementary features,” in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 778–792.
- [24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: an efficient alternative to SIFT or SURF,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.
- [25] F. Perronnin and C. Dance, “Fisher kernels on visual vocabularies for image categorization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
- [26] F. Perronnin, J. Sánchez, and T. Mensink, “Improving the fisher kernel for large-scale image classification,” in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 143–156.
- [27] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, “Image classification with the fisher vector: Theory and practice,” *Int. J. Comput. Vis.*, vol. 105, no. 3, pp. 222–245, 2013.
- [28] H. Jégou, M. Douze, C. Schmid, and P. Pérez, “Aggregating local descriptors into a compact image representation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3304–3311.

- [29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [30] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” *arXiv:1312.6229*, 2013.
- [31] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” *arXiv:1408.5093*, 2014.
- [32] H. Wang, G. Zhao, and J. Yuan, “Visual pattern discovery in image and video data: a brief survey,” *WIREs Data Min. Knowl. Discovery*, vol. 4, no. 1, pp. 24–37, 2014.
- [33] B. Russell, W. Freeman, A. Efros, J. Sivic, and A. Zisserman, “Using multiple segmentations to discover objects and their extent in image collections,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1605–1614.
- [34] Y. Su and F. Jurie, “Visual word disambiguation by semantic contexts,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 311–318.
- [35] J. Yuan, Y. Wu, and M. Yang, “From frequent itemsets to semantically meaningful visual patterns,” in *ACM SIGKDD Int. Conf. Knowl. Discovery and Data Min.*, 2007, pp. 864–873.
- [36] —, “Discovery of collocation patterns: from visual words to visual phrases,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
- [37] J. Yuan and Y. Wu, “Mining visual collocation patterns via self-supervised subspace learning,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 42, no. 2, pp. 1–13, 2012.

-
- [38] S. Zhang, Q. Tian, G. Hua, Q. Huang, and W. Gao, “Generating descriptive visual words and visual phrases for large-scale image applications,” *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2664–2677, 2011.
- [39] Y. Zhang, Z. Jia, and T. Chen, “Image retrieval with geometry-preserving visual phrases,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 809–816.
- [40] J. Yuan, M. Yang, and Y. Wu, “Mining discriminative co-occurrence patterns for visual recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2777–2784.
- [41] H. Wang, J. Yuan, and Y. Wu, “Context-aware discovery of visual co-occurrence patterns,” *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1805–1819, 2014.
- [42] X. Cai, F. Nie, H. Huang, and F. Kamangar, “Heterogeneous image feature integration via multi-modal spectral clustering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1977–1984.
- [43] A. Kumar, P. Rai, and H. D. III, “Co-regularized multi-view spectral clustering,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1413–1421.
- [44] C. Xu, D. Tao, and C. Xu, “A survey on multi-view learning,” *arXiv:1304.5634*, 2013.
- [45] H. Wang, C. Weng, and J. Yuan, “Multi-feature spectral clustering with minimax optimization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 4106–4113.
- [46] H. Wang and J. Yuan, “Collaborative multifeature fusion for transductive spectral learning,” *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 466–475, 2015.
- [47] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *J. Mach. Learn.*, vol. 3, pp. 993–1022, 2003.

- [48] J. Sivic and A. Zisserman, “Video data mining using configurations of viewpoint invariant regions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. 488–495.
- [49] T. Quack, V. Ferrari, B. Leibe, and L. Van Gool, “Efficient mining of frequent and distinctive feature configurations,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007.
- [50] S. Kim, X. Jin, and J. Han, “Disiclass: discriminative frequent pattern-based image classification,” in *KDD Workshop on Multimedia Data Mining*, 2010, pp. 7:1–7:10.
- [51] B. Fernando, E. Fromont, and T. Tuytelaars, “Effective use of frequent itemset mining for image classification,” in *Proc. Eur. Conf. Comput. Vis.*, 2012.
- [52] W. Voravuthikunchai, B. Crémilleux, F. Jurie *et al.*, “Histograms of pattern sets for image classification and object recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014.
- [53] J. Yuan and Y. Wu, “Spatial random partition for common visual pattern discovery,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
- [54] A. Faktor and M. Irani, ““clustering by composition”-unsupervised discovery of image categories,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 474–487.
- [55] J. Gao, Y. Hu, J. Liu, and R. Yang, “Unsupervised learning of high-order structural semantics from images,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 2122–2129.
- [56] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki, “Attributed graph mining and matching: An attempt to define and extract soft attributed patterns,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014.

-
- [57] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Comm. of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [58] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian, “Spatial coding for large scale partial-duplicate web image search,” in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 511–520.
- [59] W. Zhou, H. Li, Y. Lu, and Q. Tian, “Large scale image search with geometric coding,” in *Proc. ACM Int. Conf. Multimedia*, 2011, pp. 1349–1352.
- [60] R. Fergus, P. Perona, and A. Zisserman, “Object class recognition by unsupervised scale-invariant learning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 2003, pp. 264–271.
- [61] P. Felzenszwalb and D. Huttenlocher, “Pictorial structures for object recognition,” *Int. J. Comput. Vis.*, vol. 61, no. 1, pp. 55–79, 2005.
- [62] B. Ommer and J. Buhmann, “Learning the compositional nature of visual objects,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
- [63] S. Todorovic and N. Ahuja, “Unsupervised category modeling, recognition, and segmentation in images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 12, pp. 2158–2174, 2008.
- [64] M. Choi, A. Torralba, and A. Willsky, “A tree-based context model for object recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, 2012.
- [65] Y. Wu, Z. Si, H. Gong, and S. Zhu, “Learning active basis model for object detection and recognition,” *Int. J. Comput. Vis.*, vol. 90, no. 2, pp. 198–235, 2010.
- [66] G. Bouchard and B. Triggs, “Hierarchical part-based visual object categorization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 710–715.

- [67] J. Sivic, B. Russell, A. Zisserman, W. Freeman, and A. Efros, “Unsupervised discovery of visual object class hierarchies,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
- [68] D. Parikh, C. Zitnick, and T. Chen, “Unsupervised learning of hierarchical spatial structures in images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 2743–2750.
- [69] C. Galleguillos, A. Rabinovich, and S. Belongie, “Object categorization using co-occurrence, location and appearance,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
- [70] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, “Affinity aggregation for spectral clustering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 773–780.
- [71] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, and Z. Tu, “Unsupervised metric fusion by cross diffusion,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2997–3004.
- [72] H. Wang, F. Nie, H. Huang, and C. Ding, “Heterogeneous visual features fusion via sparse multimodal machine,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013.
- [73] C. Xu, D. Tao, and C. Xu, “Large-margin multi-view information bottleneck,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1559–1572, 2014.
- [74] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas, “Query specific fusion for image retrieval,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 660–673.
- [75] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian, “Semantic-aware co-indexing for image retrieval,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1673–1680.

-
- [76] L. Zheng, S. Wang, Z. Liu, and Q. Tian, “Packing and padding: Coupled multi-index for accurate image retrieval,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1947–1954.
- [77] L. Cao, J. Luo, F. Liang, and T. Huang, “Heterogeneous feature machines for visual recognition,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 1095–1102.
- [78] P. Gehler and S. Nowozin, “On feature combination for multiclass object classification,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 221–228.
- [79] Y. Yeh, T. Lin, Y. Chung, and Y. Wang, “A novel multiple kernel learning framework for heterogeneous feature fusion and variable selection,” *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 563–574, 2012.
- [80] M. Blaschko and C. Lampert, “Correlational spectral clustering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008.
- [81] Z. Cai, L. Wang, X. Peng, and Y. Qiao, “Multi-view super vector for action recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014.
- [82] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proc. ACM Conf. Comp. Learn. Theory*, 1998, pp. 92–100.
- [83] U. Brefeld and T. Scheffer, “Co-EM support vector learning,” in *Proc. Int. Conf. Mach. Learn.*, 2004, pp. 16–23.
- [84] S. Bickel and T. Scheffer, “Multi-view clustering,” in *Proc. IEEE Int. Conf. Data Min.*, vol. 4, 2004, pp. 19–26.
- [85] G. F. Tzortzis and C. Likas, “Multiple view clustering using a weighted combination of exemplar-based mixture models,” *IEEE Trans. Neural Networks*, vol. 21, no. 12, pp. 1925–1938, 2010.
- [86] T. Lange and J. M. Buhmann, “Fusion of similarity data in clustering,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2005.

- [87] S. Yu, L.-C. Tranchevent, X. Liu, W. Glanzel, J. A. Suykens, B. De Moor, and Y. Moreau, “Optimized data fusion for kernel k-means clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1031–1039, 2012.
- [88] T. Xia, D. Tao, T. Mei, and Y. Zhang, “Multiview spectral embedding,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 40, no. 6, pp. 1438–1446, 2010.
- [89] B. Long, S. Y. Philip, and Z. M. Zhang, “A general model for multiple view unsupervised learning,” in *Proc. SIAM Int. Conf. Data Min.*, 2008, pp. 822–833.
- [90] Z. Akata, C. Thurau, C. Bauckhage *et al.*, “Non-negative matrix factorization in multimodality data for segmentation and label prediction,” in *Proc. Comput. Vis. Winter Workshop*, 2011.
- [91] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, “Multi-view clustering via canonical correlation analysis,” in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 129–136.
- [92] J. Yang, Z. Wang, Z. Lin, X. Shu, and T. Huang, “Bilevel sparse coding for coupled feature spaces,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2360–2367.
- [93] Y. Guo, “Convex subspace representation learning from multi-view data,” in *Proc. AAAI Conf. Artif. Intell.*, 2013.
- [94] X. Wang, B. Qian, J. Ye, and I. Davidson, “Multi-objective multi-view spectral clustering via pareto optimization,” in *Proc. SIAM Int. Conf. Data Min.*, 2013.
- [95] H. Wang, F. Nie, and H. Huang, “Multi-view clustering and feature learning via structured sparsity,” in *Proc. Int. Conf. Mach. Learn.*, 2013.
- [96] J. Yuan and Y. Wu, “Context-aware clustering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

-
- [97] H. Wang, J. Yuan, and Y. Tan, "Combining feature context and spatial context for image pattern discovery," in *Proc. IEEE Int. Conf. Data Min.*, 2011, pp. 764–773.
- [98] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *Proc. SIAM Int. Conf. Data Min.*, 2013.
- [99] Y. Jiang, J. Liu, Z. Li, P. Li, and H. Lu, "Co-regularized pls for multi-view clustering," in *Proc. Asian Conf. Comput. Vis.*, 2012, pp. 202–213.
- [100] A. Kumar and H. D. III, "A co-training approach for multi-view spectral clustering," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 393–400.
- [101] V. R. de Sa, P. W. Gallagher, J. M. Lewis, and V. L. Malave, "Multi-view kernel construction," *Mach. Learn.*, vol. 79, no. 1-2, pp. 47–71, 2010.
- [102] X. Guo, D. Liu, B. Jou, M. Zhu, A. Cai, and S.-F. Chang, "Robust object co-detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013.
- [103] V. Sa, "Learning classification with unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst.*, 1993, pp. 112–119.
- [104] S. Yu, B. Krishnapuram, R. Rosales, and R. Rao, "Bayesian co-training," *J. Mach. Learn.*, vol. 12, pp. 2649–2680, 2011.
- [105] O. Yakhnenko and V. Honavar, "Multiple label prediction for image annotation with multiple kernel correlation models," in *Comput. Vis. and Pattern Recognit. Workshops*, 2009, pp. 8–15.
- [106] S. Hwang and K. Grauman, "Learning the relative importance of objects from tagged images for retrieval and cross-modal search," *Int. J. Comput. Vis.*, vol. 100, no. 2, pp. 134–153, 2012.
- [107] B. Xie, Y. Mu, D. Tao, and K. Huang, "m-SNE: Multiview stochastic neighbor embedding," *IEEE Trans. Syst., Man, Cybern. B*, vol. 41, no. 4, pp. 1088–1096, 2011.

- [108] J. Yu, D. Liu, D. Tao, and H. S. Seah, “On combining multiple features for cartoon character retrieval and clip synthesis,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 42, no. 5, pp. 1413–1427, 2012.
- [109] X. Wang, B. Qian, J. Ye, and I. Davidson, “Multi-objective multi-view spectral clustering via pareto optimization,” in *Proc. SIAM Int. Conf. Data Min.*, 2013.
- [110] Y. Wang, X. Lin, and Q. Zhang, “Towards metric fusion on multi-view data: a cross-view based graph random walk approach,” in *Proc. ACM Conf. Inf. and Knowl. Manage.*, 2013, pp. 805–810.
- [111] C. Christoudias, R. Urtasun, and T. Darrell, “Multi-view learning in the presence of view disagreement,” in *Proc. Uncertain. Artif. Intell.*, 2008.
- [112] C. Weng, H. Wang, and J. Yuan, “Hierarchical sparse coding based on spatial pooling and multi-feature fusion,” in *Proc. IEEE Int. Conf. Multimedia Expo*, 2013, pp. 1–6.
- [113] N. Chen, J. Zhu, F. Sun, and E. Xing, “Large-margin predictive latent subspace learning for multi-view data analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2365–2378, 2012.
- [114] S. Jones and L. Shao, “Unsupervised spectral dual assignment clustering of human actions in context,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014.
- [115] Y. Lee and K. Grauman, “Object-graphs for context-aware visual category discovery,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 346–358, 2012.
- [116] J. Winn, A. Criminisi, and T. Minka, “Object categorization by learned universal visual dictionary,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005.
- [117] T. Malisiewicz and A. Efros, “Improving apatial support for objects via multiple segmentations,” in *Proc. British Machine Vis. Conf.*, vol. 2, 2007.

-
- [118] A. Bosch, A. Zisserman, and X. Munoz, “Representing shape with a spatial pyramid kernel,” in *Proceedings of the International Conference on Image and Video Retrieval*, 2007, pp. 401–408.
- [119] L.-J. Li and L. Fei-Fei, “What, where and who? classifying events by scene and object recognition,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
- [120] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.
- [121] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, “On spectral clustering: Analysis and an algorithm,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2, 2001, pp. 849–856.
- [122] H. Lütkepohl, *Handbook of matrices*. John Wiley & Sons, 1996.
- [123] J. W. Demmel, O. A. Marques, B. N. Parlett, and C. Vömel, “Performance and accuracy of lapack’s symmetric tridiagonal eigensolvers,” *SIAM J. Sci. Comput.*, vol. 30, no. 3, pp. 1508–1526, 2008.
- [124] J. Liu, C. Wang, M. Danilevsky, and J. Han, “Large-scale spectral clustering on graphs,” in *Int. Joint Conf. Artif. Intell.*, 2013.
- [125] T. Zhou and D. Tao, “Bilateral random projections,” in *IEEE Int. Symp. Inf. Theory Proc.* IEEE, 2012, pp. 1286–1290.
- [126] K. Bache and M. Lichman, “UCI mach. learn. repository,” 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [127] M. Nilsback and A. Zisserman, “A visual vocabulary for flower classification,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 2006, pp. 1447–1454.
- [128] Y. Yang and S. Newsam, “Spatial pyramid co-occurrence for image classification,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011.

- [129] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild,” *arXiv:1212.0402*, 2012.
- [130] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained linear coding for image classification,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010.
- [131] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *Int. J. Comput. Vis.*, 2001.
- [132] F. Palermo, J. Hays, and A. A. Efros, “Dating historical color images,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 499–512.
- [133] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, “Action recognition by dense trajectories,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3169–3176.
- [134] C. Ding, T. Li, and M. I. Jordan, “Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding,” in *Proc. IEEE Int. Conf. Data Min.*, 2008, pp. 183–192.
- [135] A. Erdem and M. Pelillo, “Graph transduction as a noncooperative game,” *Neural Comput.*, vol. 24, no. 3, pp. 700–723, 2012.
- [136] U. Von Luxburg, “A tutorial on spectral clustering,” *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
- [137] A. Ng, M. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2, 2001, pp. 849–856.
- [138] J. Wang, T. Jebara, and S.-F. Chang, “Semi-supervised learning using greedy max-cut,” *J. Mach. Learn.*, vol. 14, no. 1, pp. 771–800, 2013.

-
- [139] M.-E. Nilsback and A. Zisserman, “Automated flower classification over a large number of classes,” in *Proc. Indian Conf. Comput. Vis., Graphics Image Process.*, 2008.
- [140] M. P. W. van Breukelen, D. M. J. Tax, and J. E. den Hartog, “Handwritten digit recognition by combined classifiers,” *Kybernetika*, vol. 34, pp. 381–386, 1998.
- [141] J. Han, L. Shao, D. Xu, and J. Shotton, “Enhanced computer vision with microsoft kinect sensor: A review,” *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, 2013.
- [142] G. Wang, A. Gallagher, J. Luo, and D. Forsyth, “Seeing people in social context: Recognizing people and social relationships,” in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 169–182.
- [143] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu, “RankClus: integrating clustering with ranking for heterogeneous information network analysis,” in *Proc. Int. Conf. Extending Database Technology*, 2009, pp. 565–576.
- [144] X. Jin, J. Luo, J. Yu, G. Wang, D. Joshi, and J. Han, “iRIN: image retrieval in image-rich information networks,” in *Proc. Int. Conf. World Wide Web*, 2010, pp. 1261–1264.
- [145] S. Singh, A. Gupta, and A. Efros, “Unsupervised discovery of mid-level discriminative patches,” in *Proc. Eur. Conf. Comput. Vis.*, 2012.
- [146] M. Sun and H. V. hamme, “Image pattern discovery by using the spatial closeness of visual code words,” in *Proc. IEEE Int. Conf. Image Process.*, Brussels, Belgium, 2011, pp. 205–208.
- [147] G. Zhao, J. Yuan, and G. Hua, “Topical video object discovery from key frames by modeling word co-occurrence prior,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013.