STRUCTURAL SCENE ANALYSIS OF REMOTELY SENSED IMAGES USING GRAPH MINING

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Bahadır Özdemir

July, 2010

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Selim Aksoy (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Çiğdem Gündüz Demir

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Tolga Can

Approved for the Institute of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Institute

ABSTRACT

STRUCTURAL SCENE ANALYSIS OF REMOTELY SENSED IMAGES USING GRAPH MINING

Bahadır Özdemir
M.S. in Computer Engineering
Supervisor: Assist. Prof. Dr. Selim Aksoy
July, 2010

The need for intelligent systems capable of automatic content extraction and classification in remote sensing image datasets, has been constantly increasing due to the advances in the satellite technology and the availability of detailed images with a wide coverage of the Earth. Increasing details in very high spatial resolution images obtained from new generation sensors have enabled new applications but also introduced new challenges for object recognition. Contextual information about the image structures has the potential of improving individual object detection. Therefore, identifying the image regions which are intrinsically heterogeneous is an alternative way for high-level understanding of the image content. These regions, also known as compound structures, are comprised of primitive objects of many diverse types. Popular representations such as the bag-of-words model use primitive object parts extracted using local operators but cannot capture their structure because of the lack of spatial information. Hence, the detection of compound structures necessitates new image representations that involve joint modeling of spectral, spatial and structural information.

We propose an image representation that combines the representational power of graphs with the efficiency of the bag-of-words representation. The proposed method has three parts. In the first part, every image in the dataset is transformed into a graph structure using the local image features and their spatial relationships. The transformation method first detects the local patches of interest using maximally stable extremal regions obtained by gray level thresholding. Next, these patches are quantized to form a codebook of local information and a graph is constructed for each image by representing the patches as the graph nodes and connecting them with edges obtained using Voronoi tessellations. Transforming images to graphs provides an abstraction level and the remaining operations

for the classification are made on graphs. The second part of the proposed method is a graph mining algorithm which finds a set of most important subgraphs for the classification of image graphs. The graph mining algorithm we propose first finds the frequent subgraphs for each class, then selects the most discriminative ones by quantifying the correlations between the subgraphs and the classes in terms of the within-class occurrence distributions of the subgraphs; and finally reduces the set size by selecting the most representative ones by considering the redundancy between the subgraphs. After mining the set of subgraphs, each image graph is represented by a histogram vector of this set where each component in the histogram stores the number of occurrences of a particular subgraph in the image. The subgraph histogram representation enables classifying the image graphs using statistical classifiers. The last part of the method involves model learning from labeled data. We use support vector machines (SVM) for classifying images into semantic scene types. In addition, the themes distributed among the images are discovered using the latent Dirichlet allocation (LDA) model trained on the same data. By this way, the images which have heterogeneous content from different scene types can be represented in terms of a theme distribution vector. This representation enables further classification of images by theme analysis.

The experiments using an Ikonos image of Antalya show the effectiveness of the proposed representation in classification of complex scene types. The SVM model achieved a promising classification accuracy on the images cut from the Antalya image for the eight high-level semantic classes. Furthermore, the LDA model discovered interesting themes in the whole satellite image.

Keywords: Graph-based scene analysis, graph mining, scene understanding, remote sensing image analysis.

ÖZET

UYDU GÖRÜNTÜLERİNİN ÇİZGE MADENCİLİĞİ İLE YAPISAL SAHNE ANALİZİ

Bahadır Özdemir
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Y. Doç. Dr. Selim Aksoy
Temmuz, 2010

Uydu teknolojisindeki gelişmeler ve Dünya'nın geniş bir yüzeyini kapsayan detaylı görüntülerin mevcut olması, uydu görüntülerinde otomatik içerik çıkarma ve sınıflandırma yapabilen akıllı sistemlere duyulan ihtiyacı her geçen gün arttırmaktadır. Yeni nesil sensörlerden alınan çok yüksek uzamsal çözünürlüklü görüntülerdeki artan detaylar yeni uygulamaları mümkün kılmakla birlikte temel nesnelerin sezimini zorlaştırmaktadır. Görüntü yapıları hakkındaki bağlamsal bilgiler birbirinden bağımsız nesnelerin sezimini geliştirme potansiyeline sahiptir. Bu nedenle, özünde heterojen olan görüntü bölgelerinin tanımlanması, görüntü içeriğini anlamak için alternatif bir yoldur. Bileşik yapılar olarak da bilinen bu bölgeler birçok farklı türdeki temel nesnelerden oluşmaktadır. Kelimeler-torbası gibi popüler gösterimler, yerel operatörler kullanılarak çıkarılan temel nesne parçalarını kullanır fakat mekansal bilgi eksikliği nedeniyle onların yapısını tutamaz. Dolayısıyla, bileşik yapıların sezimi spektral, uzaysal ve yapısal bilgilerin ortak modellenmesini içeren yeni görüntü gösterimlerini zorunlu kılar.

Biz, çizgelerin gösterim gücü ile kelimeler-torbası gösteriminin verimliliğini birleştiren bir görüntü gösterimi öneriyoruz. Önerilen yöntem üç bölümden oluşmaktadır. İlk bölümde, veri kümesindeki her bir görüntü yerel görüntü özellikleri ve onların uzamsal ilişkileri kullanılarak çizge yapısına dönüştürülür. Dönüştürme yöntemi ilk olarak gri seviye eşiklemesi ile elde edilen en kararlı uç bölgelerden, ilgili yerel yamaları tespit eder. Sonra, bu yamalar bir yerel bilgi çizelgesi oluşturmak için nicelendirilir, ve yamaları çizge düğümü gibi göstererek ve onları Voronoi mozaiğinden elde edilen kenarlarla birleştirerek her bir görüntü için bir çizge inşa edilir. Görüntülerin çizgelere dönüştürülmesi bir soyutlama düzeyi sağlar ve sınıflandırma için geriye kalan işlemler çizgeler üzerinde yapılır. Önerilen yöntemin ikinci bölümü görüntü çizgelerinin sınıflandırılması

için en önemli altçizgelerin kümesini seçen bir çizge madenciliği algoritmasıdır. Önerdiğimiz çizge madenciliği algoritması ilk olarak her sınıf için sık görülen altçizgeleri bulur, sonra sınıf içinde görülme dağılımları açısından altçizgeler ve sınıflar arasındaki bağıntı miktarları ölçülerek en ayırt edici olanları seçer; ve son olarak altçizgeler arasındaki fazlalığı dikkate alarak en iyi temsil edenlerin seçmesiyle küme boyutunu küçültür. Altçizge kümesi madenciliğinden sonra her bir görüntü çizgesi, her bir bileşeninin bu kümenin belli bir altçizgesinin görüntüde görülme sayısını tuttuğu bir histogram vektörü ile gösterilir. Altçizge histogram gösterimi görüntü çizgelerinin istatistiksel sınıflandırıcılar kullanılarak sınıflandırılmasını mümkün kılar. Yöntemin son bölümü etiketli verilerinden model öğrenilmesini içerir. Görüntülerin anlamsal sahne türlerine sınıflandırılması için destek vektör makineleri (DVM) kullanıyoruz. Ek olarak, görüntüler üzerine dağılan temalar, aynı veriler üzerinde öğretilen gizli Dirichlet tahsisi (GDT) modeli kullanılarak keşfedilir. Bu sayede, farklı sahne türlerinden heterojen bir içeriğe sahip görüntüler bir tema dağılım vektörü olarak gösterilebilirler. Bu gösterim tema analizi ile görüntülerin daha ileri düzeyde sınıflandırılmasını sağlar.

Antalya'nın bir Ikonos görüntüsü üzerindeki deneyler önerilen gösterimin karmaşık sahne türlerinin sınıflandırılmasındaki etkinliğini göstermektedir. DVM modeli Antalya görüntüsünden kesilen görüntülerde sekiz üst düzey anlamsal sınıf için umut verici sınıflandırma doğruluğu elde etti. Ayrıca, GDT modeli tüm uydu görüntüsünde ilginç temalar keşfetti.

Anahtar sözcükler: Çizge tabanlı sahne analizi, çizge madenciliği, sahne anlayışı, uydu görüntüsü analizi.

Acknowledgement

I would like to express my sincere thanks to my advisor, *Selim Aksoy*, for his guidance, suggestions and support throughout the development of this thesis. He introduced me to the world of research, and encouraged me to develop my own ideas for the problem while supporting each step with his knowledge and advice. Whenever I got stuck in details, he provided me a different viewpoint. Working with him has been a valuable experience for me.

I would like extend my thanks to the members of my thesis committee, *Çiğdem Gündüz Demir* and *Tolga Can*, for reviewing this thesis and their suggestions about improving this work.

My special thanks must be sent to *Fatoş Tünay Yarman-Vural* who introduced me to computer vision when I was an undergraduate student at the Middle East Technical University.

I would like to express my deepest gratitude to my family, always standing by me, for their endless support and understanding.

I am very grateful to all those with whom I was having nice days in EA226: *Fırat*, *Daniya*, *Sare* and *Aslı*. I am also grateful to *Çağlar* and *Gökhan* for their comments on the method and the scientific discussions.

Finally, I would like to thank TÜBİTAK BİDEB (The Scientific and Technological Research Council of Turkey) for their financial support during my master's studies. This work was also supported in part by the TÜBİTAK CAREER grant 104E074.

Bahadır Özdemir
20 July 2010, Ankara

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Never use epigraphs, they kill the mystery in the work!

"The Black Book" – Orhan Pamuk

1.1 Overview

The amount of high-resolution satellite images is constantly increasing every day. Huge amount of information leads the requirement of automatic processing of remote sensing data by intelligent systems. Such systems usually perform image content extraction, classification and content-based retrieval in several application areas such as agriculture, ecology and urban planning. Very high resolution images become available by the advances in the satellite technology and processing of such images becomes feasible by the increasing computing power with the help of improvements in processor technology and parallel processing. This availability has enabled the study of multi-modal, multi-spectral, multi-resolution and multi-temporal data sets for monitoring purposes such as urban land use monitoring and management, geometric information system (GIS) and mapping, environmental change, site suitability, agricultural and ecological studies [2]. However, it also makes the problem of developing such intelligent systems more challenging because of the increased complexity.

Increasing details in very high spatial resolution images obtained from new generation sensors have been the main cause of the rising popularity of object-based approaches against traditional pixel-based approaches. Object-based approaches are aiming to identify primitive objects such as buildings and roads. Unfortunately, most algorithms cannot manage to detect such small objects in a very detailed image because segmentation algorithms usually fail to produce homogeneous regions corresponding to primitive structures. Contextual information about the image structures has the potential of improving individual object detection. Consequently, finding compound structures that correspond to high-level structures such as residential areas, forests, agricultural areas has become an alternative in image classification and high-level partitioning in the recent years because compound structures enable high-level understanding of image regions which are intrinsically heterogeneous [47]. Compound structures can be detected using local image features extracted from output of a segmentation algorithm or from interest points/regions. However, the detection of objects in such a detailed image is a difficult task. Therefore, some methods use textural analysis in lower resolution for detection of compound structures [42] or for detection/segmentation in high spatial resolution [19, 39]. In this thesis, we focus on representation of images by local image features with their spatial relationships and processing this representation model to detect compound structures in high spatial resolution.

1.2 Problem Definition

Pattern classification algorithms usually use one of the two traditional pattern recognition approaches: Statistical pattern recognition and syntactical/structural pattern recognition. Statistical approach uses feature vectors for object representation and generative or discriminative methods for modeling patterns in a vector space. The main advantage of this approach is available powerful algorithmic tools. On the other hand, structural approach uses strings or graphs for object representation. The main advantage of structural approach is the higher representation power and variable representation size. Both approaches have been used for detecting compound structures and image classification.

One of the statistical methods used for image classification is the *bag-of-words model*, which was originally developed for document analysis, adapted for images in [28]. Histogram of visual words obtained using a codebook constructed by quantizing local image patches has been a very popular representation for image classification in the recent years. This representation has been shown to give successful results for different image sets; however, a commonly accepted drawback is its disregarding of the spatial relationships among the individual patches as these relationships become crucial as contextual information for the understanding of complex scenes.

Structural approach used in image classification is aiming to represent images by graphs. Graphs provide powerful models where the nodes can store the local content and the edges can encode the spatial information. However, their use for image classification has been limited due to difficulties in translating complex image content to graph representation and inefficiencies in comparison of these graphs for classification. For example, the graph edit distance works well for matching relatively small graphs [37] but it can become quite restrictive for very detailed image content with a large number of nodes and edges.

We propose an intermediate representation that combines the representational power of graphs with the efficiency of the bag-of-words representation. The proposed method has three stages: transforming images into a graph representation, selecting the best subgraphs using a graph mining algorithm, and learning a model for each class to be used for classification. Figure 1.1 shows the overall flowchart of the algorithm.

Transforming images to graphs provides an abstraction level for images. Remaining operations for classification are made on graphs. Therefore, graphs transformed from images should contain sufficient information about the image content and spatial relationships. We describe a method for transforming the scene content and the associated spatial information of that scene into graph data. The method, which will be described in detail in Chapter 3, produces promising results on an Ikonos image of Antalya, Turkey (see Chapter 6).

The proposed approach represents each graph with a histogram of subgraphs

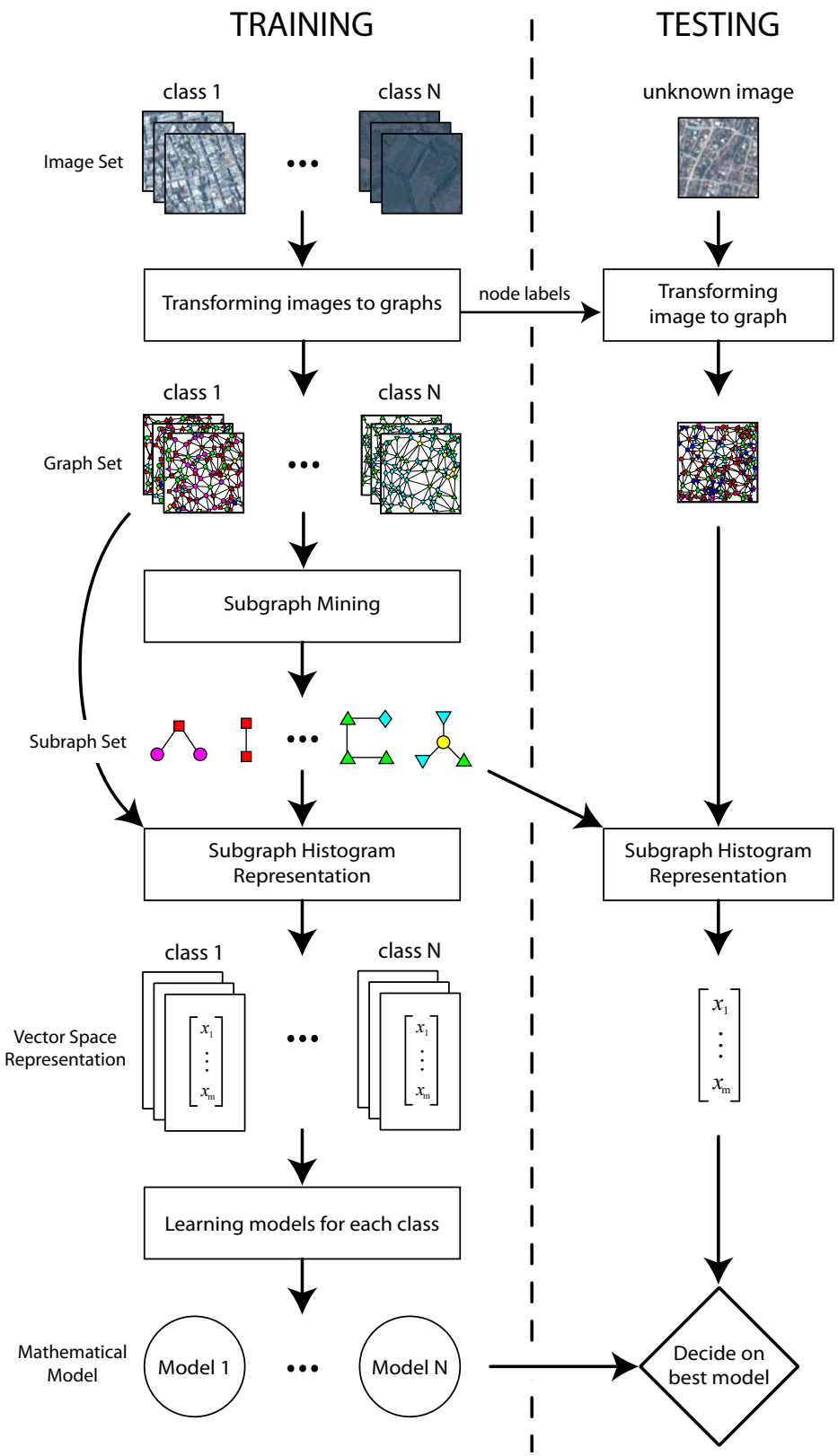


Figure 1.1: Overall flowchart of the algorithm

selected by a graph mining algorithm where the subgraphs encode the local patches and their spatial arrangements. The subgraphs are used to avoid the need of identifying a fixed arbitrary complexity (in terms of the number of nodes) and to require that they have a certain amount of support in different images in the data set. Partitioning remote sensing data into tiles usually produces images which contain heterogeneous regions of different classes. Some compound structures are naturally found near other structures. For example, orchards and greenhouses are usually detected near villages. Therefore, subgraphs selected by the algorithm should handle heterogeneous within-class content in an image set. A subgraph should also correspond to a structure particular to that class for classification purposes. Consequently, we propose a graph mining algorithm, where details can be found in Chapter 4, which tries to find a set of most important subgraphs considering frequency, correlation with classes and redundancy. Each image graph is represented by a histogram vector of this set in order to benefit from the advantages of statistical pattern recognition approach.

Finally, images represented by histogram vectors are classified in the vector space by traditional statistical classifiers. We employ *support vector machines* (SVM) for classifying images. In addition, topics/themes are discovered using latent probabilistic models such as *latent Dirichlet allocation* (LDA) that can be used for further classification of images for heterogeneous content. We show that good results for classification of images cut from large satellite scenes can be obtained for eight high-level semantic classes using support vector machines together with subgraph selection.

1.3 Data Set

The experiments are performed on an Ikonos image of Antalya, Turkey, consisting of a 12511×14204 pixel panchromatic band with 1m spatial resolution and four 3128×3551 pixel multi-spectral bands with 4m spatial resolution. In the experiments we use the panchromatic band and the pan-sharpened multi-spectral image produced by an image fusion method from visible multi-spectral bands and the

panchromatic band. The produced image approximates 1m spatial resolution in visible bands. We use the Antalya image because of its diverse content including several types of complex high-level structures such as dense and sparse residential areas with large and small buildings as well as fields and forests. The whole image was partitioned into 250×250 pixel tiles and these images were grouped into eight semantic classes, namely, (a) dense residential areas with large buildings, (b) dense residential areas with small buildings, (c) dense residential areas with trees, (d) sparse residential areas, (e) greenhouses, (f) orchards, (g) forests, and (h) fields. Only relatively homogeneous tiles, totally 585 images, are used in model learning and classification. The image and sample regions from every class are demonstrated in Figure 1.2.

1.4 Summary of Contributions

In this thesis, the goal is to correctly classify a given unknown image according to the models learned from training data for each class. Our framework for this aim has three parts and each part contains significant contributions.

The main contribution in the first part is a graph representation method for images. Although graphs offer higher representation power, their usage in computer vision has been below their usage in other fields. The primary reason for this issue is that images are not intrinsically in graph structure such as chemical compounds, program flows and social/computer networks. These data types come with their intrinsic graph structures and are perfectly suitable for structural approaches. The problem with graph representation of images is the difficulty of transforming image contents to graph structure. Most of the methods which construct graphs from images has used image segmentation algorithms so far [32, 23, 1, 22, 4]. In such methods, the regions in the output of segmentation usually correspond to graph nodes with labels determined by the features extracted from these regions whereas the edges encode the relationships between the regions. Unfortunately, precise segmentation of high spatial resolution satellite images as in Figure 1.2 is quite hard to obtain and this affects the performance

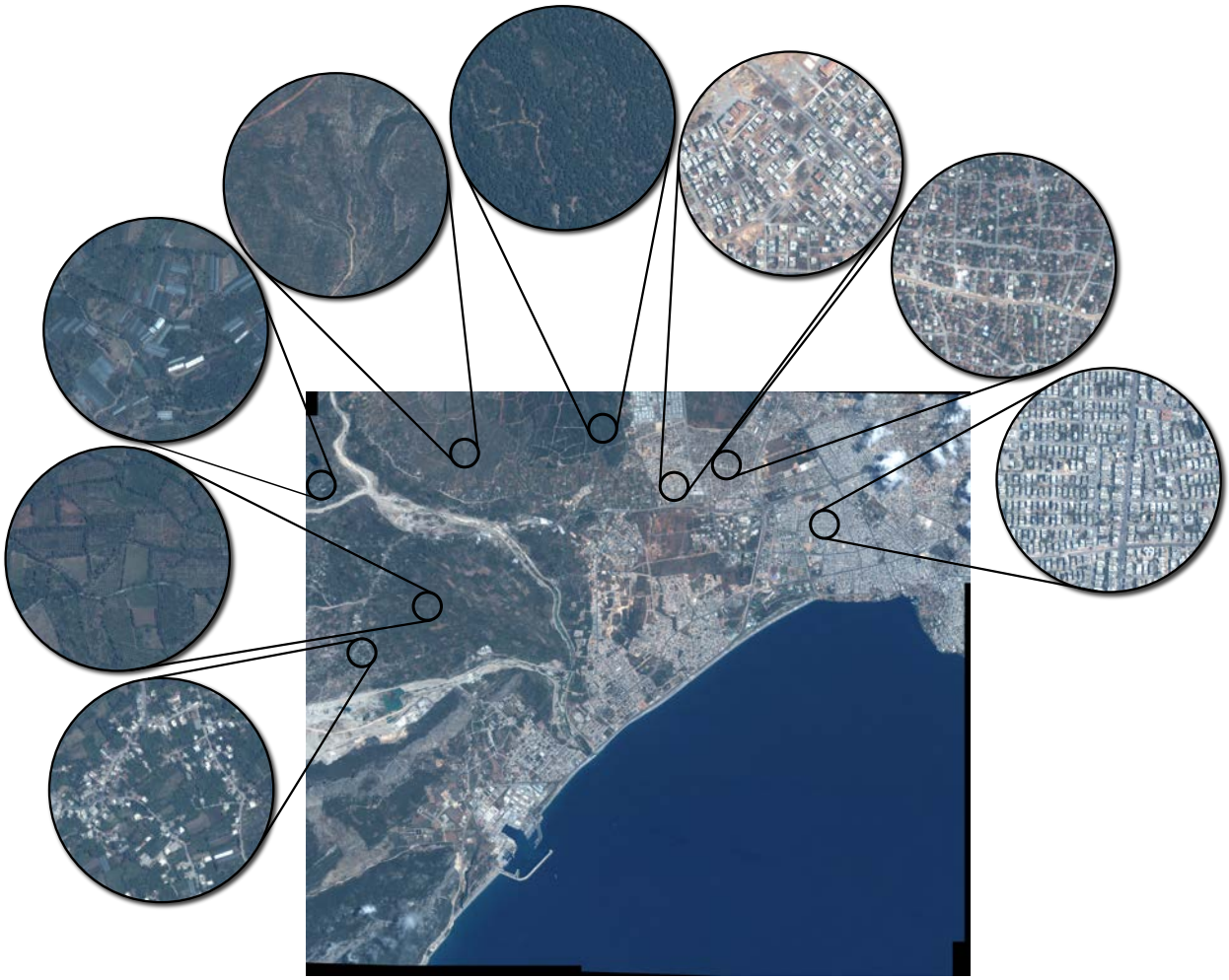


Figure 1.2: An Ikonos image of Antalya, and some compound structures of interest are zoomed in. The classes are (in clockwise order): Sparse residential areas, orchards, greenhouses, fields, forests, dense residential areas with small buildings, dense residential areas with trees, and dense residential areas with large buildings.

of graph representation negatively. Alternatively, we use regions of interests and their spatial relationships to transform image content into graph representation. Identifying only important regions in an image instead of whole image can supply sufficient information about the image content. First, local patches of interest are detected using *maximally stable extremal regions* obtained by gray level thresholding. We extract several features from these regions and their surroundings for better understanding of the regions. Next, these patches are quantized to form a codebook of local information, and a graph for each image is constructed by representing these patches as the graph nodes. The spatial relationships between the patches are identified using Voronoi tessellation and neighboring nodes are connected with edges. The abstraction level provided by the graph representation enables us to apply the same classification method on images coming from different sources like another satellite with different spatial resolution. For example, a QuickBird image can be classified in graph representation by a system trained on graphs constructed from Ikonos images as long as the node labels are compatible.

The second part proposes a graph mining algorithm to select the most important subgraphs for classification of graphs transformed from images. The mining algorithm we propose is a combination of three graph mining algorithms connected in series; in other words the output of one algorithm is the input of another one. The first algorithm seeks subgraphs frequently seen in a graph set. We use one of the popular algorithms in the graph mining literature for this purpose. Frequency criterion ensures the importance of subgraphs in the graph set. The most important contribution of this part is the second mining algorithm for finding correlated subgraphs which are frequently found in only one class of graphs and not in others. The available algorithms in the literature for correlated graph mining use a simple support definition which ignores the frequency of subgraph in a single graph and represents the support of a subgraph in a single graph as a binary relation of existence or absence [10, 34, 33]. We propose a novel algorithm where the frequency of subgraphs in a single graph are considered in the calculation of subgraph correlation (details are in Section 4.3). This method enhances classification performance considerably when images of a class cannot be fully homogeneous such as greenhouses seen in Figure 1.2. In such cases, this

method seeks subgraphs which are common among examples of that class, i.e. particular to that class. Final mining algorithm removes redundant subgraphs to avoid curse of dimensionality and selects the most significant subgraphs. The second and third mining algorithms work like a filter. They allow some subgraphs to pass to the next algorithm if they satisfy the criteria of the algorithms. The final set of subgraphs satisfying all criteria is used for representing a graph as a histogram vector where each component of the vector is the frequency of the corresponding subgraph in the given graph.

The third and last part is the classification of images using their vector representations by traditional classifiers like support vector machines. Addition to this, we use latent Dirichlet allocation to discover topics (themes) and their distribution in the image. This an important contribution because finding a homogeneous tile of a satellite image becomes harder when the tile size increases. Experimental results of the proposed methods are given in Chapter 6.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 presents an overview of related works in the literature. Chapter 3 introduces the method of transforming an image into graph representation. In Chapter 4, we first give a brief introduction to graph mining and then describe our graph mining algorithm. Chapter 5 explains learning models used for classification. Experimental results are given in Chapter 6, and Chapter 7 provides conclusions and future work.

Chapter 2

Literature Review

The knowledge and learning that we have, is, at most,
but little compared with that of which we are ignorant.

PLATO

In this chapter, we give the review of the previous studies on image classification using the bag-of-words model or the graph representation. The methods are divided into two sections according to their image representation. In the first section, we describe some image classification methods which are based on the bag-of-words model but also consider the spatial information of visual words. The second section describes the graph representation of images in the literature and their applications to image classification and retrieval.

2.1 Classification with Visual Words

The visual word concept is introduced in [28] as an image patch represented by a codeword from a large vocabulary of codewords. The vocabulary called codebook is formed by quantizing the image patches. Hence, an image is represented with a histogram of visual words. This analogy enables the usage of generative

probabilistic models of text corpora such pLSI and LDA in computer vision applications. These probabilistic models are based on the bag-of-words assumption [7], the exchangeability of visual words, that the location of patches in an image can be neglected. According to a recent survey [24], the bag-of-words model has been extended by weighting scheme, stop word removal, feature selection, spatial information and visual bi-gram. In relation to our study, we describe the extension methods which are using the spatial information and/or bi-gram of the visual words.

In [26], Lazebnik *et al.* add geometric correspondences to visual words by partitioning the image into increasingly sub-regions and computing the histograms of local features found inside each subregion. In [29], Li *et al.* propose the contextual bag-of-words representation to model two kinds of typical contextual relations between local patches, i.e., a semantic conceptual relation and a spatial neighboring relation. For the semantic conceptual relation, visual words are grouped on multiple semantic levels with respect to the similarity of the class distribution induced by the patches. To explore the spatial neighboring relation, the algorithm uses the visual n-gram approach. According to Yuan *et al.* [46], the clustering of primitive visual features tends to result in synonymous and polysemous visual words that bring large uncertainties and ambiguities in the representation. To overcome these problems, they propose a method which generates a higher-level lexicon, i.e. visual phrase lexicon, where a visual phrase is a meaningful spatially co-occurrent pattern of visual words. The method employs several data mining techniques and pattern summarization, with modifications to fit the image data.

2.2 Classification with Graph Representation

In this section, we give some previous works which use graph structure for image representation especially for classification and indexing/retrieval. An attributed relational graph (ARG) is a graph with attributes (also called labels or weights) on its nodes and/or edges. In computer vision applications, they are usually created from the output of a segmentation algorithm where each segment is denoted

by a node, and the edges are used to reflect the adjacent relations among the segments. In [23] ARGs are used to find the common pattern of the input images by finding the maximal common subgraph in the ARGs. In [1], Aksoy described a hierarchical approach for the content modeling and retrieval of satellite images using ARGs that combine region class information and spatial arrangements. The retrieval operation uses the graph edit distance [32] as the dissimilarity measure between two ARGs. Harchaoui and Bach propose *graph kernels* for supervised classification of image graphs constructed in a similar way from the morphological segmentation of images [21]. Another graph type used for image representation is *hypergraphs* where each edge is a subset of the set of nodes for modeling the higher-order relations between nodes [5]. Bunke *et al.* use hypergraphs to represent fingerprint images and classify those graphs using a hypergraph matching algorithm [11]. Unlike previous methods which construct image graph from the output of segmentation, in [20] Gao *et al.* construct graphs from corner points and Delaunay triangulation for the images of real world objects in black background. They cluster and classify image graphs by computing the graph edit distance between pairwise graphs.

Some methods transform the graphs constructed from images into feature vector and classify images in the vector space by statistical algorithms. These algorithm can be divided into two groups. In the first group of algorithm, each graph is transformed into a vector such that each of the components corresponds to the distance of the input graph to a predefined reference graph set. The studies [37] and [12] employ this approach for the datasets of symbol/letter images and fingerprint images using the Lipschitz Embedding [9] and the dissimilarity space representation [36], respectively. In the second group of algorithms, each graph is represented by a frequency vector of a subgraph set where the ith component is the number of occurrences of the $i$the subgraph in the input graph. The subgraph set is found by a graph mining algorithm for some criteria like frequency. A set of subgraphs found by the frequent subgraph mining of region-adjacency graphs is used for image indexing [22] and for clustering document images [4]. In [35], Nowozin *et al.* use weighted substructure mining which is combination of graph

mining and the boosting algorithm in order to classify images. In graph construction, each interest point is represented by one vertex and its descriptor becomes the corresponding vertex label and all vertices are connected by undirected edges with labels determined by the distance between two interest points.

Chapter 3

Transforming Images to Graphs

One morning, when Gregor Samsa woke from troubled dreams,
he found himself transformed in his bed into a horrible vermin.

"The Metamorphosis" – Franz Kafka

The first step of the algorithm is transforming every image to a graph structure as seen in Figure 1.1. Local image features and the relationships between them are encoded in the graph representation. In this chapter, we focus on this transformation process. Figure 3.1 shows the details for a sample image. First, local patches of interest in an image are detected using maximally stable extremal regions (MSER) obtained by gray level thresholding. Next, these patches are quantized to form a codebook of local information, and a graph for each image is constructed by representing these patches as the graph nodes and connecting them with edges obtained using Voronoi tessellations. The details of each step are explained in the following sections.

3.1 Finding Regions of Interest

The maximally stable extremal regions enable us to model local image content without the need for a precise segmentation that can be quite hard for high spatial

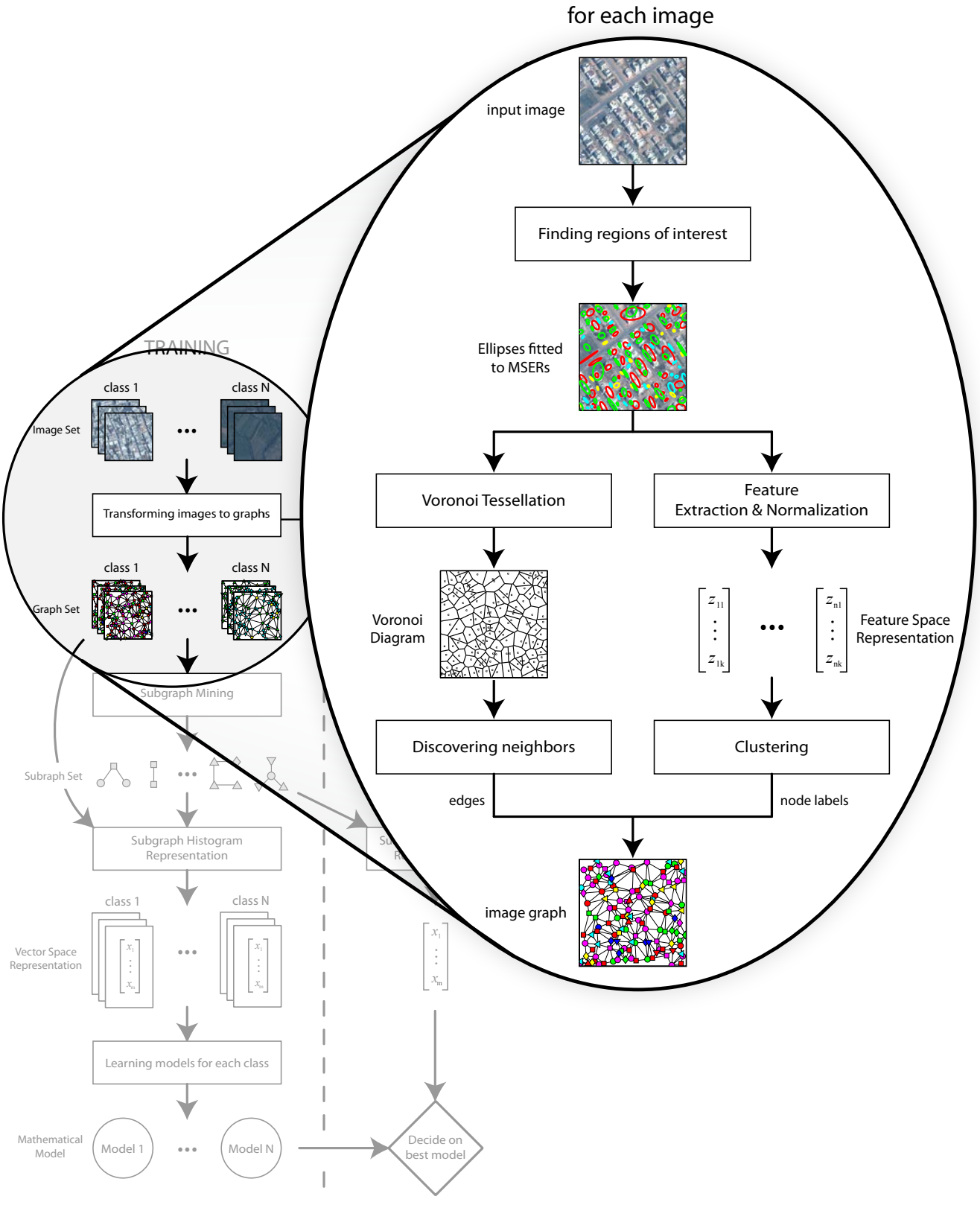


Figure 3.1: Steps of transforming images to graphs

resolution satellite images. In the following Section 3.1.1 the MSER algorithm is briefly described. The effects of MSER parameters for detecting regions of interest and different types of regions used in the algorithm are explained in Section 3.1.2.

3.1.1 Maximally Stable Extremal Regions

In this section, we introduce the *Maximally Stable Extremal Regions* (MSER), a new type of image elements proposed by Matas *et al.* in [31]. The regions are selected according to their extremal property of the intensity function in the region and on its outer boundary. The formal definition of the MSER concept and the necessary auxiliary definitions are given below.

Definition 3.1 (Maximally Stable Extremal Regions, [31])**.**
Image I is a mapping $I : D \subset \mathbb{Z}^2 \to S$. Extremal regions are well defined on images if:

1. S is totally ordered, i.e. reflexive, antisymmetric and transitive binary relation $\leq$ exists. Extremal regions can be defined on $S = \{0, 1, \ldots, 255\}$ or real-valued images ($S = \mathbb{R}$).
2. An adjacency relation $A \subset D \times D$ is defined. For example, 4-neighborhoods are used; $p, q \in D$ are adjacent (pAq) iff $\sum_{i=1}^{d} |p_i - q_i| \leq 1$.

Region Q is a contiguous subset of D, i.e. for each $p, q \in Q$ there is a sequence $p, a_1, a_2, \ldots, a_n, q$ and $pAa_1, a_iAa_{i+1}, a_nAq$.

(Outer) Region Boundary $\partial Q = \{q \in D \setminus Q \mid \exists p \in Q : qAp\}$, i.e. the boundary ∂Q of Q is the set of pixels adjacent to at least one pixel of Q but not belonging to Q.

Extremal Region $Q \subset D$ is a region such that either for all $p \in Q$, $q \in \partial Q$: $I(p) > I(q)$ (maximum intensity region) or $p \in Q, q \in \partial Q : I(p) < I(q)$ (minimum intensity region).

Maximally Stable Extremal Region Let $Q_1, \ldots, Q_{i-1}, Q_i, \ldots$ be a sequence of nested extremal regions, i.e. $Q_i \subset Q_{i+1}$. Extremal region Q_{i*} is maximally stable iff $q(i) = |Q_{i+\Delta} \setminus Q_{i-\Delta}| \,/\, |Q_i|$ has a local minimum at $i*$. $\Delta \in S$ is a parameter of the method.

—

The MSER algorithm is similar to the *watershed algorithm* except their outputs. In watershed computation, we deal with only the thresholds where regions merge, so resultant regions are highly unstable. In MSER detection, we seek a range of thresholds where the size of regions are effectively unchanged. Since every extremal region is a connected component of a thresholded image, all possible thresholds are applied to image and the stability of extremal regions are evaluated to find MSERs.

As given in the formal definition 3.1 the intensity of extremal regions can be less or greater than its boundary. We prefer calling dark MSER and bright MSER for minimum intensity MSER and maximum intensity MSER, respectively. The algorithm is generally implemented to detect dark MSERs and the intensity of input image is inverted to detect bright MSERs.

In our study, we use the VLFeat implementation of the MSER algorithm [43]. This implementation provides a rotation-invariant region descriptor and additional parameters which offer extra control over selection of MSERs. These parameters are related to area, variation (stability) and the diversity of extremal regions.

Let Q_i be an extremal region at the threshold level i. The following tests are performed for every MSER:

- *Area*: exclude too small or too big MSERs, $a_- \leq |Q_i| \leq a_+$.
- *Variation*: exclude too unstable MSERs, $v(Q_i) < v_+$ where VLFeat implementation differently uses stability score as $v(Q_i) = |Q_{i+\Delta} \setminus Q_i| \,/\, |Q_i|$.
- *Diversity*: remove duplicated MSERs, for any MSER Q_i find the parent

MSER Q_j and check if $|Q_j \setminus Q_i| \,/\, |Q_j| < d_+$ where Q_j is the parent of Q_i iff $Q_i \subset Q_j$ for $i \leq j \leq i + \Delta$.

We denote MSER parameter set as $\Omega = (\Delta, a_-, a_+, v_+, d_+)$. These parameters are used to eliminate less important extremal regions, i.e. too small or too big regions. The stability criterion is adjusted by parameters both Δ and v_+. The graph representation should encode both local image features and their spatial relationships correctly. Therefore, regions of interests should not share any pixel like in segmentation to transform planar relationships between regions. However, multiple thresholds may yield stable extremal regions for some parts of the image and the output is nested subset regions [31]. In this study, we always set $d_+ = 0$ to prevent overlapping extremal regions (actually one covers another).

Ellipsoids

MSERs have arbitrary shapes as seen in Figures 3.2(b) and 3.2(c) for given input image in Figure 3.2(a). Therefore, many implementations return extremal regions as a set of ellipsoids fitted to actual regions. Ellipsoids are represented with two parameters: mean vector and covariance matrix of the pixels composing the region. The parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of extremal region Q are computed as

$$\boldsymbol{\mu} = \frac{1}{|Q|} \sum_{\mathbf{x} \in Q} \mathbf{x}, \qquad \boldsymbol{\Sigma} = \frac{1}{|Q|} \sum_{\mathbf{x} \in Q} (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top \tag{3.1}$$

where the pixel coordinate $\mathbf{x} = (x_1, \ldots, x_n)^\top$ uses the standard index order and ranges. The MSER algorithm can also be applied to volumetric images; however, in this study we only deal with 2D grayscale images ($n = 2$). Thus, $\boldsymbol{\mu}$ has two components and $\boldsymbol{\Sigma}$ has three independent components because covariance matrix is a symmetric positive definite matrix. Ellipses fitted to MSERs in Figures 3.2(b) and 3.2(c) are drawn in Figures 3.2(d) and 3.2(e), respectively. The ellipses are drawn at $(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) = 1$. *

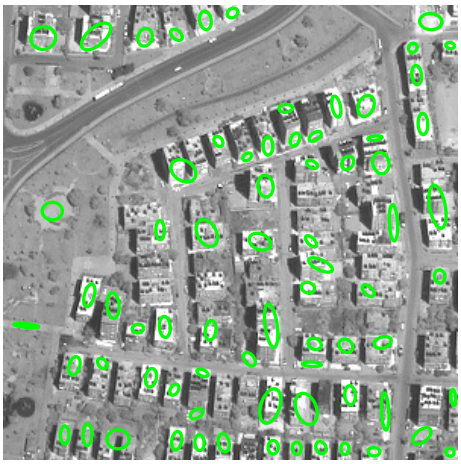
*The quantity $r^2 = (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$ is called the squared Mahalanobis distance from $\mathbf{x}$ to $\boldsymbol{\mu}$.

(a) input image

(b) dark MSERs

(c) bright MSERs

(d) ellipses fitted to dark MSERs



(e) ellipses fitted to bright MSERs

Figure 3.2: A given input image dark and bright MSERs, and ellipses fitted to them for parameters $\Omega = (\Delta, a_-, a_+, v_+, d_+) = (10, 60, 5000, 0.4, 1)$.

3.1.2 Types of Interest Regions

To handle all regions of interest by a single global parameter set is hard to obtain for an image set including different complex scene types. For example, extremal regions observed in urban areas are usually highly stable while such an observation in fields is less possible. We define two parameter sets with different stability criteria, Ω_{high} and Ω_{low}, to detect extremal regions such as in both urban areas and fields. In addition, it allows us to group extremal regions according to their stability scores. Applying the MSER algorithm with these parameters on both the intensity image (for dark MSERs) and on the inverted image (for bright MSER) results in four different region groups as:

- Highly stable dark MSERs (stable_dark)
- Highly stable bright MSERs (stable_bright)
- Less stable dark MSERs (unstable_dark)
- Less stable bright MSERs (unstable_bright)

Due to the definition of MSER, less stable MSERs cover highly stable ones. Therefore, we use restrictions on less stable ones. The set definitions of these four groups are given by

$$\mathsf{stable\_dark}(I) = \{R \,|\, R \subset I \,\wedge\, R \text{ is an MSER satisfying } \Omega_{\text{high}}\}, \tag{3.2}$$

$$\mathsf{stable\_bright}(I) = \{R \,|\, R \subset \bar{I} \,\wedge\, R \text{ is an MSER satisfying } \Omega_{\text{high}}\} \tag{3.3}$$

where $\bar{I}$ denotes the intensity inverted image of I. Similarly, less stable ones are defined as

$$\begin{aligned}\mathsf{unstable\_dark}(I) = \{R \,|\, R \subset I \,\wedge\, R \text{ is an MSER satisfying } \Omega_{\text{low}},\\ \wedge\, \forall R' \in \mathsf{stable\_dark}(I) : R \cap R' = \emptyset\},\end{aligned} \tag{3.4}$$

$$\begin{aligned}\mathsf{unstable\_bright}(I) = \{R \,|\, R \subset \bar{I} \,\wedge\, R \text{ is an MSER satisfying } \Omega_{\text{low}}\\ \wedge\, \forall R' \in \mathsf{stable\_bright}(I) : R \cap R' = \emptyset\}\end{aligned} \tag{3.5}$$

Figure 3.3 shows these four groups of MSERs for three different scene types. As seen in the figure, stable MSERs are observed especially on buildings and their shadows while unstable ones are seen everywhere like random sampling.

3.2 Feature Extraction

We extract several features from MSERs to identify the location where they are observed. Interest regions become more discriminative with their surroundings. The size of ellipses fitted to MSERs are expanded before extracting features from these regions. This method is proposed by Sivic *et al.* in [40]. We group the pixels inside expanded ellipses into two sets. The first set represents the MSER region and consists of pixels near to ellipse center whereas the other group containing outer pixels represents the surroundings of the MSER. As mentioned previously, each MSER is represented with two parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We denote the inner and outer groups of pixels as R_{in} and R_{out}, respectively. Image I is defined on $D \subset \mathbb{Z}^2$, then two groups are defined by

$$R_{in} = \left\{ \mathbf{x} \in D \;\middle|\; (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \leq r_1^2 \right\}, \tag{3.6}$$

$$R_{out} = \left\{ \mathbf{x} \in D \;\middle|\; r_1^2 < (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \leq r_2^2 \right\} \tag{3.7}$$

where every $\mathbf{x}$ represents a single pixel coordinate. For a given MSER, expanded ellipses and the pixels in regions R_{in}, R_{out} are shown on both panchromatic and multispectral bands in Figure 3.4.

We extract 17 rotation-invariant features from each MSER. Exactly 10 of them are basic features such as mean and standard deviation extracted from both R_{in} and R_{out}. Table 3.1 shows these basic 10 features.

The other 7 features are computed from the union group, $R_{all} = R_{in} \cup R_{out}$. These are 4 granulometry features, area and aspect ratio of ellipse, and moment of inertia.

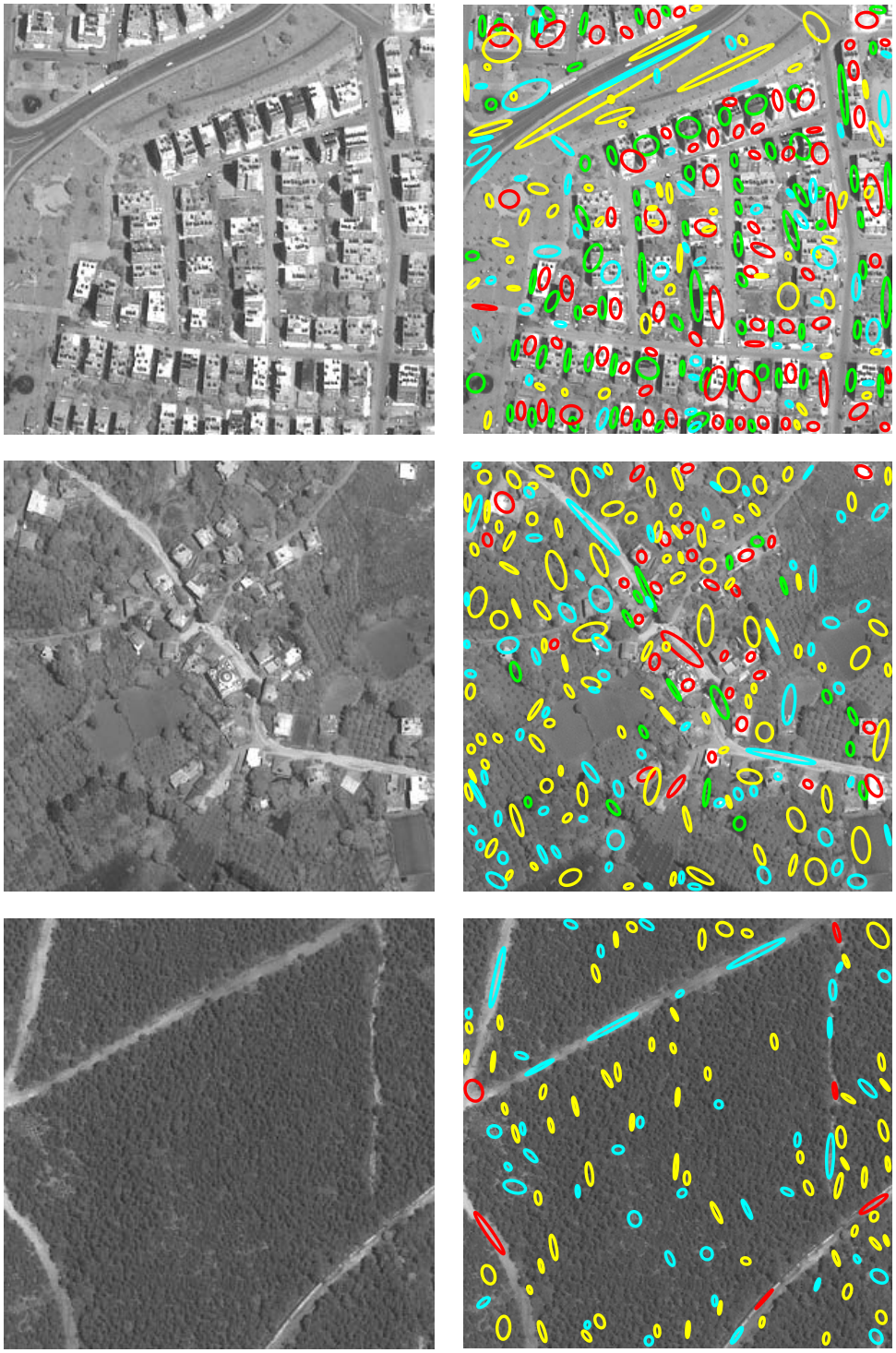


Figure 3.3: Ellipses fitted to MSER groups stable_dark, stable_bright, unstable_dark and unstable_bright are drawn with green, red, yellow and cyan, respectively on different scene types for parameter sets $\Omega_{\text{high}} = (10, 60, 5000, 0.4, 1)$ and $\Omega_{\text{low}} = (5, 35, 1000, 4, 1)$.

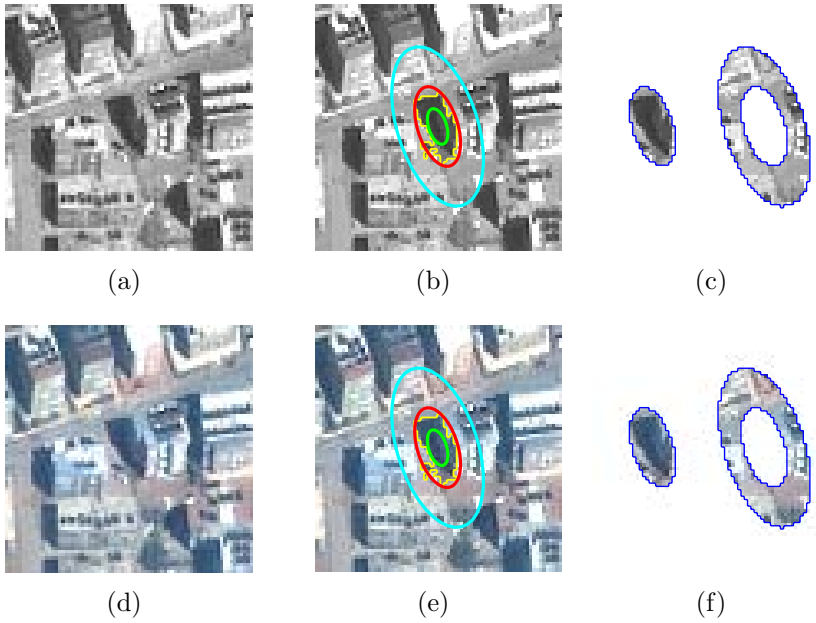


Figure 3.4: Satellite image of same region is given in (a) panchromatic and (d) visible multispectral bands. In (b) and (e), a given MSER is drawn with yellow and ellipse fitted to this MSER is drawn with green. Expanded ellipses at squared Mahalanobis distance $r_1^2 = 5$ and $r_2^2 = 20$ are drawn with red and cyan, respectively. In (c) and (f), pixels in R_{in} and R_{out} are shown for different bands.

Table 3.1: Ten basic features extracted from four bands and two regions.

			Region	
			R_{in}	R_{out}
Image	Panchromatic band		mean, standard deviation	mean, standard deviation
	Multispectral bands	Red band	mean	mean
		Green band	mean	mean
		Blue band	mean	mean

Granulometry

Granulometry is a technique to analyze the size and shape of granular materials. The idea is based on sieving a sample through various sized and shaped sieves [44]. A collection of grains is analyzed by sieving through sieves with increasing mesh size while measuring the mass retained by each sieve [41].

The concept of granulometry is extended for images by considering them as grains and applying morphological opening and closing with a family of structuring elements with increasing sizes [41]. Morphological opening provides information about image contents which are brighter than their neighborhoods and in contrast closing operation gives information about regions darker than their neighborhoods. Size information of these structures are obtained from the size of structuring element used in the morphological operation. Besides the information gained from standard deviation, granulometry produces useful information about the arrangement of objects in the expanded ellipse region.

We use only two sizes of structuring element, a disk with radii 2 and 7. They are employed to detect smaller and bigger structures in the image, respectively. The granulometry features are extracted from the region R_{all} in panchromatic band using morphological opening and closing, resulting in 4 granulometry features. Let ψ denote the structuring element, we compute the granulometry feature, Φ, known as normalized size distribution as

$$\Phi(I, \psi) = \frac{\sum_{\mathbf{x} \in R_{all}} \big(I \circ \psi\big)(\mathbf{x})}{\sum_{\mathbf{x} \in R_{all}} I(\mathbf{x})} \tag{3.8}$$

where $\circ$ denotes morphological opening; for morphological closing features it should be replaced by $\bullet$ denoting morphological closing.

Figure 3.5 shows results of morphological opening and closing with disk structuring element with radii 2 and 7 on sample images from three different classes. As shown in the figure, the urban area image is affected from morphological operations the most and the forest image is affected the least.

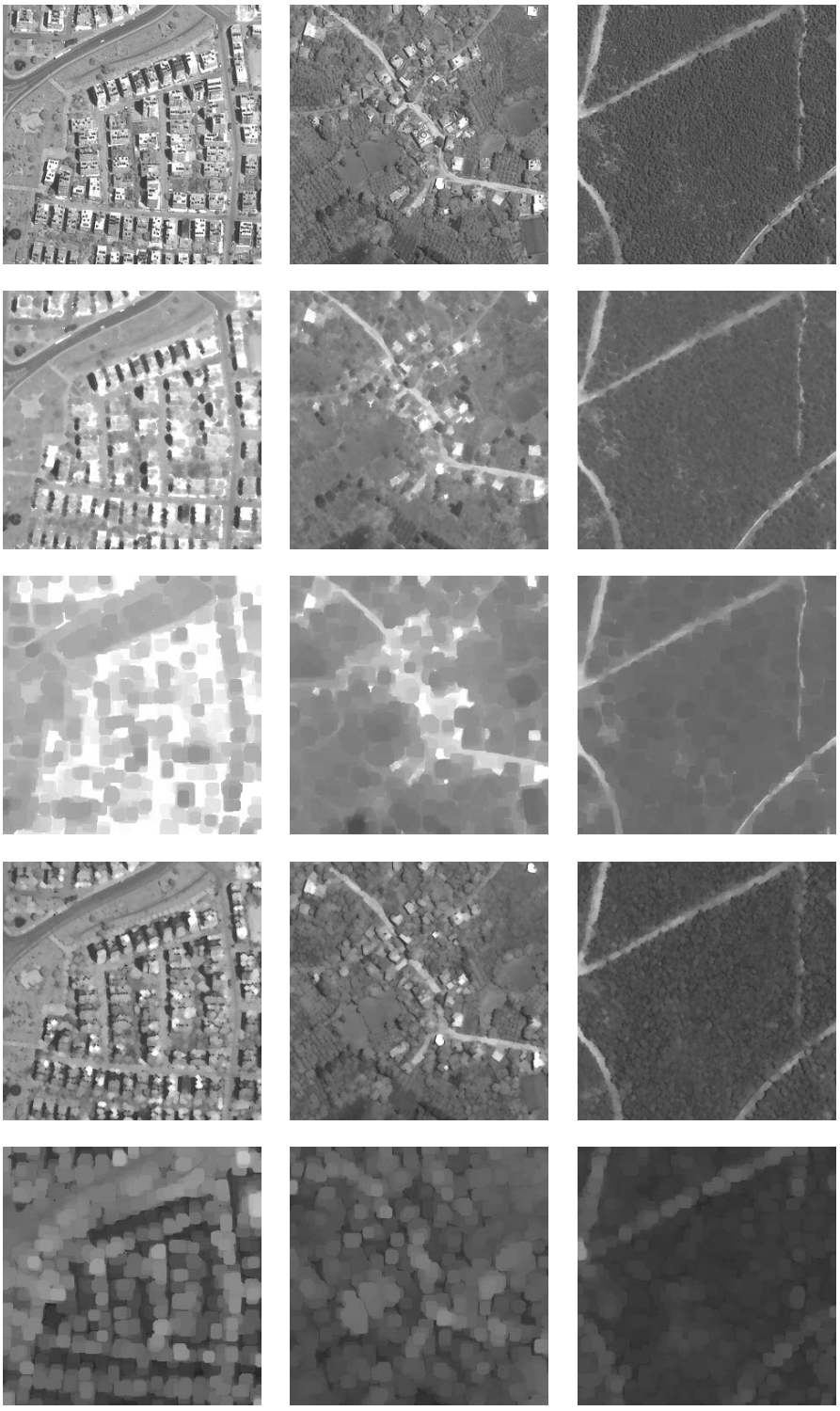


Figure 3.5: Results of morphological operations on images from three different classes. Images from top to down are in the order: original images, images closed by disk with radii 2, images closed by disk with radii 7, images opened by disk with radii 2 and images opened by disk with radii 7.

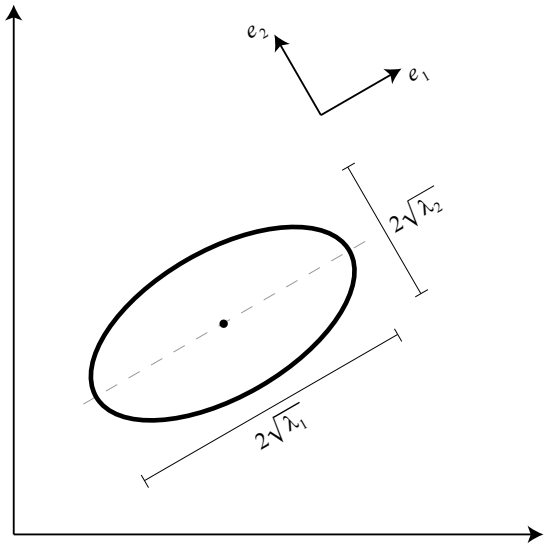


Figure 3.6: A sample ellipse and its eigenvectors e_1 and e_2 are shown, corresponding eigenvalues are λ_1 and λ_2, respectively. Major and minor diameters are also shown.

Moment of Inertia

Another feature computed from R_{all} is the moment of inertia. It provides useful information about intensity distribution in the expanded region with respect to the distance to ellipse center. The level of intensity change between the MSER and its surrounding can be identified with this feature. The formula is given below

$$M_I = \frac{\sum_{\mathbf{x}\in R_{all}} I(\mathbf{x}) \cdot (\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu}) \,/\, r_2^2}{\sum_{\mathbf{x}\in R_{all}} I(\mathbf{x})}. \tag{3.9}$$

The value of M_I is in the range $[0, 1]$ due to division by r_2^2 in the numerator.

Area and Aspect Ratio of Ellipse

The last two features are the area and aspect ratio of ellipse. These features give information about the shape of MSER. These features are calculated using the eigenvalues of $\boldsymbol{\Sigma}$. Figure 3.6 shows a sample ellipse, its eigenvectors and eigenvalues. Let λ_1 and λ_2 be the eigenvalues of $\boldsymbol{\Sigma}$ in descending order. The area of the ellipse is equal to $\pi\sqrt{\lambda_1\lambda_2}$ and the aspect ratio is equal to $\sqrt{\lambda_1/\lambda_2}$.

3.3 Graph Construction

We have tried to extract local image features thus far. As the next step, we discretize the features extracted from MSERs in order to construct a codebook. By this way, each MSER will be a visual word from the codebook. Image representation by visual words is called the *bag of words* representation [28]. However, this method ignores the relationships between visual words. Instead, we propose a graph representation which encapsulates local image features as well as the spatial information of the scene.

The definition of a labeled graph is given below and the graph construction steps are described in the following subsections.

Definition 3.2 (Labeled graph, [18])**.**
A labeled or attributed graph is a triplet $G = (V, E, \ell)$, where V is the set of vertices, $E \subseteq V \times V - \{(v, v) \,|\, v \in V\}$ is the set of edges, and $\ell : V \cup E \to \Gamma$ is a function that assigns labels from the set Γ to nodes and edges.

3.3.1 Nodes and Labels

Now, we have 17-dimensional feature vectors for each MSER in 4 different groups. These are discretized using *k-means clustering* separately for each group. We employ the *k-means++ algorithm* proposed by Arthur and Vassilvitskii in [3] owing to its better seed initialization. It can be seen in Algorithm 1. Each MSER corresponds to a graph node where its label is determined from the output of the k-means algorithm. In other words, the set of vertices V is the union of four region groups and the labeling function ℓ is a mapping from MSERs to the output of the clustering algorithm performed for every region group. The parameter of the clustering algorithm, number of clusters k, has a major effect on the performance of image classification. This effect will be discussed in Chapters 6. The algorithm is applied to each region group, so the parameter set for the number of labels is denoted by $\Upsilon = (k_{\mathsf{sd}}, k_{\mathsf{sb}}, k_{\mathsf{ud}}, k_{\mathsf{ub}})$ where the initials of the region groups are used as the indexes of the parameters. We normalize each feature to zero mean

and unit variance before applying the k-means algorithm. Cluster centers and normalization parameters are also used in the testing stage. For an unknown image, the labels of graph nodes are assigned according to the closest cluster center to the feature vector after the normalization.

Algorithm 1 `k-means++` Algorithm, [3]

Input: Set of data points, $\mathcal{X}$
Number of clusters, k
Output: Clusters of data points, C

1: Choose an initial center c_1 uniformly at random from $\mathcal{X}$.
2: Choose the next center c_i, selecting $c_i = x' \in \mathcal{X}$ with probability $\frac{D(x')^2}{\sum_{x \in \mathcal{X}} D(x)^2}$ where $D(x)$ denotes the shortest distance from a data point x to the closest center we have already chosen.
3: Repeat Step 2 until we have chosen a total of k centers.
4: Proceed as with the standard k-means algorithm.

3.3.2 Spatial Relationships and Edges

The final step of graph construction is to connect every neighboring node pair with an undirected edge. To do so, we locate the nodes in V at ellipse centers. We can determine whether given two nodes are neighbors or not by computing the Euclidean distance between the nodes and comparing it to a threshold. However, such a threshold is scale dependent [17] and cannot be automatically set for different scenes because the density of nodes in different types of scenes differs. In addition, a global threshold defined for all scene types creates more complex graphs for the images in which large number of nodes are found such as urban areas and it may produce unconnected nodes for the images with fewer number of nodes such as fields. To handle such problems we use the Voronoi tessellation where the nodes correspond to the cell centroids. The nodes whose cells are neighbors (sharing an edge) in the Voronoi tessellation are considered as neighbor nodes and are connected by undirected edges. In other words, the set of edges can be given by

$$E = \left\{(u, v) \,|\, u, v \in V \ \wedge \ u \text{ and } v \text{ are neighbors in the Voronoi diagram}\right\} \quad (3.10)$$

and the labeling function ℓ assigns the same trivial label to every edge, means we ignore edge labels.

The Voronoi tessellation successfully partitions the image region; however, some cell pairs which are not neighboring inside the image region may become neighboring outside the image region as in Figure 3.7(a). The graph constructed from this tessellation includes unnecessary edges between some outer nodes that can be seen in Figure 3.7(c). Our solution to this problem is to construct graph from whole remote sensing image and then to cut this graph into tiles (see Figures 3.7(b) and 3.7(d)).

All steps of graph construction are shown in Figure 3.8. This process is applied to every image in both training and testing stages. As a result, we produce a set of graphs which encode image content appropriately for each image and this set provides an abstraction level for new images.

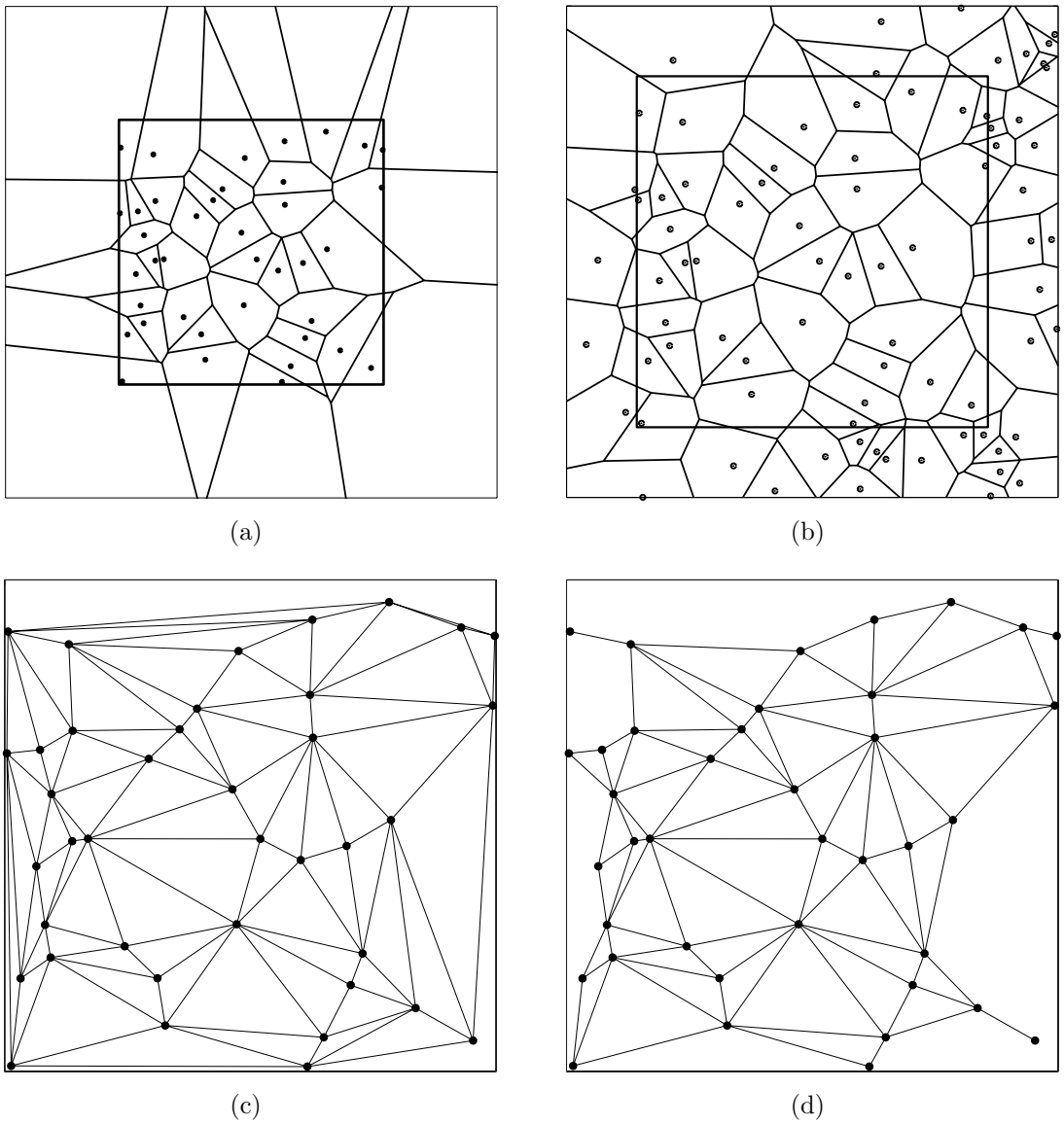
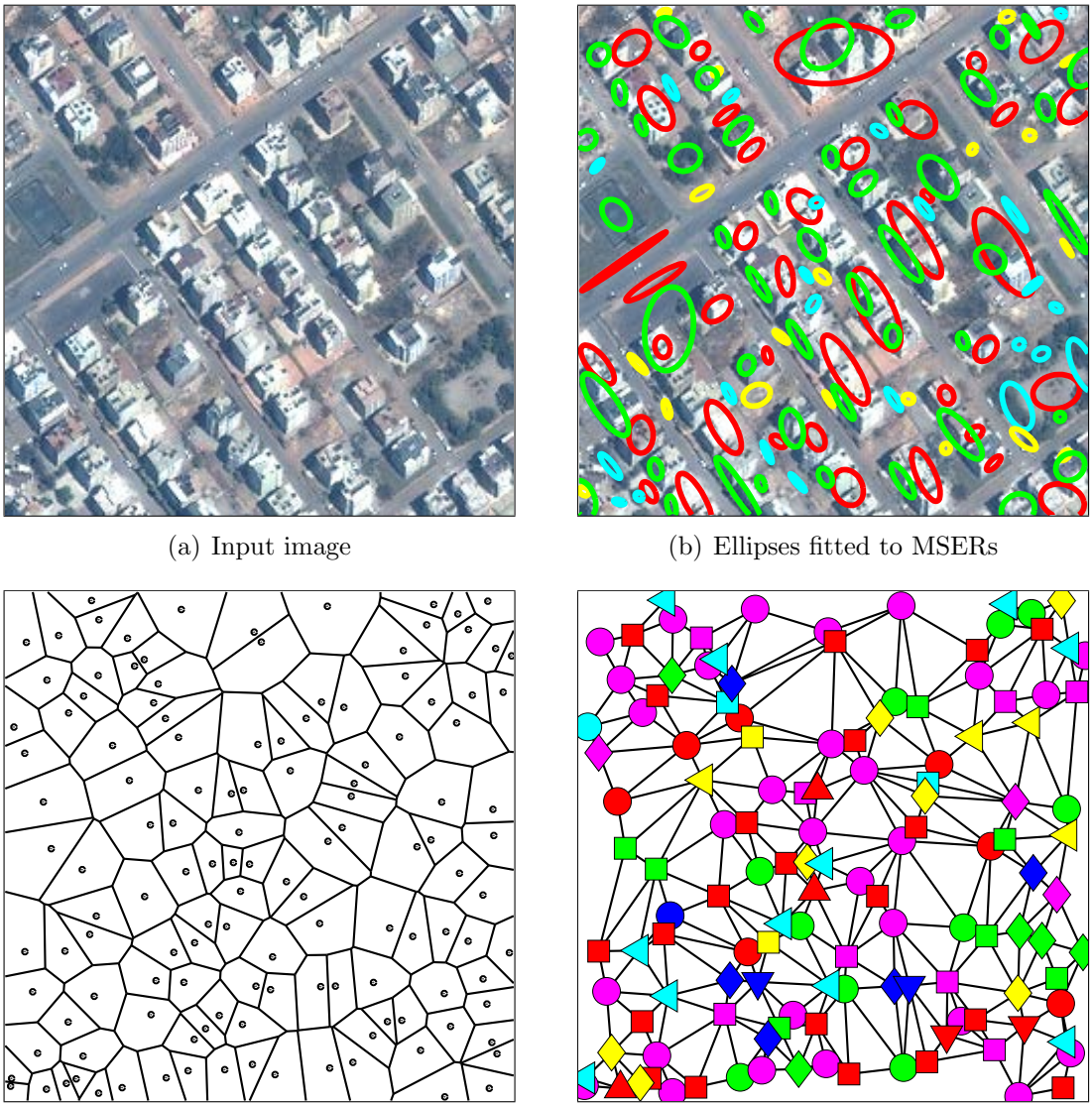


Figure 3.7: The problem of discovering neighboring node pairs in the Voronoi tessellation is shown in (a) and solution to this problem using external nodes is seen in (b). Corresponding graphs are given in (c) and (d), respectively.



(a) Input image

(b) Ellipses fitted to MSERs

(c) Voronoi Diagram

(d) Image graph

Figure 3.8: Graph construction steps. The color and shape of a node in (d) represent its label after k-means clustering.

Chapter 4

Graph Mining

11:15, restate my assumptions:

1. *Mathematics is the language of nature.*
2. *Everything around us can be represented and understood through numbers.*
3. *If you graph these numbers, patterns emerge.*

Therefore, there are patterns everywhere in nature.

Maximillian Cohen – from the movie π

At the end of previous chapter we manage to represent every image with a graph. Graphs are powerful in representing image content; however, their use for image classification has been limited due to inefficiencies in comparisons of these graphs for classification. All algorithmic tools for feature-based object representations can be available for graphs if they are embedded in vector spaces. For example, the *dissimilarity representation* [36] developed by Pekalska converts an input graph to feature vector with respect to a set of graph patterns called *prototypes*. The ith element of this vector is equal to the *graph edit distance* [37] between the input graph and the ith prototype. This method works quite well for matching relatively small graphs but it can become quite restrictive for very detailed image content with a large number of nodes and edges such as the graph in Figure 3.8(d). Furthermore, graph edit distance produces unreliable results

when the number of edit operations are too large and it is inefficient due to high computational complexity. Another graph embedding method is representing a graph as a *frequency vector* (histogram vector) for a given set of subgraphs [16]. The ith element of this vector is equal to the number of times (frequency) that the ith subgraph occurs in the input graph. The difficult part of this approach is to determine the subgraph set. For image classification, such a subgraph set should contain

1. Frequent graph patterns,
2. Discriminative graph patterns, and
3. Graph patterns having low redundancy.

The first criterion ensures that the subgraphs in the set can also be found in an unknown image graph. The second criterion guarantees the performance of classifiers, and the final criterion avoids redundancy that leads to the curse of dimensionality. To find the set satisfying these criteria, we propose a graph mining algorithm that first discovers frequent subgraphs from the image graph set, then discriminative subgraphs in the set are selected and finally redundant ones are removed from the set. We employ two methods from the literature for the first and third criteria, and develop a novel algorithm for mining discriminative patterns. The flowchart of the algorithm is displayed in Figure 4.1.

In this study we are dealing with image graphs but the subgraph-graph relation is analogous with term-document and symbol-string relations. Hence, the histogram vector method can also be extended for these relations, i.e. in the field of information retrieval. Therefore, we will use the term *pattern* to generalize subgraphs/terms/symbols in this chapter until Section 4.6. We first explain our data mining method in the following sections, then we specialize the method for graph mining.

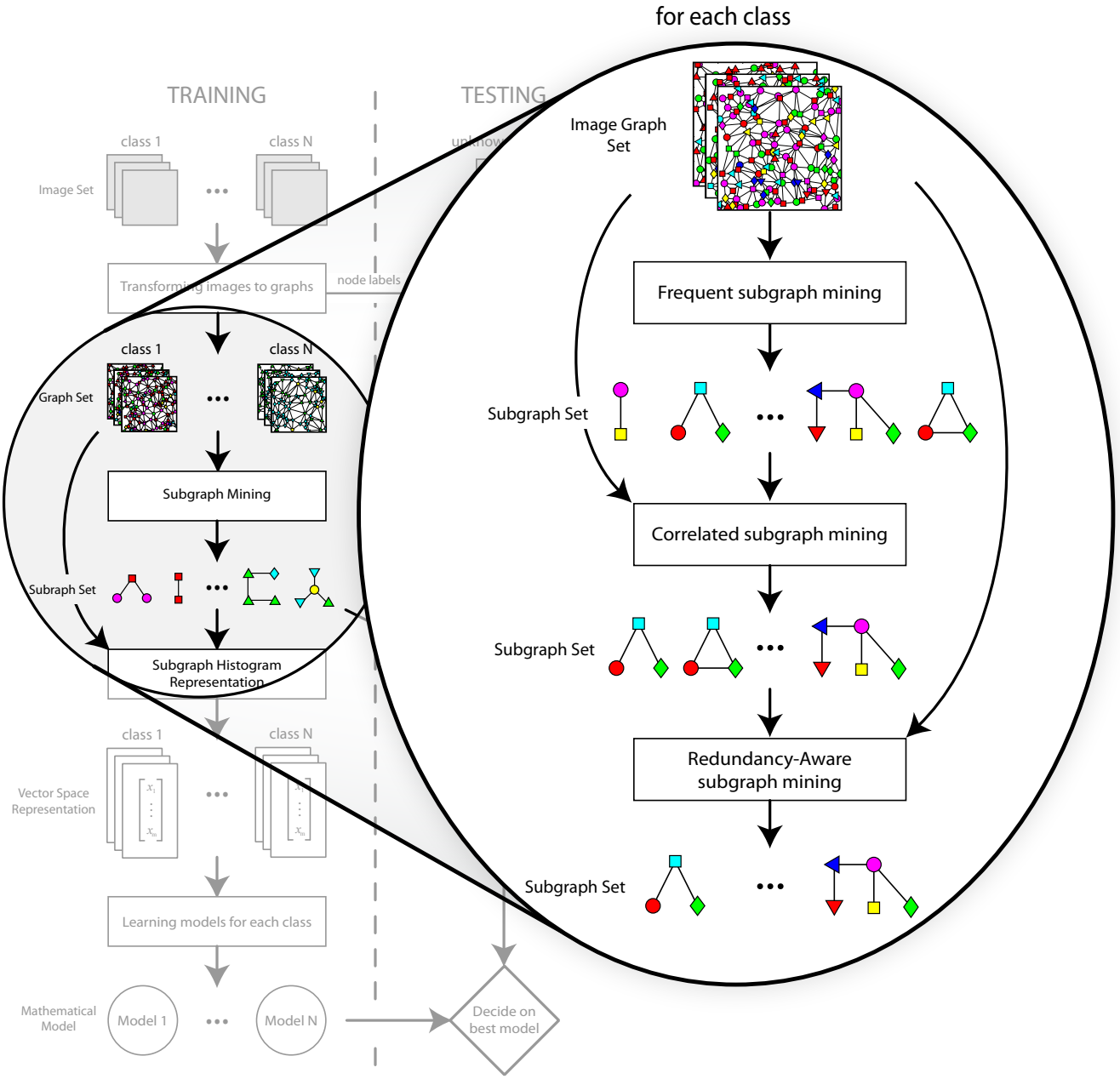


Figure 4.1: Steps of graph mining algorithm

4.1 Foundations of Pattern Mining

Before the details of the algorithm, we would like to give some background information about pattern mining. In this chapter we use a similar notation to Bringmann's in [10] and the definitions in this section are mainly taken from that study.

A definition of the task of finding all potentially interesting patterns is given by Mannila and Toivonen [30]. The result of a data mining task is defined as a theory depending on three parameters: a *pattern language* $\mathcal{L}$, a *dataset* $\mathcal{D}$, and a *selection predicate* ϕ.

Definition 4.1 (Theory of ϕ with respect to $\mathcal{L}$ and $\mathcal{D}$, [30])**.**
Assume a dataset $\mathcal{D}$, a pattern language $\mathcal{L}$ for expressing properties or defining subgroups of the data, and a selection predicate ϕ are given. The predicate ϕ is used for evaluating whether a pattern $\pi \in \mathcal{L}$ defines a potentially interesting subclass of $\mathcal{D}$. The task of finding the theory of $\mathcal{D}$ with respect to $\mathcal{L}$ and ϕ is defined as

$$Th(\mathcal{L}, \mathcal{D}, \phi) = \{\pi \in \mathcal{L} \,|\, \phi(\pi, \mathcal{D}) \text{ is true}\} \tag{4.1}$$

In our problem, the selection predicate ϕ is true if the pattern π is frequent, discriminative and not redundant for the dataset $\mathcal{D}$. We continue our definitions with the *matching function*. Many graph mining researchers define matching function as whether given subgraph occurs in example graph or not as in [10]. However, our study requires the number of times that a pattern occurs in an example (Details will be given in the following sections). Therefore, we define the matching function differently as follows.

Definition 4.2 (Matching Function)**.**
Assume a pattern language $\mathcal{L}$, a dataset $\mathcal{D}$, and an evaluation predicate φ is given. The number of *valid occurrences* of pattern π in $x \in \mathcal{D}$ is defined as $\mathsf{match} : \mathcal{L} \times \mathcal{D} \to \mathbb{Z}_+^0$ such that

$$\mathsf{match}(\pi, x, \varphi) = \left|\{h \,|\, h(\pi) \subseteq x \wedge \varphi(h, \pi, x) \text{ is true}\}\right| \tag{4.2}$$

where h is called a mapping of pattern π into example x.

We use the terms *valid occurrences* and *mapping* in this definition instead of simply saying the number of occurrences of π in x because the occurrence of graph patterns in other graphs needs additional evaluations than term or symbol patterns. We will describe some evaluation predicates for graph patterns in Section 4.6; until there omitting the parameter φ from $\mathsf{match}(\pi, x, \varphi)$ for simplicity, we have $\mathsf{match}(\pi, x)$ as an equivalent to the former.

The frequency vector described in the introductory paragraph of this chapter is called propositionalization and defined below.

Definition 4.3 (Propositionalization, [10])**.**
Given a set of n patterns $\mathcal{S} = \{\pi_1, \ldots, \pi_n\}$, we define the feature vector of an example x as

$$\overrightarrow{f_{\mathcal{S}}}(x) = \big(\mathsf{match}(\pi_1, x), \ldots, \mathsf{match}(\pi_n, x)\big)^\top. \tag{4.3}$$

Total number of valid occurrences of a pattern in a dataset is called *support* of that pattern. Again, we drop the evaluation predicate φ for the support definition.

Definition 4.4 (Support)**.**
Given a pattern language $\mathcal{L}$ and a dataset $\mathcal{D}$, support of a pattern π in $\mathcal{D}$ is defined as

$$\mathsf{supp}(\pi, \mathcal{D}) = \sum_{x \in \mathcal{D}} \mathsf{match}(\pi, x). \tag{4.4}$$

And, our last definition in this section is *frequency*.

Definition 4.5 (Frequency)**.**
Given a pattern language $\mathcal{L}$, a dataset $\mathcal{D}$, the frequency of a pattern π in $\mathcal{D}$ is defined as

$$\mathsf{freq}(\pi, \mathcal{D}) = \frac{\mathsf{supp}(\pi, \mathcal{D})}{|\mathcal{D}|}. \tag{4.5}$$

4.2 Frequent Pattern Mining

Our graph mining algorithm starts with discovering frequent patterns in the dataset. Frequent patterns have broad application areas such as association

rule mining, indexing, clustering and classification [14]. We are interested in the usefulness of frequent patterns in classification. Frequent pattern mining was extensively studied in the data mining community and numerous algorithms has been proposed in domains of different pattern types.

Assume dataset $\mathcal{D}$ is a set of examples where each example is labeled by one class in a domain of classes $\mathcal{C}$. The set of examples labeled by the class c is denoted by $\mathcal{D}_c$. In notation, we can say $\mathcal{D} = \bigcup_{i \in \mathcal{C}} \mathcal{D}_i$. The problem of frequent pattern discovery for class c can be formulated as finding all patterns generated by the pattern language $\mathcal{L}$, whose support in dataset $\mathcal{D}_c$ is grater than a threshold θ_c^{supp}. The set of all frequent patterns for class c is

$$\mathcal{F}_c = \left\{ \pi \in \mathcal{L} \,|\, \mathsf{supp}(\pi, \mathcal{D}_c) \geq \theta_c^{\mathsf{supp}} \right\}. \tag{4.6}$$

Assume we try to find frequent patterns in $\mathcal{D}_c$. Let examples labeled by class c be the set of positive examples, denoted by $\mathcal{D}_+$ and the set of all other examples labeled by other classes be the set of negative examples, denoted by $\mathcal{D}_-$. Some frequent pattern mining applications limit the support of frequent patterns in negative set. In set definition, it is given by

$$\mathcal{F}_c = \left\{ \pi \in \mathcal{L} \,|\, \mathsf{supp}(\pi, \mathcal{D}_+) \geq \theta_+^{\mathsf{supp}} \wedge \mathsf{supp}(\pi, \mathcal{D}_-) \leq \theta_-^{\mathsf{supp}} \right\}. \tag{4.7}$$

Since the size and property of a dataset varies among classes we can set different thresholds for each class. In our study, we mine the set of frequent patterns for each class, then they will be used as input to the next step which is correlated pattern mining.

4.3 Class Correlated Pattern Mining

Our second objective for the selected patterns is being discriminating for classification. Assume a dataset $\mathcal{D}$ labeled with three classes such as $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$. A discriminative pattern is expected to be found in examples of one class and not in the other classes. Such a relation between a pattern and a class is called

correlation. Two types of correlation have been defined: Positive and negative correlation. Assume we focus on finding correlated patterns of the first class. A pattern observed in only the first class, $\mathcal{D}_1$, is called positively correlated pattern with the first class or shortly *class-correlated pattern*. In contrast, a negatively correlated pattern is found in all classes except the first class (in this case $\mathcal{D}_2$ and $\mathcal{D}_3$). Both positive and negative correlation can be useful for classification. For example, an unseen example can be assigned to the first class if it includes positively correlated patterns with the first class. On the other hand, observation of negatively correlated patterns with the first class in an unseen example indicates that the example does not belong to the first class. Previous works on class-correlated patterns in [10, 33] only involve two-class datasets and the study in [34] handles multi-class correlation. However, all previous methods on correlated pattern mining base on binary matching function. Hence, more than one occurrences of a pattern in an example have the same effect with a single occurrence on correlation measure. We propose a novel technique to mine multi-class correlated patterns, in which the number of occurrences are taken into consideration. Unlike previous works, we are only dealing with positively correlated patterns and do not consider negatively correlated patterns. The following subsection explains the mathematical modeling of patterns in examples and the second subsection gives details of mining correlated patterns according to the model.

4.3.1 Mathematical Modeling of Pattern Support

We start our algorithm by deriving the probability that a pattern π occurs k times in an example. Let x be a document containing N_t terms and the probability that a randomly selected term t from x is an instance of the pattern π be a *Bernoulli distribution* with $\Pr\{T = \pi\} = p$ and $\Pr\{T \neq \pi\} = 1 - \Pr\{T = \pi\} = 1 - p$. Then, the probability that x contains π for k times is a *Binomial distribution* with

$$\Pr\{K = k\} = \binom{N_t}{k} p^k (1-p)^{N_t - k} \qquad \text{for } k = 0, 1, 2, \ldots, N_t. \tag{4.8}$$

We now define the expected number of occurrences as $\lambda = pN_t$. We generalize

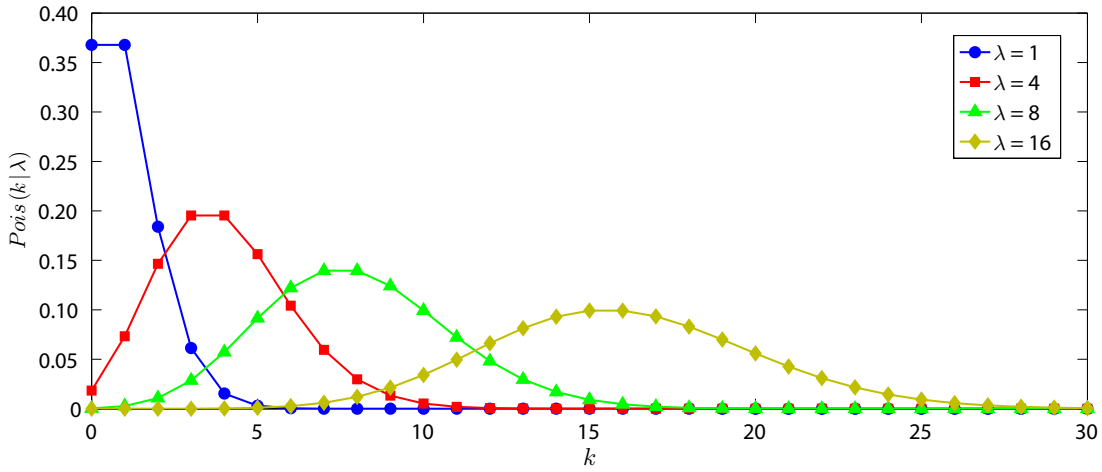


Figure 4.2: Poisson distributions with four different expected values.

this probability by assuming that the size of document is unbounded, then

$$\Pr\{K = k\} = \lim_{N_t \to \infty} \binom{N_t}{k} p^k (1-p)^{N_t - k} = \frac{\lambda^k e^{-\lambda}}{k!} \qquad \text{for } k \in \mathbb{Z}_+^0. \tag{4.9}$$

The limiting case of Binomial distribution is known as *Poisson distribution* and is denoted by $Pois(k \mid \lambda)$. Poisson distributions with different expected values are shown in Figure 4.2.

We model the term frequency in a document as a Poisson distribution. However, Church and Gale claim in [15] that term rates vary from author-to-author, topic-to-topic, document-to-document, section-to-section, and paragraph-to-paragraph (and theme-to-theme if we consider images). They propose *Poisson mixture* to capture much of this heterogeneous structure by allowing the Poisson parameter λ to vary over documents. It is subject to a density function ϑ aimed to capture dependencies on hidden variables such author, topic, etc.

Definition 4.6 (Poisson Mixtures, [15])**.**
Given the density function ϑ to capture dependencies on hidden variables, the general form of a Poisson mixture is

$$p(k) = \int_0^\infty \vartheta(\lambda) Pois(k \mid \lambda)\, \mathrm{d}\lambda \tag{4.10}$$

where ϑ density function should integrate to 1. That is, $\int_0^\infty \vartheta(\lambda)\, \mathrm{d}\lambda = 1$.

In this study we will use a mixture of finite number of Poisson distributions. Therefore, we use coefficients α_j instead of ϑ function:

$$p(k) = \sum_{j=1}^{m} \alpha_j Pois(k \mid \lambda_j) \tag{4.11}$$

where $\sum_{j=1}^{m} \alpha_j = 1$ such that $\alpha_j \geq 0$ for $j = 1, \ldots, m$.

The next step of modeling is parameter estimation for Poisson mixtures from a dataset. The *Expectation-Maximization(EM) Algorithm* can be used for the solution to the maximum-likelihood parameter estimation problem. The EM algorithm is exhaustively described in [6] for Gaussian mixture model and we will extend this study for Poisson mixture model.

We have a density function $p(k \mid \Theta)$ governed by the set of parameters $\Theta = (\alpha_1, \ldots, \alpha_m, \lambda_1, \ldots, \lambda_m)$ and a dataset of size n, supposedly drawn independently from this distribution, i.e. $\mathcal{K} = \{k_1, \ldots, k_n\}$. The resulting density for samples is

$$p(\mathcal{K} \mid \Theta) = \prod_{i=1}^{n} p(k_i \mid \Theta) = L(\Theta \mid \mathcal{K}) \tag{4.12}$$

where $L(\Theta \mid \mathcal{K})$ is called the likelihood function. The goal of EM algorithm is to find Θ^* iteratively where

$$\Theta^* = \arg\max_{\Theta} L(\Theta \mid \mathcal{K}). \tag{4.13}$$

We assume that a complete data set exists as a combination of the observed but incomplete data $\mathcal{K}$ and the missing data $\mathcal{Z}$. The EM algorithm seeks to find the maximum likelihood estimate (MLE) of the marginal likelihood by iteratively applying the following two steps:

Expectation step: The expected value of the log likelihood function is calculated with respect to the unknown data $\mathcal{Z}$ given the observed data $\mathcal{K}$ and the current parameter estimates $\Theta^{(i-1)}$.

$$Q\big(\Theta, \Theta^{(i-1)}\big) = \mathbb{E}\big[\log p(\mathcal{K}, \mathcal{Z} \mid \Theta) \,\big|\, \mathcal{K}, \Theta^{(i-1)}\big]. \tag{4.14}$$

Maximization step: The expectation can be maximized by finding optimum values for the new parameters Θ as

$$\Theta^{(i)} = \arg\max_{\Theta} Q\big(\Theta, \Theta^{(i-1)}\big). \tag{4.15}$$

We can maximize Q with respect to the two sets of parameters α_j and λ_j, independently. Fortunately, the estimates for these parameters are same as Gaussian mixture model (estimate for λ of Poisson distribution is the same with estimate for μ of Gaussian distribution).

The estimate for α_j can be computed as

$$\hat{\alpha}_j = \frac{1}{n}\sum_{i=1}^{n} p\big(j \mid k_i, \Theta^{(g)}\big) \tag{4.16}$$

where

$$p\big(j \mid k_i, \Theta^{(g)}\big) = \frac{\alpha_j^{(g)} Pois\big(k_i \mid \lambda_j^{(g)}\big)}{\sum_{t=1}^{m} \alpha_t^{(g)} Pois\big(k_i \mid \lambda_t^{(g)}\big)}. \tag{4.17}$$

Equating the partial derivative of $Q\big(\Theta, \Theta^{(g)}\big)$ with respect to λ_j to zero gives

$$\hat{\lambda}_j = \frac{\sum_{i=1}^{n} p\big(j \mid k_i, \Theta^{(g)}\big) k_i}{\sum_{i=1}^{n} p\big(j \mid k_i, \Theta^{(g)}\big)}. \tag{4.18}$$

These steps continue until the change in log-likelihood between two iterations is less than a threshold or the number of iterations reaches a limit.

Corollary 4.1 (Relationship between weighted average and sample mean)**.** After each iteration of the EM algorithm, the weighted average equals to the sample mean:

$$\sum_{j=1}^{m} \hat{\alpha}_j \hat{\lambda}_j = \frac{1}{n}\sum_{i=1}^{n} k_i. \tag{4.19}$$

Proof. Putting (4.16) and (4.18) into (4.19) gives

$$\begin{aligned} \sum_{j=1}^{m} \hat{\alpha}_j \hat{\lambda}_j &= \sum_{j=1}^{m} \left(\frac{\sum_{i=1}^{n} p\big(j \mid k_i, \Theta^{(g)}\big)}{n} \times \frac{\sum_{i=1}^{n} p\big(j \mid k_i, \Theta^{(g)}\big) k_i}{\sum_{i=1}^{n} p\big(j \mid k_i, \Theta^{(g)}\big)} \right) \\ &= \frac{1}{n}\sum_{i=1}^{n} k_i \sum_{j=1}^{m} p\big(j \mid k_i, \Theta^{(g)}\big) = \frac{1}{n}\sum_{i=1}^{n} k_i. \end{aligned} \tag{4.20}$$

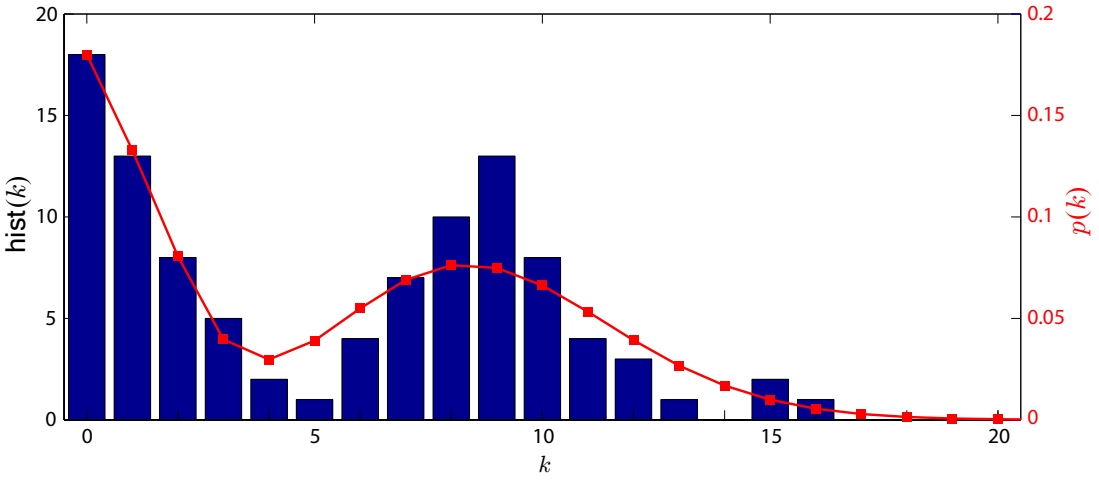


Figure 4.3: A sample histogram of a dataset with 100 elements and fitting mixtures of 3 Poisson distributions to this histogram are shown in blue and red, respectively.

This shows the equality between weighted average and sample mean. □

A sample histogram and the Poisson mixture trained on this data can be seen in Figure 4.3.

Returning to our problem, we employ Poisson mixture to model the probability of the term frequency of pattern π in a document randomly selected from the corpus $\mathcal{D}$. Thus, the dataset used in the EM algorithm is $\mathcal{K} = \{k_1, \ldots, k_n\} = \{\mathsf{match}(\pi, x_1), \ldots, \mathsf{match}(\pi, x_n)\}$ for $\mathcal{D} = \{x_1, \ldots, x_n\}$.

4.3.2 Correlated Patterns

In this section we give details of correlated-pattern mining according the probability calculations drawn in the previous section. A pattern π is called correlated with a class $c \in \mathcal{C}$ if that pattern is frequently observed in $\mathcal{D}_c$ while it is seldom seen in $\mathcal{D}_{c'}$ for all $c' \in \mathcal{C}$ such that $c' \neq c$. Using the support of a pattern in correlation measure may result in wrong classification of unknown documents because the support in a dataset does not give any frequency information of the pattern in individual documents. For example, a pattern may occur frequently

in some documents of a dataset while not occurring in other documents. Therefore, frequency analysis of a pattern should be done using the frequency in each document as in the previous section.

Assume a pattern π which does not occur in a dataset $\mathcal{D}$, in notational form $\mathsf{supp}(\pi, \mathcal{D}) = 0$, then training the EM algorithm on such a case produces the following density function p_{ref}, which we call the *reference distribution*,

$$p_{\text{ref}}(k) = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.21}$$

As understood from its name we employ this density function for comparison with other distributions in correlation measurement. Let $p_c(k \mid \pi, \mathcal{D}_c)$ denote the Poisson mixture distribution trained for pattern π on dataset $\mathcal{D}_c$. We compute the distance (dissimilarity) between density functions $p_c(k \mid \pi, \mathcal{D}_c)$ and $p_{\text{ref}}(k)$ using the *Earth mover's distance* technique.

In terms of positive correlation between pattern π and class c, the distance between $p_c(k \mid \pi, \mathcal{D}_c)$ and $p_{\text{ref}}(k)$ should be as large as possible while the distance between $p_{c'}(k \mid \pi, \mathcal{D}_{c'})$ and $p_{\text{ref}}(k)$ should be as small as possible for all $c' \in \mathcal{C}$ such that $c' \neq c$.

Earth mover's distance is proposed by [38] to measure the dissimilarity between not only probability distributions but also histograms and clusters. Levina and Bickel [27] showed that Earth mover's distance is conceptually equivalent to *Mallow distance* on probability distributions but we continue calling it the Earth mover's distance in this study.

Definition 4.7 (Earth mover's distance, [38])**.**
Let $Q = \{(q_1, w_{q_1}), \ldots, (q_r, w_{q_r})\}$ be the signature with r clusters, where q_i is the representative of the ith data cluster and w_{q_i} is the number of points in the cluster. Let $Q' = \{(q'_1, w_{q'_1}), \ldots, (q'_s, w_{q'_s})\}$ be the second signature with s clusters; and $\mathbf{D} = [d_{ij}]$ is the ground distance matrix where d_{ij} is some measure of distance between clusters of q_i and q'_j. Earth mover's distance is computed by solving the optimization problem of finding a flow $\mathbf{F} = [f_{ij}]$, with f_{ij} the flow from q_i to q'_j,

that minimizes the overall work

$$\text{Minimize} \quad \text{WORK}(Q, Q', \mathbf{F}) = \sum_{i=1}^{r}\sum_{j=1}^{s} d_{ij} f_{ij} \tag{4.22}$$

$$\text{Subject to} \quad f_{ij} \geq 0, \qquad 1 \leq i \leq r,\ 1 \leq j \leq s, \tag{4.23}$$

$$\sum_{j=1}^{s} f_{ij} \leq w_{q_i}, \qquad 1 \leq i \leq r, \tag{4.24}$$

$$\sum_{i=1}^{r} f_{ij} \leq w_{q'_j}, \qquad 1 \leq j \leq s, \tag{4.25}$$

$$\sum_{i=1}^{r}\sum_{j=1}^{s} f_{ij} = \min\left(\sum_{i=1}^{r} w_{q_i}, \sum_{j=1}^{s} w_{q'_j} \right). \tag{4.26}$$

Once the optimal flow f^*_{ij} is found, the Earth Mover's distance between Q and Q' is defined as

$$\text{EMD}(Q, Q') = \frac{\sum_{i=1}^{r}\sum_{j=1}^{s} f^*_{ij} d_{ij}}{\sum_{i=1}^{r}\sum_{j=1}^{s} f^*_{ij}}. \tag{4.27}$$

—

According to (4.24) and (4.25), we can transform a discrete distribution $p(k)$ defined in $\mathbb{Z}^0_+$ to the signature form by

$$P = \big\{ \big(k, p(k)\big) \,\big|\, k \in \mathbb{Z}^0_+ \wedge p(k) > 0 \big\}. \tag{4.28}$$

The signature size of a Poisson mixture may be infinite, or the signature may be reduced to finite elements if the condition in (4.28) is changed to $p(k) > \epsilon$ for a small number ϵ.

From (4.28), the signature form of the distribution $p_{\text{ref}}(k)$ becomes $P_{\text{ref}} = \big\{(q'_1, w_{q'_1})\big\} = \big\{(0, 1)\big\}$. Let P_c denote the signature form of probability distribution $p_c(k \,|\, \pi, \mathcal{D}_c)$, then the optimal flow from P_c to P_{ref} is

$$f^*_{ij} = \begin{cases} p_c(k_i \,|\, \pi, \mathcal{D}_c) & \text{if } j = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{4.29}$$

Constraint (4.26) is actually equal to $\sum_{i=1}^{r}\sum_{j=1}^{s} f_{ij} = 1$ for probability distributions and this constraint forces all earth to flow from every q_i to the only element of P_{ref}, which is q'_1. Finally, we need to define a ground distance function, $d : \mathbb{Z}^2 \to \mathbb{R}^0_+$, for the ground distance matrix $\mathbf{D} = \big[d_{ij}\big] = \big[d(q_i, q'_j)\big]$. Then, the Earth mover's distance is computed as

$$\begin{aligned}
\text{EMD}(P_c, P_{\text{ref}}) &= \frac{\sum_{i=1}^{r}\sum_{j=1}^{s} f^*_{ij} d_{ij}}{\sum_{i=1}^{r}\sum_{j=1}^{s} f^*_{ij}} \\
&= \sum_{i=1}^{r} p_c(k_i \,|\, \pi, \mathcal{D}_c)\, d(k_i, 0) \\
&= \sum_{i=1}^{r} p_c(k_i \,|\, \pi, \mathcal{D}_c)\, d'(k_i) \\
&= \mathbb{E}\big[d'(K) \,\big|\, \pi, \mathcal{D}_c\big]
\end{aligned} \tag{4.30}$$

where $d'(k) = d(k, 0)$ and K is a random variable $K \in \{k_1, \ldots, k_r\}$ or more generally $K \sim \mathbb{Z}^0_+$. One possible definition of the distance function for two distributions is $d(i, j) = 1 - e^{-\xi|i-j|}$ where ξ is a regulation constant. Note that $d(i, j) \in [0, 1]$, and

$$d'(i) = 1 - e^{-\xi i} \qquad \text{for } i \geq 0. \tag{4.31}$$

Accordingly, the Earth mover's distance is in the same range $[0, 1]$ due to definition of distance function. One reason why we choose this nonlinear function is to prevent outliers in the dataset from dominating the Earth mover's distance. The regulation constant ξ is used for this purpose.

Assume we try to find correlated patterns of class c. Each frequent pattern $\pi \in \mathcal{F}_c$ mined in the previous section should be tested by a correlation measure γ based on the Earth mover's distance. The correlation measure that we define has two parameters: positive distance p^c_π and negative distance n^c_π. The positive distance is the Earth mover's distance between P_c and P_{ref} for dataset $\mathcal{D}_c$. On the other hand, the negative distance is computed from datasets $\mathcal{D}_{c'}$ for all $c' \in \mathcal{C}$ such that $c' \neq c$. In terms of classification, the class which will cause confusion with class c is the one with the maximum Earth mover's distance between $P_{c'}$ and P_{ref} from all $c' \in \mathcal{C}$ such that $c' \neq c$. Thus, the positive and negative distance

used in correlation measure γ is given by

$$p_\pi^c = \mathbb{E}\big[d'(K) \,\big|\, \pi, \mathcal{D}_c\big], \qquad n_\pi^c = \max_{c' \neq c,\, c' \in \mathcal{C}} \mathbb{E}\big[d'(K) \,\big|\, \pi, \mathcal{D}_{c'}\big]. \tag{4.32}$$

The computation procedure of positive and negative distances are illustrated in Figure 4.4 for four classes. As shown in the figure, for each class the probability distribution is computed by the EM algorithm and the Earth mover's distance is computed between this distribution and the reference distribution. Then, the distance computed for the interest class c is assigned p_π^c and the greatest distance computed for the other classes is assigned to n_π^c.

We use p_π^c and n_π^c as parameters to measure the correlation between the pattern π and the class c. The correlation measures commonly used in the literatures are chi-square (χ^2) test and information gain, which are computed from a contingency table. Instead, we derive a correlation function γ from the χ^2 test as follows without indices:

$$\gamma(p, n) = \frac{(p-n)|p-n|}{(p+n)(2-p-n)}. \tag{4.33}$$

The range of γ function is $[-1, 1]$. Positive values of γ for pattern π imply positive correlation and negative values indicate negative correlation. The correlation function γ can be seen in Figure 4.5.

Similar to frequent pattern mining, correlated pattern mining is defined as finding all patterns $\pi \in \mathcal{L}$ whose correlation with $\mathcal{D}_c$ is greater than a threshold θ_c^{cor}, that is $\gamma(p_\pi^c, n_\pi^c) \geq \theta_c^{\mathsf{cor}}$. According to our objectives for pattern mining, we check correlation of only frequent patterns for each class $c \in \mathcal{C}$. To decrease computational cost we can define a lower bound for support threshold θ_c^{supp} in the previous section with respect to correlation threshold θ_c^{cor}. The rest of this section is devoted to the calculations of this relation.

Lemma 4.1 (Lower and upper bounds of γ function)**.**
The lower and upper bounds of $\gamma(p, n)$ is given by

$$\min_{p \geq p',\, n \leq n'} \gamma(p, n) = \max_{p \leq p',\, n \geq n'} \gamma(p, n) = \gamma(p', n'). \tag{4.34}$$

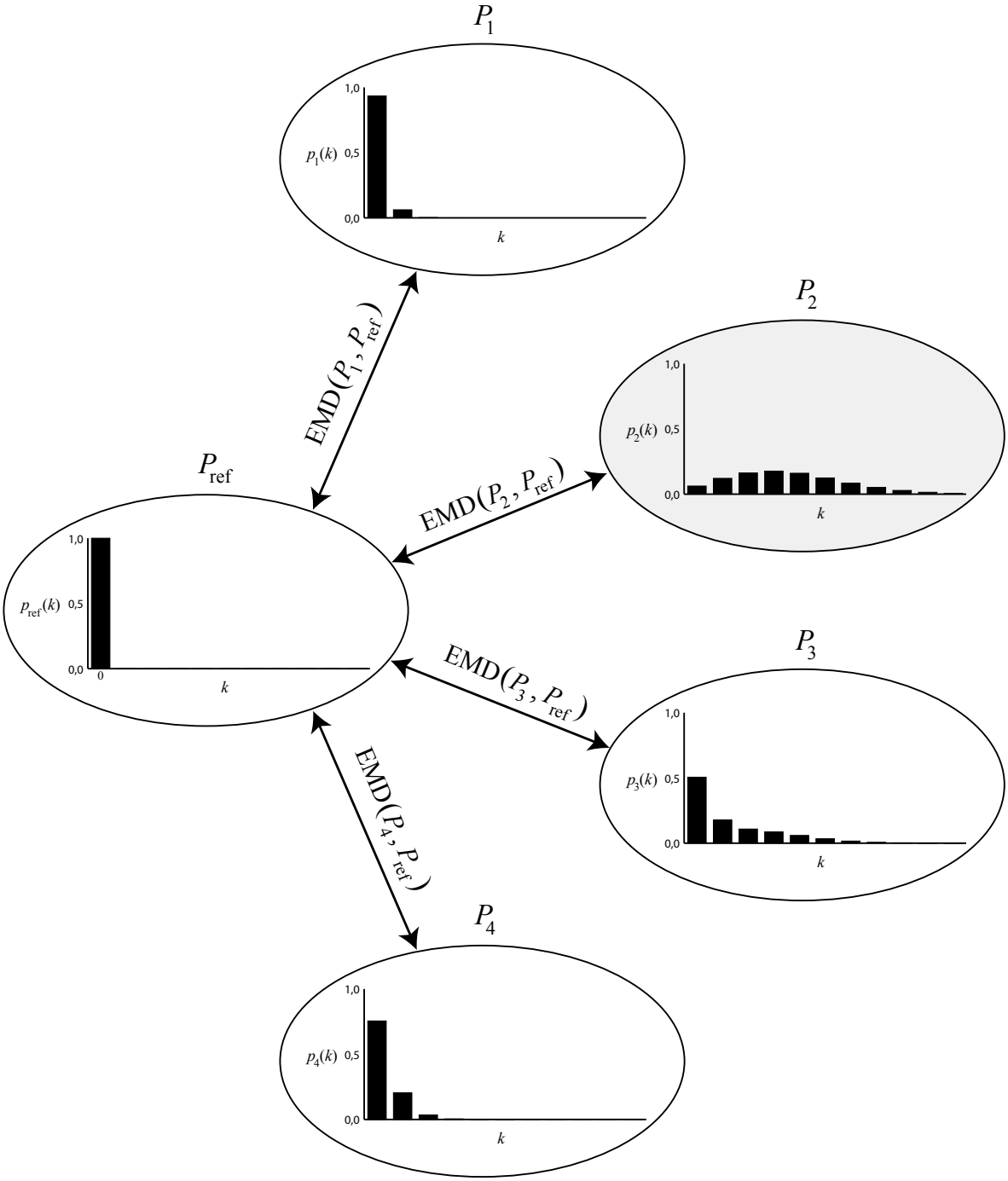


Figure 4.4: The procedure for positive and negative distance computation is illustrated for four classes. The interest class is the second one and the distances are computed as $p = \text{EMD}(P_2, P_{\text{ref}})$ and $n = \text{EMD}(P_3, P_{\text{ref}})$.

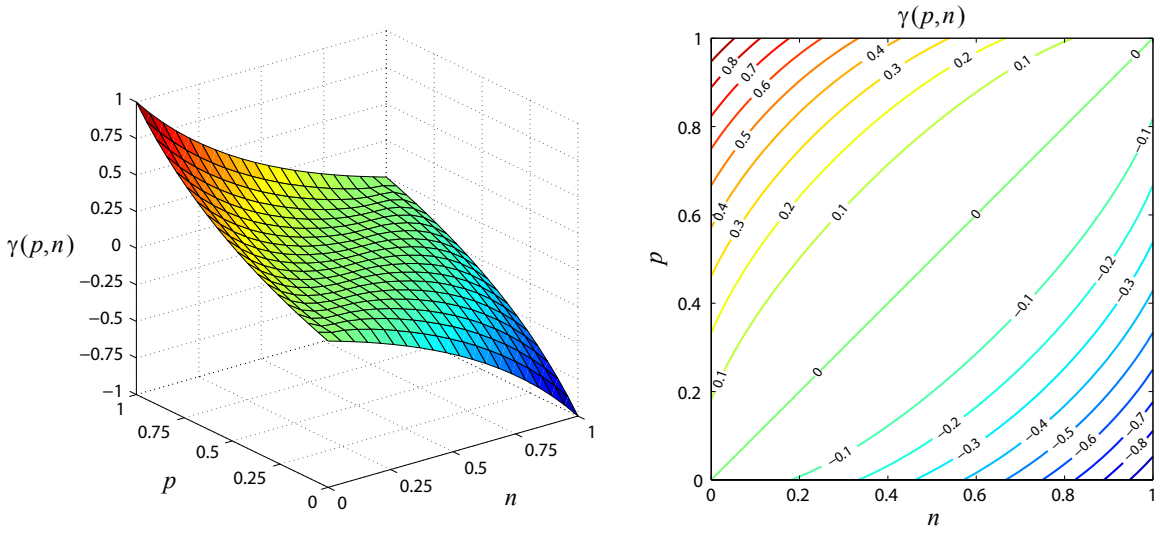


Figure 4.5: The correlation function $\gamma(p, n)$

Proof. Partial derivatives give the limits of $\gamma(p, n)$.

$$\begin{aligned}\frac{\partial\gamma(p,n)}{\partial p} &= \frac{2(p-2pn+3n-2n^2)|p-n|}{(p+n)^2(2-p-n)^2} \\ &= \frac{2\big(p(1-n)+n(3-p-2n)\big)|p-n|}{(p+n)^2(2-p-n)^2}.\end{aligned} \tag{4.35}$$

Due to definition of $d(i, j)$, the pn space is defined for $0 \le p, n \le 1$. Hence, all terms in (4.35) are nonnegative. As a result, $\frac{\partial\gamma(p,n)}{\partial p} \ge 0$ and we can say

$$\gamma(p, n) \ge \gamma(p', n) \qquad \text{for} \quad 0 \le p' \le p \le 1. \tag{4.36}$$

Similarly,

$$\begin{aligned}\frac{\partial\gamma(p,n)}{\partial n} &= -\frac{2(3p-2p^2-2pn+n)|p-n|}{(p+n)^2(2-p-n)^2} \\ &= -\frac{2\big(p(3-2p-n)+n(1-p)\big)|p-n|}{(p+n)^2(2-p-n)^2}.\end{aligned} \tag{4.37}$$

All terms in (4.37) are nonnegative. Consequently, $\frac{\partial\gamma(p,n)}{\partial n} \le 0$ and we can also say

$$\gamma(p, n) \ge \gamma(p, n') \qquad \text{for} \quad 0 \le n \le n' \le 1. \tag{4.38}$$

Combining (4.36) and (4.38) gives (4.34). □

Theorem 4.1 (Correlation–Positive support relation)**.**
For a given pattern $\pi \in \mathcal{L}$, if $\gamma(p_\pi^c, n_\pi^c) \geq \theta_c^{\mathsf{cor}}$ for some thresholds $\theta_c^{\mathsf{cor}} \geq 0$ where p_π^c and n_π^c are computed from $\mathcal{D}_c$ and $\mathcal{D}_{c'}$ for all $c' \in \mathcal{C}$ such that $c' \neq c$, respectively, then

$$\mathsf{supp}(\pi, \mathcal{D}_c) \geq -\frac{|\mathcal{D}_c|}{1 - e^{-\xi}} \ln\left(\frac{1 - \theta_c^{\mathsf{cor}}}{1 + \theta_c^{\mathsf{cor}}}\right). \tag{4.39}$$

Proof. According to Lemma 4.1, we can say

$$\gamma(p_\pi^c, 0) \geq \gamma(p_\pi^c, n_\pi^c) \geq \theta_c^{\mathsf{cor}}. \tag{4.40}$$

Putting (4.33) into (4.40),

$$\frac{p_\pi^c}{2 - p_\pi^c} \geq \theta_c^{\mathsf{cor}} \quad \Rightarrow \quad p_\pi^c \geq \frac{2\theta_c^{\mathsf{cor}}}{1 + \theta_c^{\mathsf{cor}}}. \tag{4.41}$$

Now, we compute p_π^c from (4.32). Let the parameters of Poisson mixture be $\Theta = (\alpha_1, \ldots, \alpha_m, \lambda_1, \ldots, \lambda_m)$ found by the EM algorithm for $p_c(k \,|\, \pi, \mathcal{D}_c)$ on observed data $\mathcal{K} = \big\{\mathsf{match}(\pi, x_1), \ldots, \mathsf{match}(\pi, x_n)\big\}$ where $\mathcal{D}_c = \{x_1, \ldots, x_n\}$. The positive distance is computed as

$$\begin{aligned}
p_\pi^c \;&=\; \mathbb{E}\big[d'(K) \,\big|\, \pi, \mathcal{D}_c\big] \;=\; \sum_{i=0}^{\infty} p_c(i \,|\, \pi, \mathcal{D}_c)\, d'(i) \\
&= \sum_{i=0}^{\infty} \sum_{j=1}^{m} \alpha_j Pois(i \,|\, \lambda_j)\, d'(i) \\
&= \sum_{i=0}^{\infty} \sum_{j=1}^{m} \alpha_j \frac{\lambda_j^i\, e^{-\lambda_j}}{i!} (1 - e^{-\xi i}) \\
&= \sum_{j=1}^{m} \alpha_j e^{-\lambda_j} \sum_{i=0}^{\infty} \left(\frac{\lambda_j^i}{i!} - \frac{\left(\lambda_j e^{-\xi}\right)^i}{i!}\right) \\
&= \sum_{j=1}^{m} \alpha_j e^{-\lambda_j} \Big(e^{\lambda_j} - e^{(\lambda_j e^{-\xi})}\Big) \\
&= \sum_{j=1}^{m} \Big(\alpha_j - \alpha_j e^{-\lambda_j (1 - e^{-\xi})}\Big) \;=\; 1 - \sum_{j=1}^{m} \alpha_j e^{-\lambda_j d'(1)}.
\end{aligned} \tag{4.42}$$

Introducing $\varepsilon = d'(1)$ results in $p_\pi^c = 1 - \sum_{j=1}^{m} \alpha_j e^{-\varepsilon\lambda_j}$. Combining (4.41) and (4.42) gives

$$1 - \sum_{j=1}^{m} \alpha_j e^{-\varepsilon\lambda_j} \geq \frac{2\theta_c^{\mathsf{cor}}}{1+\theta_c^{\mathsf{cor}}} \quad \Rightarrow \quad \sum_{j=1}^{m} \alpha_j e^{-\varepsilon\lambda_j} \leq \frac{1-\theta_c^{\mathsf{cor}}}{1+\theta_c^{\mathsf{cor}}}. \tag{4.43}$$

Before continuing the proof, we need to define *convex function* and *Jensen's inequality.*

Definition 4.8 (Convex function)**.**
Let C be a convex subset of $\mathbb{R}^n$. A function $f : C \to \mathbb{R}$ is called convex if

$$f\big(ax_1 + (1-a)x_2\big) \leq af(x_1) + (1-a)f(x_2) \quad \text{for } a \in [0,1] \text{ and } x_1 \neq x_2. \tag{4.44}$$

See Figure 4.6 for a sample convex function.

Theorem 4.2 (Jensen's inequality)**.**
If f is a convex function and X is a random variable, then

$$f\big(\mathbb{E}[X]\big) \leq \mathbb{E}\big[f(X)\big]. \tag{4.45}$$

In finite form, numbers $x_1, x_2, \ldots, x_n$ are in domain of f, and positive weights a_i such that $\sum_{i=1}^{n} a_i = 1$, Jensen's inequality can be stated as

$$f\left(\sum_{i=1}^{n} a_i x_i\right) \leq \sum_{i=1}^{n} a_i f(x_i). \tag{4.46}$$

—

We define function $f : \mathbb{R} \to \mathbb{R}$ such as $f(x) = e^{-\varepsilon x}$. This function is strictly convex because $\frac{d^2 f(x)}{dx^2} = \varepsilon^2 e^{-\varepsilon x} > 0$. Now, we can extend (4.43) using (4.46) and f function as follows

$$\sum_{j=1}^{m} \alpha_j f(\lambda_j) \leq \frac{1-\theta_c^{\mathsf{cor}}}{1+\theta_c^{\mathsf{cor}}} \qquad \Rightarrow \qquad f\left(\sum_{j=1}^{m} \alpha_j \lambda_j\right) \leq \frac{1-\theta_c^{\mathsf{cor}}}{1+\theta_c^{\mathsf{cor}}}. \tag{4.47}$$

Substituting $f(x)$ with $e^{-\varepsilon x}$,

$$\Rightarrow \quad \exp\left(-\varepsilon \sum_{j=1}^{m} \alpha_j \lambda_j\right) \leq \frac{1-\theta_c^{\mathsf{cor}}}{1+\theta_c^{\mathsf{cor}}} \quad \Rightarrow \quad -\varepsilon \sum_{j=1}^{m} \alpha_j \lambda_j \leq \ln\left(\frac{1-\theta_c^{\mathsf{cor}}}{1+\theta_c^{\mathsf{cor}}}\right). \tag{4.48}$$

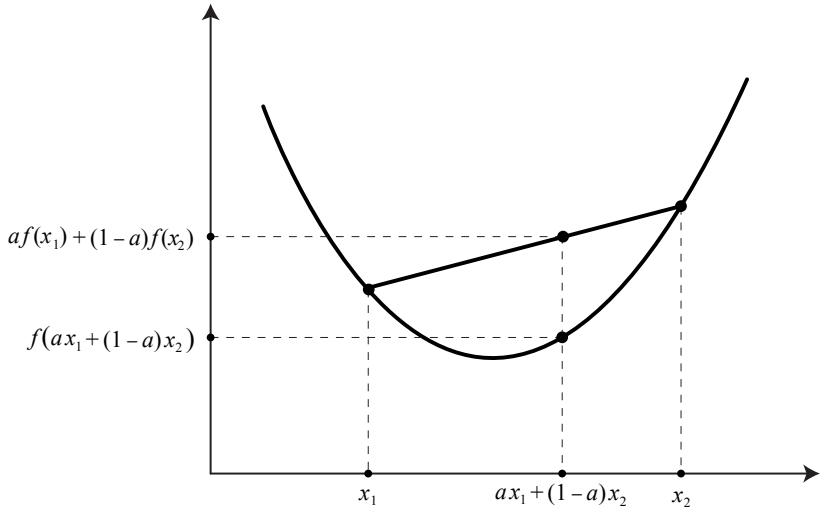


Figure 4.6: Plot of a convex function f

Using (4.19) and the definition of $\mathcal{K}$, we can say

$$\sum_{j=1}^{m} \alpha_j \lambda_j \;=\; \frac{1}{n} \sum_{i=1}^{n} k_i \;=\; \frac{1}{n} \sum_{i=1}^{n} \mathsf{match}(\pi, x_i) \;=\; \frac{1}{n}\, \mathsf{supp}(\pi, \mathcal{D}_c). \tag{4.49}$$

Finally, using (4.48) and (4.49), and substituting ε with $1 - e^{-\xi}$, and n with $|\mathcal{D}_c|$ makes

$$\mathsf{supp}(\pi, \mathcal{D}_c) \geq -\frac{|\mathcal{D}_c|}{1 - e^{-\xi}} \ln\!\left(\frac{1 - \theta_c^{\mathsf{cor}}}{1 + \theta_c^{\mathsf{cor}}}\right). \tag{4.50}$$

This completes our proof. □

The support threshold should be defined considering this relation. Likewise, the positive distance of a frequent pattern depends on the support threshold as follows.

Corollary 4.2 (Support threshold–positive distance relation)**.**
Let $\pi \in \mathcal{L}$ be a frequent pattern whose support is greater than the threshold θ_c^{supp} for dataset $\mathcal{D}_c$, then

$$p_\pi^c \geq 1 - \exp\!\left(-\frac{\theta_c^{\mathsf{supp}}}{|\mathcal{D}_c|}\left(1 - e^{-\xi}\right)\right). \tag{4.51}$$

Proof. Using same convex f function of the previous proof in (4.42), we get

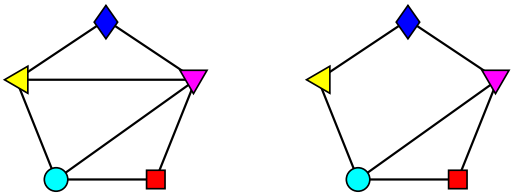
$$\begin{aligned} p_\pi^c &= 1 - \sum_{j=1}^{m} \alpha_j e^{-\lambda_j d'(1)} = 1 - \sum_{j=1}^{m} \alpha_j f(\lambda_j) \\ &\geq 1 - f\left(\sum_{j=1}^{m} \alpha_j \lambda_j \right) \\ &\geq 1 - \exp\left(-d'(1) \sum_{j=1}^{m} \alpha_j \lambda_j \right) \\ &\geq 1 - \exp\left(-\frac{d'(1)}{n} \mathsf{supp}(\pi, \mathcal{D}_c) \right) \\ &\geq 1 - \exp\left(-\frac{\left(1 - e^{-\xi}\right) \theta_c^{\mathsf{supp}}}{|\mathcal{D}_c|} \right). \end{aligned} \tag{4.52}$$

□

4.4 Redundancy-Aware Top-k Patterns

After two steps of pattern mining, we have a set of frequent and class-correlated patterns. Usage of this set for classification suffers from curse of dimensionality because the set contains many redundant patterns and its size is not limited. Let the graph patterns in Figure 4.7 be frequent and correlated patterns. As seen in the figure, the only difference between patterns is an edge. The pattern in 4.7(b) is found everywhere the pattern in 4.7(a) is found. The number of occurrences of these two patterns is very close or same in every example of a dataset. As a result, the significance of these patterns together in a set is equal to significance of only one of them. In this section we seek a set of most significant k patterns, which has low redundancy. The method described in this section is suggested by Xin *et al.* in [45]. This study gives the definitions for *pattern significance* and *pattern redundancy* as a part of formal problem formulation.

Definition 4.9 (Pattern Significance, [45])**.**
Given pattern language $\mathcal{L}$, a significance measure is a function $S : \mathcal{L} \to \mathbb{R}$, $S(\pi)$ is the degree of interestingness (or usefulness) of pattern π.



(a) (b)

Figure 4.7: Two sample redundant graph patterns

Since we focus on classification problem in this study, we choose γ function as a measure of pattern significance, so $S(\pi) = \gamma(p_\pi^c, n_\pi^c)$. Xin *et al.* extend the pattern significance to *combined significance* and *relative significance*. Let the combined significance $S(\pi, \pi')$ denote the collective significance of two individual patterns π and π'. Before defining relative significance, we need to define pattern redundancy. Given significance measures, pattern redundancy is defined as follows.

Definition 4.10 (Pattern Redundancy, [45])**.**
Given the significance measure S, the redundancy R between two patterns π and π' is defined as $R(\pi, \pi') = S(\pi) + S(\pi') - S(\pi, \pi')$.

Subsequently, the relative significance of π given π' is $S(\pi \mid \pi') = S(\pi) - R(\pi, \pi')$. Assuming that combined significance is not less than the significance of any individual pattern and not grater than the sum of two individual significance; then the redundancy between patterns π and π' should satisfy

$$0 \leq R(\pi, \pi') \leq \min\big(S(\pi), S(\pi')\big). \tag{4.53}$$

According to [45], the ideal redundancy measure $R(\pi, \pi')$ is usually hard to obtain. Therefore, they suggest using distance between patterns in order to measure patterns redundancy.

Definition 4.11 (Pattern Distance, [45])**.**
A distance measure $D : \mathcal{L} \times \mathcal{L} \to [0, 1]$ is a mapping from two patterns $\pi, \pi' \in \mathcal{L}$ to a value in $[0, 1]$, where 0 means π and π' are completely relevant and 1 means π and π' are totally independent.

The pattern distance is used for approximating the pattern redundancy. The following equation which satisfies (4.53) is given in [45].

$$R(\pi, \pi') = \big(1 - D(\pi, \pi')\big) \times \min\big(S(\pi), S(\pi')\big). \tag{4.54}$$

The distance between two patterns depends on the pattern structure. It can be the edit distance for sequences or the graph edit distance [37] for graph patterns. Alternatively, in the distance function we use the number of occurrences of patterns in a dataset because it provides the correlation between two patterns in document level. We define the distance function using *cosine similarity* in the field of information retrieval as follows.

$$D(\pi, \pi') = 1 - \cos_{\mathcal{D}}(\pi, \pi') = 1 - \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \, \|\mathbf{v}\|} \tag{4.55}$$

where $\mathbf{u} = \overrightarrow{f_{\mathcal{D}}}(\pi)$ and $\mathbf{v} = \overrightarrow{f_{\mathcal{D}}}(\pi')$. We use the whole dataset $\mathcal{D} = \bigcup_{i \in \mathcal{C}} \mathcal{D}_i$, in distance computation where $\overrightarrow{f_{\mathcal{D}}}$ is defined as

$$\overrightarrow{f_{\mathcal{D}}}(\pi) = \big(\mathsf{match}(\pi, x_1), \ldots, \mathsf{match}(\pi, x_n)\big)^{\top} \tag{4.56}$$

where $\mathcal{D} = \{x_1, \ldots, x_n\}$. By the help of this distance function, we try to find a set of patterns which occur in not only some examples of the dataset $\mathcal{D}_c$ in which we are interested, but in all examples of $\mathcal{D}_c$ if possible.

Finally, the formulation is extended to a set of k patterns $\mathcal{P}^k \subset \mathcal{L}$. Let G be a significance measure for a set of patterns, and the redundancy measure for a set of patterns be L which is hard to obtain [45]. In general,

$$G_{gen}(\mathcal{P}^k) = \sum_{i=1}^{k} S(\pi_i) - L(\mathcal{P}^k). \tag{4.57}$$

The authors of [45] suggest two heuristic evaluation functions G_{as} (average significance) and G_{ms} (marginal significance). We prefer using marginal significance so we only explain this function. The computational model for the new concept is a graph called *redundancy graph*:

Definition 4.12 (Redundancy Graph, [45])**.**
Given a significance measure S and a redundancy measure R for individual patterns, a redundancy graph of a set of k patterns $\mathcal{P}^k$ is a weighted graph where

each pattern, π_i, corresponds to node i whose weight is pattern significance $S(\pi_i)$ and the weight on an edge (i, j) is the redundancy $R(\pi_i, \pi_j)$.

Marginal significance of a set of patterns is computed as

$$G_{ms}(\mathcal{P}^k) = \sum_{i=1}^{k} S(\pi_i) - w\big(MST(\mathcal{P}^k)\big) \tag{4.58}$$

where $w\big(MST(\mathcal{P}^k)\big)$ denotes the sum of edge weights on the *maximum spanning tree* of the redundancy graph. Given a pattern language $\mathcal{L}$, the problem of maximal marginal significance (MMS) is to find a set of k patterns $\mathcal{P}^k$ such that $G_{ms}(\mathcal{P}^k)$ is maximized. Finally, the study [45] gives Algorithm 2 for the problem of MMS.

According to our definitions of pattern significance and distance, we obtain a redundancy-aware set of frequent top-k patterns in terms of correlation. We denote this set by $\mathcal{S}_c$ for class $c \in \mathcal{C}$ and we apply these mining steps for each class. Then, all selected patterns are collected in one set $\mathcal{S}$ which will be used for classification. This completes the process of finding a set of patterns satisfying our objectives given in the beginning of this chapter.

4.5 Summary of the Mining Algorithm

The procedure so far is aiming to find a set of patterns satisfying our objectives given at the beginning of this chapter. Each step can be considered as a pattern filter which is responsible for one criterion. To reduce the computational cost, the filters are connected in series, in other words the input patterns of a filter is the output of another. The order of filters are same with the section order. First, the patterns generated by the language $\mathcal{L}$ are tested for support with respect to the threshold θ_c^{supp} and the frequent ones are collected in the set $\mathcal{F}_c$. Next, the members of this set are tested for the correlation with respect to the threshold θ_c^{cor} and the set $\mathcal{R}_c$ contains patterns which are both frequent and correlated. The last filter seeks a small subset $\mathcal{P}_c$ which has significant and not redundant

Algorithm 2 Greedy Algorithm for MMS, [45]

Input: A set of n patterns, $\mathcal{L} = \{\pi_1, \ldots, \pi_n\}$
Number of output patterns, k
Significance measure, S
Divergence measure, D
Output: Top-k pattern set, $\mathcal{P}^k$

1: $t \leftarrow 0$, $T \leftarrow \max_{\pi \in \mathcal{L}} S(\pi)$
2: $\mathsf{selected}[i] \leftarrow \mathit{false}$ for $i = 1, \ldots, n$
3: $\mathsf{removed}[i] \leftarrow \mathit{false}$ for $i = 1, \ldots, n$
4: **for** $i \leftarrow 1$ to k **do**
5: **if** there is no pattern left **then**
6: $T \leftarrow \frac{T+t}{2}$
7: **goto** line 2
8: **end if**
9: $\pi_s \leftarrow$ the most significant pattern s.t.
$\mathsf{selected}[s] = \mathit{false}$ and $\mathsf{removed}[s] = \mathit{false}$
10: $\mathsf{selected}[s] \leftarrow \mathit{true}$
11: $\mathsf{removed}[s] \leftarrow \mathit{true}$
12: **for** $j \leftarrow 1$ to n **do**
13: **if** $\mathsf{removed}[j] = \mathit{false}$ and $\mathsf{selected}[j] = \mathit{false}$ **then**
14: **if** $S(\pi_j \mid \pi_s) \leq \frac{T+t}{2}$ **then**
15: $\mathsf{removed}[j] \leftarrow \mathit{true}$
16: **end if**
17: **end if**
18: **end for**
19: **end for**
20: **if** there are patterns left (π_i s.t. $\mathsf{removed}[i] = \mathit{false}$) **then**
21: $t = \frac{t+T}{2}$
22: **goto** line 2
23: **end if**
24: **return** selected k-patterns (π_i s.t. $\mathsf{selected}[i] = \mathit{true}$)

patterns. These steps are repeated for each class $c \in \mathcal{C}$ and then the patterns satisfying these criteria for some classes are grouped into the final set $\mathcal{S}$ which is used for classification. Algorithm 3 shows the steps of the mining algorithm.

Algorithm 3 Pattern Mining Algorithm

Input: A pattern language, $\mathcal{L}$
A dataset $\mathcal{D}$ labeled with classes $\mathcal{C}$
Support thresholds, θ_c^{supp} for each $c \in \mathcal{C}$
Correlation thresholds, θ_c^{cor} for each $c \in \mathcal{C}$
Number of top patterns for each class, k
An evaluation predicate, φ
Output: A pattern set, $\mathcal{S}$

$\mathcal{S} \leftarrow \emptyset$
for each class $c \in \mathcal{C}$ **do**
  $\mathcal{F}_c \leftarrow \emptyset$
  for each pattern π generated by $\mathcal{L}$ **do**
    if $\mathsf{supp}(\pi, \mathcal{D}_c, \varphi) \geq \theta_c^{\mathsf{supp}}$ **then** {supp defined in Definition 4.4}
      $\mathcal{F}_c \leftarrow \mathcal{F}_c \cup \{\pi\}$
    end if
  end for
  $\mathcal{R}_c \leftarrow \emptyset$
  for each pattern $\pi \in \mathcal{F}_c$ **do**
    compute p_π^c and n_π^c on $\mathcal{D}$ {defined in (4.32)}
    if $\gamma(p_\pi^c, n_\pi^c) \geq \theta_c^{\mathsf{cor}}$ **then** {γ defined in (4.33)}
      $\mathcal{R}_c \leftarrow \mathcal{R}_c \cup \{\pi\}$
    end if
  end for
  for each pattern $\pi \in \mathcal{R}_c$ **do**
    compute pattern significance as $\mathbf{S}[\pi] = \gamma(p_\pi^c, n_\pi^c)$
    for each pattern $\pi' \in \mathcal{R}_c$ **do**
      compute pattern distances $\mathbf{D}[\pi, \pi'] = 1 - \cos_{\mathcal{D}}(\pi, \pi')$ {defined in (4.55)}
    end for
  end for
  $\mathcal{P}_c \leftarrow$ output of Greedy Algorithm for MMS with $\mathcal{R}_c$, k, $\mathbf{S}$ and $\mathbf{D}$
  $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{P}_c$
end for
return $\mathcal{S}$

We can analyze search regions of patterns in the *pn* space. Assume that the support threshold θ_c^{supp} is chosen greater than the lower bound given in Theorem 4.1. The members of $\mathcal{F}_c$ are found in the shaded area (union of dark and

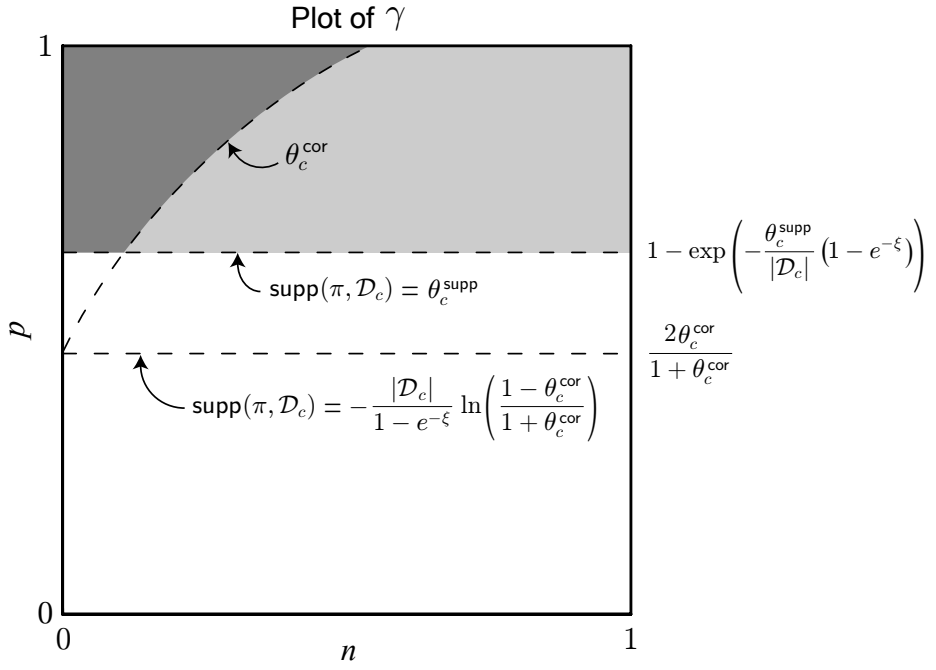


Figure 4.8: The *pn* space showing the search regions for the first two steps of the algorithm. The shaded area (union of dark and light gray) represent the domain region of $\mathcal{F}_c$ and dark gray area represents the domain region of $\mathcal{R}_c$.

light gray) in Figure 4.5 according to Corollary 4.2. After the correlation check, the region of $\mathcal{R}_c$ is reduced to the dark gray area in the figure. The final step does not reduce the search region in the *pn* space because set $\mathcal{P}_c$ may contain a pattern π on the correlation boundary $S(\pi) = \theta_c^{\mathsf{cor}}$ because of the redundancy definition.

Someone may claim that changing the order of the first two steps causes the computational cost to reduce. However, it is not sensible most of the time because the support test has negligible computation cost compared to the correlation test which includes the execution of the EM algorithm. Furthermore, a structural language which has a generalization relation among its patterns enables us to define an *anti-monotonic matching function*. Such a function provides the pruning ability in generating frequent patterns of the language using the *pattern growth approach*. For example, many frequent graph mining algorithms generate subgraphs from the dataset. They start generating subgraphs from one node subgraphs and continue generating new subgraphs by adding nodes and edges to

the previous ones. In generating process, the subgraphs whose support in the dataset is less than the threshold are eliminated and only the frequent ones are preserved. The algorithm stops when no new subgraph can be generated from the frequent subgraphs.

To recall the theory of ϕ in Section 4.1, the mining algorithm also tries to find the theory of a selection predicate. Now, we can define the selection predicate ϕ to find the theory with respect to the language $\mathcal{L}$ and the dataset $\mathcal{D}$ as follows.

$$\phi(\pi, \mathcal{D}) = \begin{cases} \mathit{true} & \text{if } \exists c \in \mathcal{C} : \mathsf{supp}(\pi, \mathcal{D}_c) \geq \theta_c^{\mathsf{supp}} \ \wedge \ \gamma(p_\pi^c, n_\pi^c) \geq \theta_c^{\mathsf{cor}} \ \wedge \ \pi \in \mathcal{P}_c \\ \mathit{false} & \text{otherwise.} \end{cases} \tag{4.59}$$

4.6 Graph Patterns

In this section, we narrow down the theoretical concepts in the previous sections to graph patterns. All computations in Sections 4.2 to 4.4 are based on definitions in Section 4.1 and the abstraction of these definitions for all pattern types is based on the evaluation predicate φ. We will define alternative φ predicates for graph patterns. Therefore, we start to use the following graph terminology:

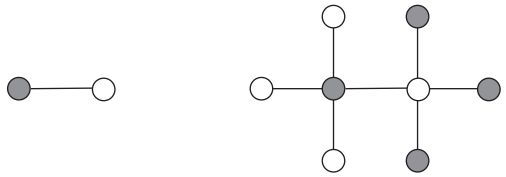
pattern $\pi \ \longrightarrow$ subgraph g

example $x \ \longrightarrow$ image (input) graph G

dataset $\mathcal{D} \ \longrightarrow$ graph set $\mathcal{G}$

Accordingly, a mapping h of pattern π into example x, given in the matching function definition is called *subgraph isomorphism* in the graph terminology. The definition is given by Fiedler and Borgelt as follows.

Definition 4.13 (Subgraph isomorphism, [18])**.**
Let $g = (V_g, E_g, \ell_g)$ and $G = (V_G, E_G, \ell_G)$ be two labeled graphs. A subgraph isomorphism of g to G is an injective function $h : V_g \to V_G$ satisfying $\forall v \in V_g : \ell_g(v) = \ell_G\big(h(v)\big)$ and $\forall (u, v) \in E_g : \big(h(u), h(v)\big) \in E_G \ \wedge$



(a) subgraph (b) input graph

Figure 4.9: An example for overlapping embeddings

$\ell_g\big((u,v)\big) = \ell_G\big(\big(f(u), f(v)\big)\big)$.

Every subgraph isomorphism of subgraph g to image graph G defines an *embedding* of subgraph g. Different embeddings of g may refer to same nodes in G as in Figure 4.9. They are called *overlapping subgraph isomorphism.*

Definition 4.14 (Overlapping subgraph isomorphism, [18])**.**
Let $g = (V_g, E_g, \ell_g)$ and $G = (V_G, E_G, \ell_G)$ be two labeled graphs, h_1 and h_2 two subgraph isomorphisms of g to G, and $V_i = \big\{v \in V_G \,\big|\, \exists u \in V_g : v = h_i(u)\big\}$, $E_i = \big\{e \in E_G \,\big|\, \exists (u,v) \in E_g : e = \big(h_i(u), h_i(v)\big)\big\}$ for $i = 1$ or 2. Two subgraph isomorphisms h_1 and h_2 are called overlapping iff $V_1 \cap V_2 \neq \emptyset$ and written $h_1 \circledcirc h_2$. Also, h_1 and h_2 are called equivalent, written $h_1 \circ h_2$, iff $V_1 = V_2$ and $E_1 = E_2$. Finally, h_1 and h_2 are called identical, written $h_1 \equiv h_2$, iff $\forall v \in V_g : h_1(v) = h_2(v)$.

Two identical subgraph isomorphisms are treated as only one embedding because they refer to the same nodes. However, two equivalent subgraph isomorphisms may not be identical if the subgraph contains the same label for more than one node; for example, subgraph A-B-A and input graph A-B-A-C. We define the set of all embeddings which are not identical as below:

$$\mathcal{H}(g,G) = \big\{h \,|\, h(g) \subseteq G \wedge \forall h' \in \mathcal{H}(g,G) : h \neq h' \rightarrow \neg(h \equiv h')\big\}. \tag{4.60}$$

The simplest evaluation predicate is defined by

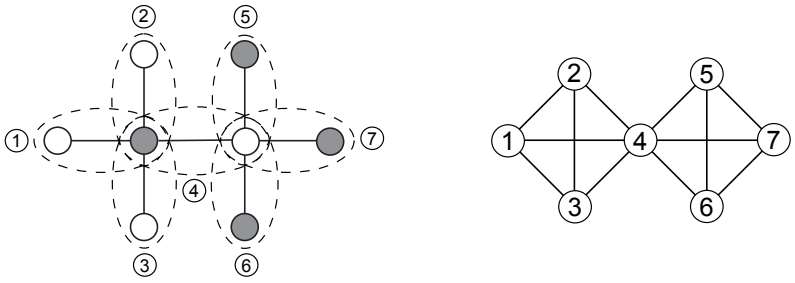
$$\varphi_{all}(h,g,G) = \begin{cases} \mathit{true} & \text{if } h \in \mathcal{H}(g,G) \\ \mathit{false} & \text{otherwise.} \end{cases} \tag{4.61}$$

Using this definition for the subgraph g in Figure 4.9(a) and the input graph G in Figure 4.9(b), the result of match function is $\mathsf{match}(g, G, \varphi_{all}) = 7$. Let G' be an image graph containing 7 instances of g independently. The matching function using the predicate φ_{all} produces the same result for both cases. However, a desirable matching function should produce a greater value for the input graph G'. This example shows the importance of handling overlapping embeddings. Fortunately, more sophisticated methods have been proposed in the graph mining literature, that we will describe later.

Most of the graph mining methods use binary matching function which only checks whether input graph contains an instance of given subgraph, or not. This function can be obtained by an evaluation predicate, φ_{bin}, which returns true if just one single embedding exists. However, we do not prefer binary matching function because the image graphs we used in the experiments may contain a structure belonging to another class, i.e. a single house in an instance of forest image.

A method called the *maximum independent set (MIS) support* is proposed by Kuramochi and Karypis for handling overlapping embeddings [25]. They introduce the *overlap graph* for computing this support measure where each embedding corresponds to a node and an edge is inserted between two nodes if they are overlapping. Every embedding of the subgraph in Figure 4.9(a) is shown in Figure 4.10(a) and the corresponding overlap graph can be seen in Figure 4.10(b). The maximum independent set is found by removing minimum number of nodes from this graph, that makes the remaining nodes independent (unconnected) an the MIS support in a single graph is computed as the number of remaining nodes. The maximum independent set of the overlap graph in Figure 4.10(b) has two nodes: one node from the set $\{1, 2, 3\}$ and one node from the set $\{5, 6, 7\}$. We formally define this support measure as follows.

Definition 4.15 (Maximum independent set of embeddings)**.**
Given a subgraph g, an input graph G, and an independence predicate ω which takes a set of embeddings as an argument and returns true if its members are not



(a) (b)

Figure 4.10: In (a) The embeddings of the subgraph in Figure 4.9(a); in (b) the corresponding overlap graph.

overlapping. The maximum independent set (MIS) of embeddings is defined as

$$H_{\text{MIS}}(g, G) = \underset{\substack{H \subset \mathcal{H}(g,G), \\ w(H) \text{ is true}}}{\arg\max} |H| \tag{4.62}$$

The evaluation predicate of the MIS support is given by

$$\varphi_{\text{MIS}}(h, g, G) = \begin{cases} \textit{true} & \text{if } h \in H_{\text{MIS}}(g, G) \\ \textit{false} & \text{otherwise.} \end{cases} \tag{4.63}$$

According to this predicate, the result of the matching function for Figure 4.9 is only 2, which is more realistic. The evaluation predicate which we use for graph mining is an extension of φ_{MIS}. It also evaluates the subgraph $g = (V_g, E_g, \ell_g)$ in terms of graph size with the predefined size limits: minimum number of nodes $v_{\min}$ and maximum number of nodes $v_{\max}$. We define the new evaluation predicate as

$$\varphi(h, g, G) = \begin{cases} \textit{true} & \text{if } v_{\max} \geq |V_g| \geq v_{\min} \ \wedge \ \varphi_{\text{MIS}}(h, g, G) \text{ is true} \\ \textit{false} & \text{otherwise.} \end{cases} \tag{4.64}$$

This concludes our discussion of graph mining. Algorithm 3 summarizes the graph mining procedure with the evaluation predicate φ. Sample subgraphs mined by the algorithm using the predicate φ, and the embeddings of these subgraphs in the image graphs are given in Figure 4.11 for three different classes. The effects of parameters, thresholds and the computational complexity will be discussed in Chapter 6.

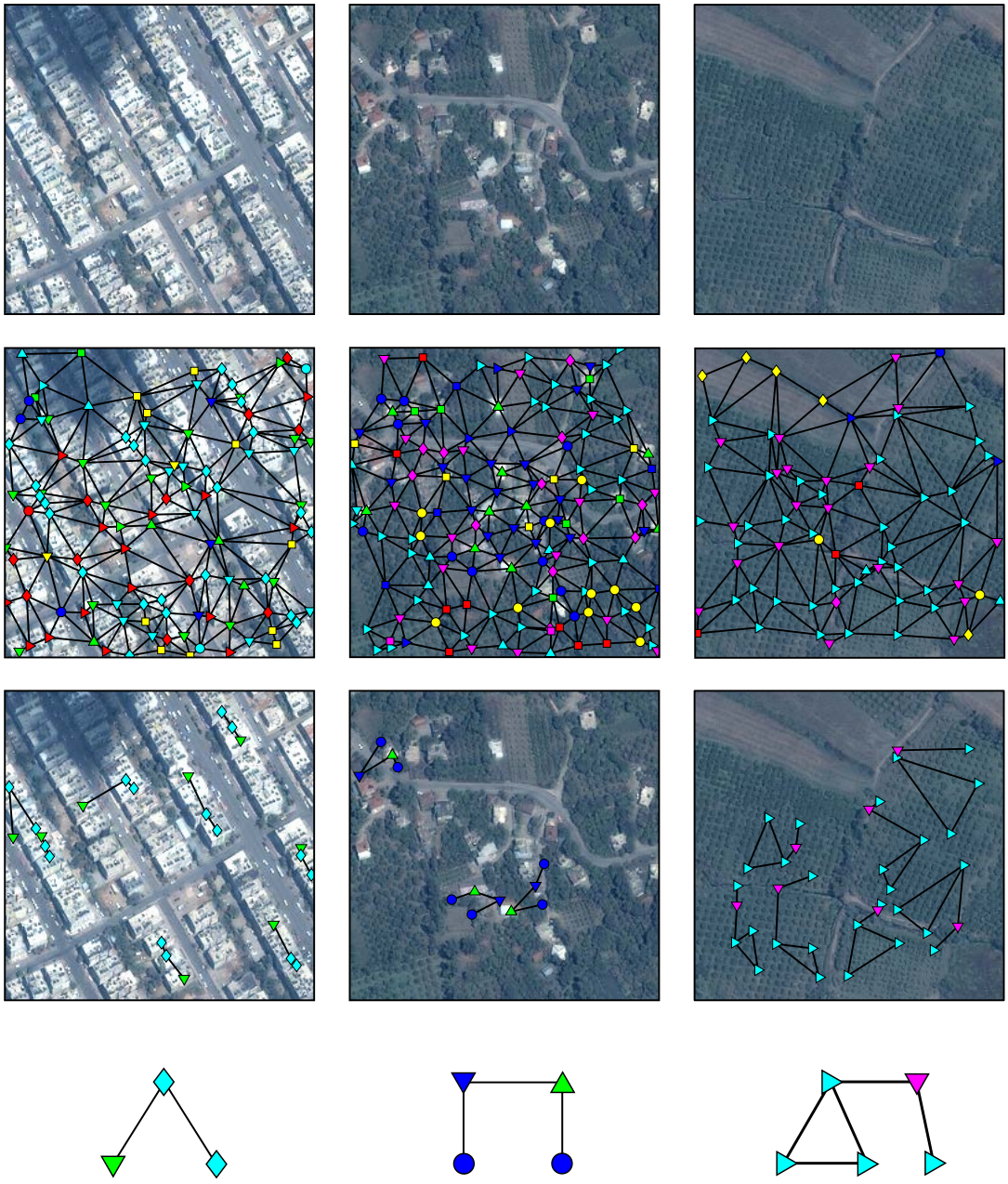


Figure 4.11: Images from top to down are original images from three different classes, image graphs for 36 labels, embeddings of sample subgraphs found by the mining algorithm and the sample subgraphs where the color and shape of a node represents its label.

Chapter 5

Scene Classification

All models are wrong, but some are useful.

George E. P. Box

After finding the set of subgraphs $\mathcal{S}$, we represent every image graph as a subgraph histogram vector. These histogram vectors are used for model learning for each class. Finally, the support vector machine (SVM) using these models decides the best model for an unseen image. Another learning model which we use is Latent Dirichlet Allocation (LDA). Given a subgraph histogram of an image, LDA provides further representation of the image based on the theme distribution in the image. This representation enables classification of images according to their theme distributions and helps us to identify heterogeneous image content. In the following sections we describe the image representation, classification of images using SVM and theme discovery in the images using LDA.

5.1 Subgraph Histogram Representation

The subgraph histogram provides a powerful representation that is not as complex as full graph models, and reduces the complexity of graph similarity computation. The histogram is constructed using the support of each subgraph in the set $\mathcal{S}$

selected by graph mining. Each image graph G in the graph set $\mathcal{G}$ is transformed into a histogram feature vector

$$\mathbf{x} = (x_1, \ldots, x_n)^\top \tag{5.1}$$

where $x_i = \mathsf{match}(g_i, G, \varphi)$ and $g_i \in \mathcal{S}$ for $i = 1, \ldots, n$. By this way, images can be classified in this feature space using statistical pattern recognition techniques.

5.2 Support Vector Machines

Support vector machines (SVM) are popular data classification technique. In this section we give a brief description of the SVM and discuss only the parameters; more details can be found in [13]. Given a training dataset with class labels for each instance, SVM maps training vectors $\mathbf{x}_i$ to a higher dimensional space. The goal is to find a linear separating hyperplane with the maximal margin in this higher dimensional space. We use a multi-class support vector machine with a radial basis function kernel (RBF) for image classification. The kernel function is

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2} \qquad \text{for } \gamma > 0. \tag{5.2}$$

The multi-class SVM is a combination of one-against-one class SVMs where the output class is the one with the maximum number of votes [13].

We have only two SVM parameters to decide: The error parameter C and the kernel parameter γ. A *grid search* on parameters C and γ is recommended by libSVM [13] using cross-validation. Various values for the pair (C,γ) are tried and the pair with the best cross-validation accuracy on training data is selected. The values tried are selected from exponentially growing sequences to identify good parameters. Additionally, we normalize each feature to the range [0,1] before training the SVM.

5.3 Latent Dirichlet Allocation

The dataset used in the experiments consists of homogeneous tiles cut from the Antalya image shown in Figure 1.2 where the tile size is 250×250 pixels. We have approximately 950 tiles excluding the sea tiles when we cut the Antalya image; however, the dataset contains only 585 tiles. Almost 40 percent of land tiles are not selected to the dataset due to heterogeneous content (or unclassified content). Further classification of these tiles enables partitioning of the whole satellite image into semantic class regions. For this purpose, we employ a generative probabilistic model for theme discovery in these images.

Latent Dirichlet allocation (LDA) introduced by Blei *et al.* in is a generative probabilistic model for collections of discrete data such as text corpora [7]. The basic idea of LDA is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. LDA, which is originally developed from text modeling, is easily adapted to our graph data by making analogies between document–image graph, word–subgraph, corpus–image graph set and topic–theme. Likewise, LDA has been adapted to computer vision by drawing an analogy between words and image patches in [28]. LDA is defined using the following terms:

- A subgraph is the basic unit of an image graph, defined to be an item from a graph set indexed by $\{1, \ldots, T\}$. The vth subgraph is represented by a T-vector g such that $g^v = 1$ and $g^u = 0$ for $u \neq v$.
- An image graph is a set of N subgraphs indexed by $G = \{g_1, \ldots, g_N\}$, where g_i is the ith subgraph in the set.
- An image graph set is a collection of M image graphs denoted by $\mathcal{G} = \{G_1, \ldots, G_M\}$.

LDA assumes the following generative process for each image graph G in an image graph set $\mathcal{G}$:

1. For each image graph, choose $\delta \sim \text{Dir}(\eta)$ where δ is the parameter of a multinomial distribution for choosing the themes and η is the K-dimensional Dirichlet parameter. Total number of themes is assumed known and fixed as K.

2. For each of the N subgraphs g_i in the image graph

   (a) Choose a theme $z_i \sim \text{Multinomial}(\delta)$ where z_i is a K-dimensional unit vector. $z_i^k = 1$ indicates that the kth theme is selected.

   (b) Choose a subgraph g_i from $p(g_i \,|\, z_i, \beta)$, a multinomial probability conditioned on the theme z_i and β is a $K \times T$ matrix where $\beta_{ij} = p(g^j = 1 \,|\, z^i = 1)$, is a fixed quantity for the graph set.

A K-dimensional Dirichlet random variable δ satisfies $\delta_i \geq 0$ and $\sum_{i=1}^{K} \delta_i = 1$. It has the following probability density:

$$p(\delta \,|\, \eta) = \frac{\Gamma\left(\sum_{i=1}^{K} \eta_i\right)}{\prod_{i=1}^{K} \Gamma(\eta_i)} \delta_i^{\eta_1 - 1} \cdots \delta_K^{\eta_K - 1} \tag{5.3}$$

where the parameter η is a K-vector with components $\eta_i > 0$, and $\Gamma(\cdot)$ is the Gamma function.

Given the parameters η and β, the joint distribution of a theme mixture δ, a set of N themes $Z = \{z_1, \ldots, z_N\}$, and an image graph G having N subgraphs $\{g_1, \ldots, g_N\}$ is given by

$$p(\delta, Z, G \,|\, \eta, \beta) = p(\delta \,|\, \eta) \prod_{n=1}^{N} p(z_n \,|\, \delta)\, p(g_n \,|\, z_n, \beta) \tag{5.4}$$

where $p(z_n \,|\, \delta)$ is simply δ_i for the unique i such that $z_n^i = 1$.

The LDA model is represented as a probabilistic graphical model in Figure 5.1. The parameters η and β are the dataset-level parameters. They are assumed to be sampled once for generating a graph set. The variable δ is graph-level variable which is sampled once per image graph. Finally, the variables z and g are subgraph-level parameters, sample once for each subgraph in an image graph.

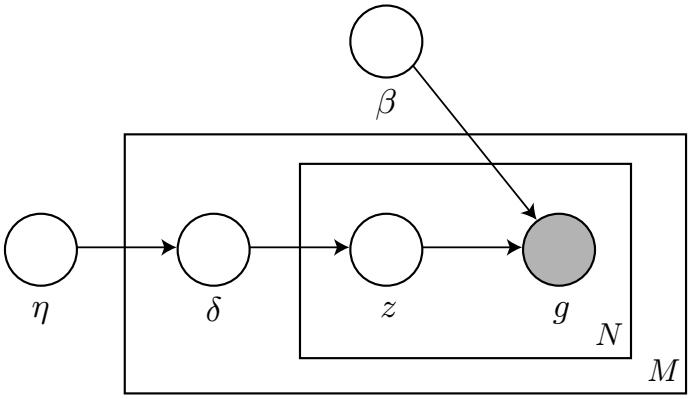


Figure 5.1: Graphical model representation of LDA. The boxes are *plates* representing replicates. The outer plate represents image graphs, while the inner plate represents the repeated choice of themes and subgraphs within an image graph [7].

In the context of text modeling, the topic probabilities provide an explicit representation of a document [7]. Equivalently, we use the theme probabilities to represent an image graph. To do so, we need to compute the posterior distribution of the hidden variables given an image graph:

$$p(\delta, Z \mid G, \eta, \beta) = \frac{p(\delta, Z, G \mid \eta, \beta)}{p(G \mid \eta, \beta)}. \tag{5.5}$$

Integrating (5.4) over δ and summing over z, we obtain the marginal distribution of an image graph in terms of the model parameters:

$$p(G \mid \eta, \beta) = \frac{\Gamma\left(\sum_{i=1}^{K} \eta_i\right)}{\prod_{i=1}^{K} \Gamma(\eta_i)} \int \left(\prod_{i=1}^{K} \delta_i^{\eta_1 - 1} \right) \left(\prod_{n=1}^{N} \sum_{i=1}^{K} \prod_{j=1}^{T} (\delta_i \beta_{ij})^{g_n^j} \right) d\delta. \tag{5.6}$$

Unfortunately, this distribution is intractable due to the coupling between δ and β in the summation over latent themes [7]. However, a wide range of approximate inference algorithms can be considered, including Laplace approximation, variational approximation and MCMC method [7]. The solution proposed by Blei *et al.* is approximating the distribution $p(\delta, Z \mid G, \eta, \beta)$ by a simplified graphical model with free variational parameters as

$$q(\delta, Z \mid \psi, \phi) = q(\delta \mid \psi) \prod_{n=1}^{N} q(z_n \mid \phi_n) \tag{5.7}$$

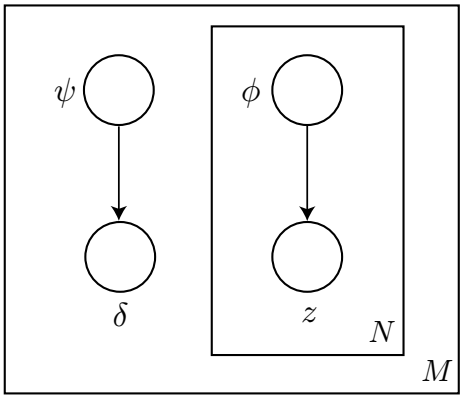


Figure 5.2: Graphical model representation of the variational distribution used to approximate the posterior in LDA [7].

where the Dirichlet parameter ψ and the multinomial parameters $(\phi_1, \ldots, \phi_N)$ are the free variational parameters. The next step is to find the values of the variational parameters ψ and ϕ, which satisfy the best approximation as

$$(\psi^*, \phi^*) = \arg\min_{(\psi,\phi)} KL\big(q(\delta, Z \mid \psi, \phi) \,\big\|\, p(\delta, Z \mid G, \eta, \beta)\big). \tag{5.8}$$

Thus, the optimal values of the variational parameters are found by minimizing the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior $p(\delta, Z \mid G, \eta, \beta)$. The values are computed using the EM algorithm; however, we do not include these steps to this thesis. The details of the computation can be found in [7]. The computations show that the optimal parameters (ψ^*, ϕ^*) are functions of the given graph G such that $\big(\psi^*(G), \phi^*(G)\big)$. Therefore, the theme probabilities of an unseen image graph can be computed in the same manner. In addition, the values of the model parameters η and β are estimated from the variational parameter values of all image graphs in the graph set, iteratively. The graphical model of this variational distribution can be seen in Figure 5.2.

Given a set of image graphs, we can estimate the model parameters η and β using variational inference. The graph-level variable δ can be used for graph representation. It is a K-dimensional vector where the ith component equals to the probability $p(z_i \mid \delta, G)$ for the given graph G. The variational method finds the optimal Dirichlet parameter $\psi^*(G)$ of the distribution that generates the variable

δ for the given graph G. Therefore, normalizing the K-dimensional Dirichlet parameter gives the expected value of the theme distribution which can be used for graph representation.

After training the LDA model on homogeneous tiles for a defined number of themes, we compute the theme distribution vectors of all tiles in the whole satellite image using variational inference. Finally, the theme probabilities of tiles are used for partitioning the whole image into semantic regions.

Chapter 6

Experimental Results

Where is the 'any' key?

Homer Simpson – in response to the message "press any key"

In this chapter, we present the results of the experiments conducted for the proposed method in comparison to the bag-of-words model. The dataset used in the experiments is previously described in Section 1.3. In the following sections we first describe the experimental setup and then demonstrate the experimental results.

6.1 Experimental Setup

We explained the proposed method for image classification in the previous three chapters. The following section gives the values for the parameters of the method and the external softwares used as a part of experiments. We conduct experiments also for the bag-of-words model in the same dataset. For the bag-of-words model we extract the histogram vector of the node labels computed from individual MSERs for each image and then the same classification procedure given in Chapter 5 is applied.

6.1.1 Graph Construction Parameters

The transformation process has three parameter sets: The MSER parameter sets Ω_{low}, Ω_{high} and the parameter set for the number of labels Υ. We determine the MSER parameters experimentally as $\Omega_{\text{high}} = (10, 60, 5000, 0.4, 1)$ and $\Omega_{\text{low}} = (5, 35, 1000, 4, 1)$. Also, the ellipse expanding parameters in (3.6) and (3.7) are experimentally set to $r_1^2 = 5$ and $r_2^2 = 20$. The MSER algorithm is applied to the whole remote sensing image with these parameters and the features are extracted from the image. Then, the node labels are determined by the k-means algorithm with $\Upsilon = (k_{\mathsf{sd}}, k_{\mathsf{sb}}, k_{\mathsf{ud}}, k_{\mathsf{ub}})$. We set k_{sd} and k_{sb} to equal values; and similarly k_{ud} and k_{ub}. The number of highly stable MSERs is less than the number of less stable ones. Therefore, we choose the number of labels for less stable MSERs as twice the number of labels for highly stable ones. Hence, we use the total number of labels denoted by N_ℓ instead of the set Υ. For example, we say $N_\ell = 36$ for $\Upsilon = (6, 6, 12, 12)$. The parameter value of N_ℓ is selected from the set $\{18, 26, 36, 54, 71\}$ in the experiments. Next, the graph transformed from the whole image are cut into tiles and every homogeneous tile is labeled by one of the eight semantic classes. Finally, these tiles are divided into two sets which have (almost) same number of tiles for each class. One of them which has 295 images is used for model training and the other set containing 290 images is used for testing. The external software used in this section is VLFeat [43] for not only the MSER algorithm but also the k-means++ algorithm.

6.1.2 Graph Mining Parameters

The mining algorithm is based on the evaluation predicate. We use the predicate φ in (4.64) for graph mining. The graph mining algorithm has three steps. The parameter of the first step is the support threshold θ_c^{supp} for each class $c \in \mathcal{C}$. A global threshold for all classes is not reasonable because the size of dataset $\mathcal{D}_c$ varies from one class to another. A threshold for $\mathsf{freq}(\pi, \mathcal{D})$ may be applied; however, the density of MSERs also differs between classes. We choose three different threshold values for each class in order to simulate the effects of high,

medium and low thresholds which allow small, medium and large number of subgraphs to pass to the next filer, respectively. The desired size of $\mathcal{F}_c$ for high, medium and low thresholds are determined as 200, 500 and 800, respectively. We denote this number by N_θ. To choose the support thresholds accordingly, we rank all subgraphs in order of support and select the support value of the N_θth subgraph as the threshold θ_c^{supp} for each class. The same procedure is applied for the correlation threshold θ_c^{cor}. Note that the correlation thresholds are chosen from the ranking of *all subgraphs*, not only the frequent ones. In the experiments, we use the same parameter N_θ for the selection of both thresholds in a single run of the algorithm. The last significant parameter of the algorithm is the number N_s of subgraphs selected for the set $\mathcal{S}_c$ for each class. We have eight classes, so each image is represented by a $8 \times N_s$-dimensional subgraph histogram vector. The domain of the parameter N_s is $\{1, 2, \ldots, 20\}$ in the experiments.

Addition to the parameters N_ℓ, N_θ and N_s, there are some less important parameters: Subgraph size limits $v_{\min}$ and $v_{\max}$ in (4.64), the number of components in the Poisson mixtures m in (4.11), and the regularization constant ξ in (4.31). We mine subgraphs which have at least two nodes $v_{\min} = 2$, and not more than five nodes $v_{\max} = 5$. The upper limit is used for reducing the computational complexity of frequent subgraph mining and the lower limit is set for differentiating subgraphs from the visual words of individual nodes (MSERs). The number of components in the Poisson mixtures is set to $m = 5$ for all subgraphs and the parameter ξ is chosen differently for each class as $\xi_c = \ln(2) \,/\, \mathsf{freq}(\pi, \mathcal{D}_c)$ because the distance which corresponds to the average match in the dataset becomes equal to 0.5 by

$$d'\big(\mathsf{freq}(\pi, \mathcal{D}_c)\big) = 1 - e^{-\xi_c \, \mathsf{freq}(\pi, \mathcal{D}_c)} = 0.5. \tag{6.1}$$

The external software of this section is the Molecular Substructure Miner (MoSS) implemented by Borgelt [8] for mining frequent subgraphs.

6.1.3 Classifier Parameters

As mentioned in the previous chapter, there are two parameters of the SVM classifiers: C and γ. We set these parameters using grid search on 5-fold cross-validation with several parameter pairs from exponentially growing sequences. The domains are $C \in \left\{2^{-3}, 2^{-2}, \ldots, 2^{5}\right\}$ and $\gamma \in \left\{2^{-5}, 2^{-4}, \ldots, 2^{2}\right\}$. The parameter pair with the highest classification accuracy is selected as SVM parameters. If more than one pair has the same classification accuracy, the pair with the maximum-margin is selected. On the other hand, LDA has only one parameter: The number of themes K. The LDA model is learned from the training graph set and this model is used for inference of variational parameters for all tiles in the Antalya image.

6.2 Classification Results

The experiments are performed on an Ikonos image of Antalya, Turkey given in Figure 1.2. The tiles (250×250 pixel size) cut from this image are used in the classification experiments for eight semantic classes:

(a) dense residential areas with large buildings,

(b) dense residential areas with small buildings,

(c) dense residential areas with trees,

(d) sparse residential areas,

(e) greenhouses,

(f) orchards,

(g) forests, and

(h) fields.

Table 6.1: The number of images in the training and testing datasets for each class. Class names are in the text.

class	**#training**	**#testing**	**total**
(a)	40	39	79
(b)	35	35	70
(c)	18	17	35
(d)	24	24	48
(e)	17	17	34
(f)	82	81	163
(g)	38	37	75
(h)	41	40	81
total	295	290	585

The number of images in the training and testing datasets for each class are given in Table 6.1. The experiments are repeated on different parameter combinations in order to demonstrate the effects of parameters on classification performance.

The distribution of some MSER clusters over the whole image is given in Figures 6.1 and 6.2. The instances of the MSER clusters in these images are concentrated over a unique compound structure type. This shows the success of feature extraction step. In other words, the features extracted from the MSERs and their surroundings are so adequate to capture the local image content that we can distinguish the scene types by monitoring the locations in which they are detected. This situation is the main reason for the high performance of the bag-of-words model.

The classification accuracy of the graph mining algorithm for all parameter combinations in the experiments is shown in Table 6.2. The accuracy is computed for the test dataset, as the ratio of correctly classified images to the total number images. Note that the number of images that belong to one class varies from 17 (greenhouses) to 81 (orchards) in the test dataset of 290 images of 8 classes. The best classification accuracy is 92.069 percent, achieved by the parameter set $(N_\ell, N_\theta, N_s) = (36, 200, 9)$.

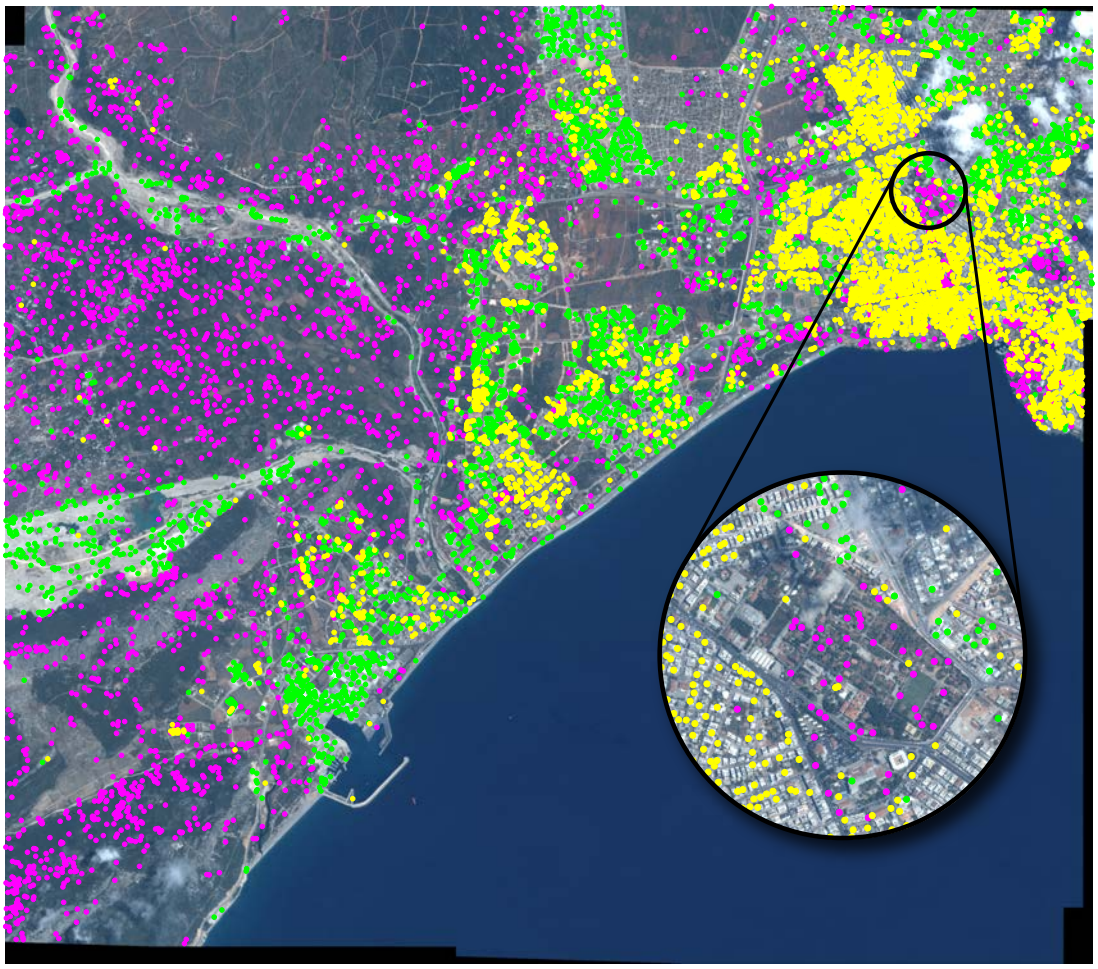


Figure 6.1: Three clusters of stable dark MSERs are drawn with different colors at ellipse centers for $N_\ell = 36$. Yellow, green and magenta points are concentrated on dense residential areas with large buildings, dense residential areas with small buildings and orchards, respectively.

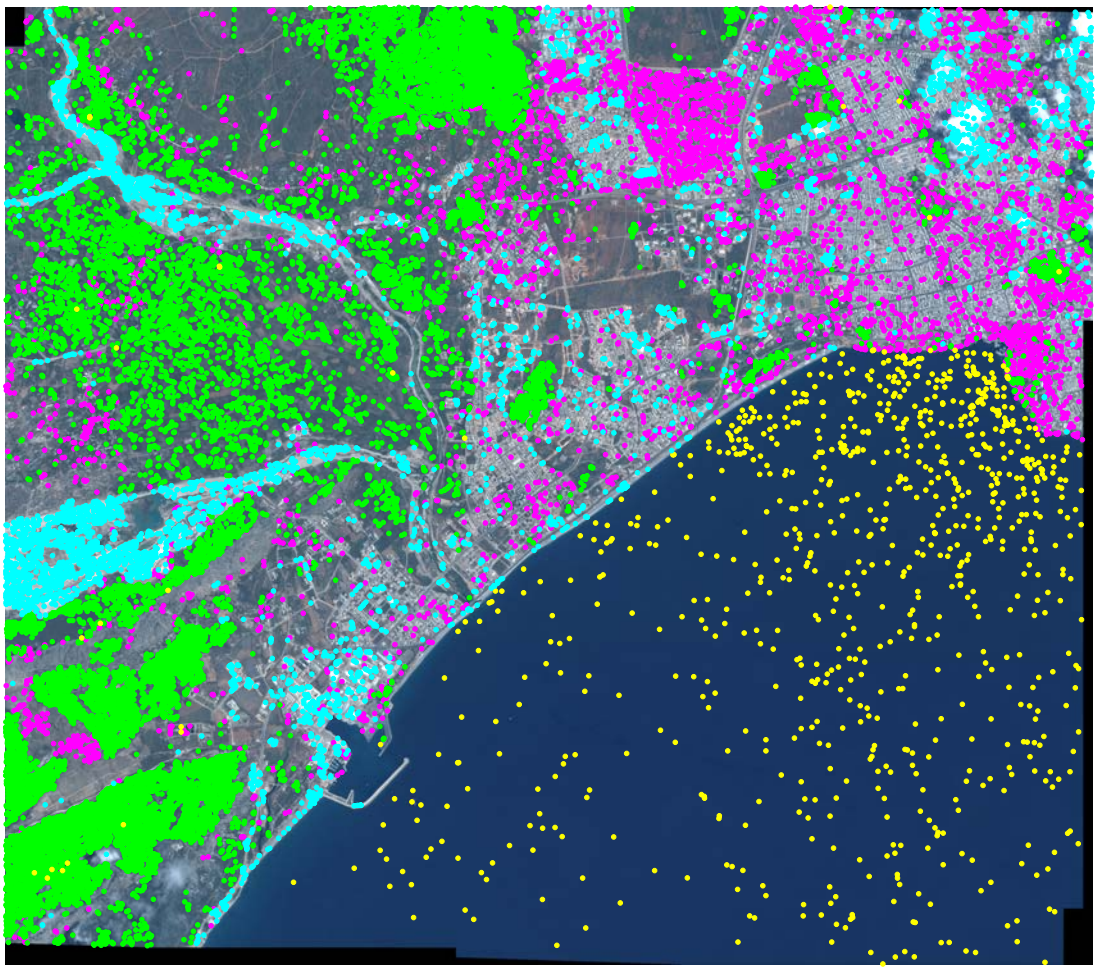


Figure 6.2: Four clusters of different type MSERs are drawn with different colors at ellipse centers for $N_\ell = 36$. Yellow, green, cyan and magenta points are concentrated on sea, forests, stream bed/clouds and dense residential areas with trees, respectively.

Table 6.2: The classification accuracy of the graph mining algorithm, in percentage (%), for all parameter sets in the experiments.

				Number of Subgraphs per class (N_s)																		
				1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Number of Labels (N_ℓ)	**18**	N_θ	200	83	81	81	87	86	88	86	87	87	87	86	86	85	79	83	82	81	76	81	78
			500	73	82	85	83	82	82	88	86	88	87	88	87	85	82	82	79	78	76	75	74
			800	68	77	82	84	85	86	86	87	88	89	88	87	86	86	84	84	82	79	75	76
	26[a]	N_θ	200	79	79	82	87	88	90	91	89	87	86	85	89	88	89	88	88	88	86	86	84
			500	74	72	84	83	84	85	86	81	86	88	88	87	83	86	84	83	84	84	83	81
			800	71	77	81	83	85	86	88	87	89	87	87	85	88	84	82	83	76	77	75	75
	36	N_θ	200	80	78	82	87	89	90	91	91	**92**	90	90	90	88	91	88	91	91	89	85	84
			500	80	77	87	85	87	90	90	89	88	88	89	88	89	88	87	87	73	71	73	69
			800	79	81	87	86	86	86	86	88	88	88	87	84	88	79	76	85	76	83	71	69
	54	N_θ	200	73	77	80	83	88	89	88	86	89	88	86	86	86	87	88	85	78	80	68	68
			500	74	79	81	82	82	84	84	88	87	88	88	84	81	72	82	73	76	69	67	66
			800	72	77	79	83	85	85	85	85	84	81	80	83	79	78	72	68	74	67	66	67
	72	N_θ	200	79	80	79	83	86	83	86	82	84	87	84	81	80	77	81	73	77	76	64	71
			500	79	78	80	80	81	81	80	82	84	81	81	80	81	80	76	72	68	67	65	62
			800	79	79	81	84	82	81	80	81	76	71	71	77	74	71	70	68	72	64	61	60

[a]Corresponding cluster parameters is $\Upsilon = (4, 4, 9, 9)$

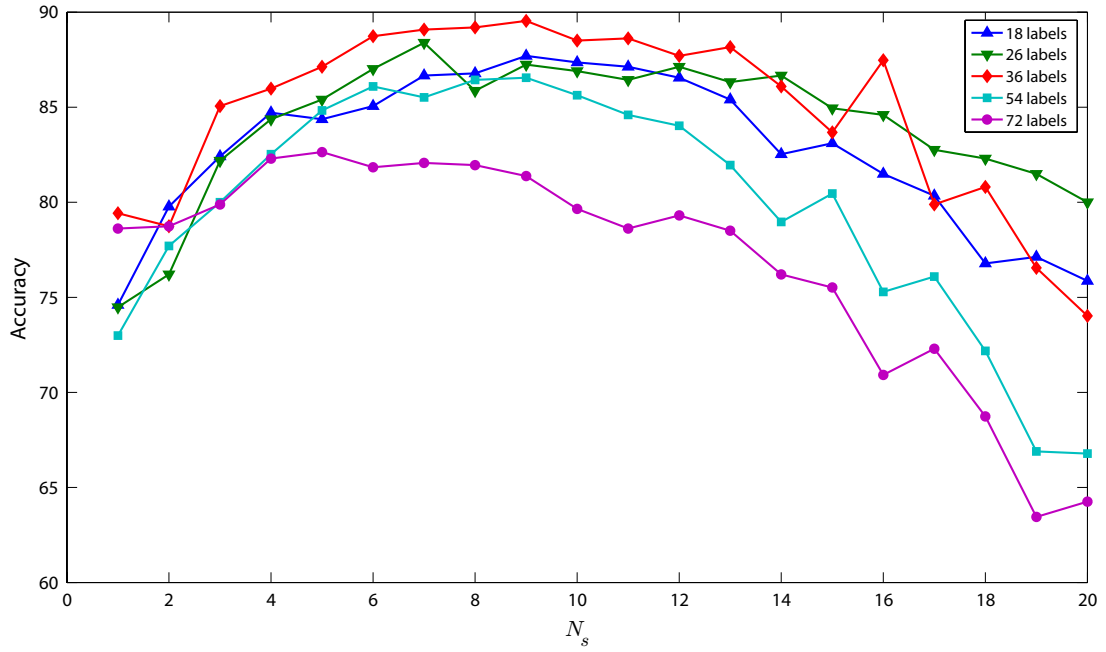


Figure 6.3: Plot of classification accuracy of the graph mining algorithm for five different number of labels over the number of subgraphs per class. The lines are drawn by averaging the accuracy values for the parameters $N_\theta \in \{200, 500, 800\}$.

Figure 6.3 demonstrates the effects of N_ℓ and N_s. Given these parameters, the line is drawn by averaging the classification accuracy over the set $N_\theta \in \{200, 500, 800\}$. If we start from $N_s = 1$ and increase the number of subgraphs selected for the set $\mathcal{S}$, i.e. the dimensionality of the feature space, the classification accuracy also increases until about $N_s = 8$ or 10 for all values of N_ℓ. However, after this point the classification accuracy decreases dramatically as expected because of the curse of dimensionality. A similar behavior is observed for the parameter N_ℓ. The best accuracy results are obtained when the number of labels is 36. High number of node labels such as 72 causes a significant decrease in subgraph frequency which leads to a fall in the accuracy. On the other hand, small number of node labels like 18 increases the subgraph frequency but reduces the correlation between subgraphs and classes. As a result, the confusion caused by less correlated subgraphs results in low classification accuracy. To sum up, every number of node labels entails trade-offs between frequency and correlation. Finding the optimal number of node labels in terms of classification accuracy requires experiments on the dataset with a range of N_ℓ values.

Besides classification performance, the number of node labels plays an important role in computational complexity. The crucial part of the graph mining algorithm with respect to the computational complexity is the frequent subgraph mining. The computational cost depends heavily on the number of overlapping embeddings because the MIS support solves an optimization problem for every overlapping embeddings. Small number of labels increases the number of overlapping embeddings and some classes, especially natural lands such as forest, fields and orchards tend to produce overlapping embeddings more than man-made structures like residential areas.

Figure 6.4 demonstrates the effects of N_θ and N_s. Given these parameters, the line is drawn by averaging the classification accuracy over the set $N_\ell \in \{18, 26, 36, 54, 72\}$. The parameter N_s, namely the number of subgraphs selected per class represents similar property as in Figure 6.3. On the other hand, a decrease in the thresholds θ_c^{supp} and θ_c^{cor} (inversely proportional to N_θ) causes the classification accuracy to decrease. The analysis of this case is harder than the previous. One possible explanation is that the set $\mathcal{S}$ contains strongly correlated but less frequent subgraphs for low thresholds. The last part of the mining algorithm uses the correlation function as a significance measure. Therefore, the selected subgraphs may be still highly correlated despite the low correlation threshold but they may be less frequent. For example, assume a pattern π which is found once or twice in almost all examples of class c but not found in other examples. The pattern π is considered to be a correlated pattern but it is not frequent and may not be found in test examples. In summary, the last part of the mining algorithm compensates a lower correlation threshold by selecting the most correlated and not redundant pattern. Although the correlation measure and pattern support are somewhat related (see Theorem 4.1), a lower threshold for the subgraph support may not be fully recovered by the other parts. Consequently, the selection of support threshold should be done carefully, that allows a sufficient number of subgraphs to satisfy the frequency criterion considering the redundancy between patterns.

The classification performance of the bag-of-words model in comparison to the graph mining algorithm can be seen in Table 6.3. As seen in the table, the

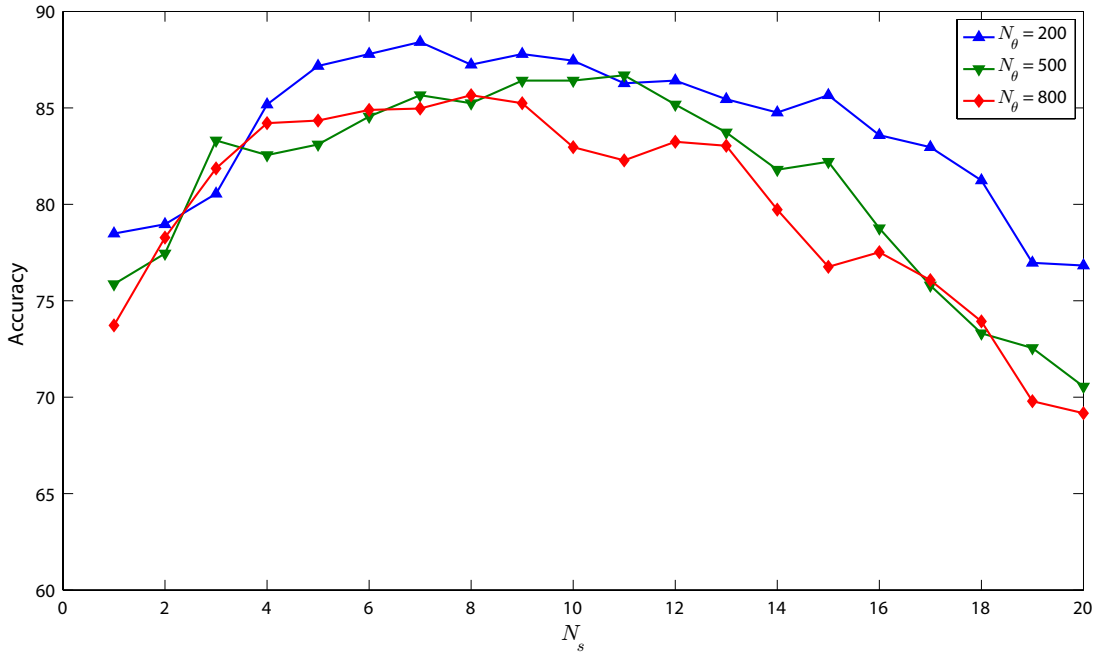


Figure 6.4: Plot of classification accuracy of the graph mining algorithm for three different N_θ values over the number of subgraphs per class. The lines are drawn by averaging the accuracy values for the parameters $N_\ell \in \{18, 26, 36, 54, 72\}$.

frequency of visual words in the bag-of-words model is less affected from the number of node labels because the graph mining method seek structural elements in the images. On the hand, bag-of-words model fails to classify images for small number of node labels. The graph mining is less affected from the confusion using the spatial relationships between local image patches. The best classification accuracies of both methods are almost same. However, further improvement cannot be obtained because of the intrinsic properties of the dataset. The small tile size due to the heterogeneous content of the satellite image limits the frequency of the subgraphs and the method suffers from the low subgraph support. Using bigger tiles might improve the performance of the graph mining method.

The confusion matrices of the graph mining algorithm and the bag-of-words models for the best parameter sets are presented in Figures 6.5 and 6.6, respectively. The most confused class in both matrices is greenhouses. It is about another problem of the dataset. Greenhouses are naturally located near sparse residential areas (villages) and orchards. Therefore, the images of greenhouses

Table 6.3: Classification accuracy of the bag-of-word model and the mining algorithm, in percentage terms, for different number of words/labels.

#Labels (N_ℓ)	**BoW Accuracy**	**Max. Mining Accuracy**
18	79.310	88.966 ($N_s = 10$, $N_\theta = 500$)
26	92.414	91.034 ($N_s = 7$, $N_\theta = 200$)
36	91.035	92.069 ($N_s = 9$, $N_\theta = 200$)
54	91.035	88.621 ($N_s = 9$, $N_\theta = 200$)
72	91.724	86.897 ($N_s = 10$, $N_\theta = 200$)

in the dataset are not completely homogeneous and contain structures belonging to other classes as seen in Figure 6.7. Our graph mining algorithm essentially handles such problems by mining class correlated subgraphs. It seeks a set of subgraphs which are commonly found among the examples of the class and some images having other structures do not constitute an important problem for the class. However, greenhouses are located sparsely in the Antalya image and almost all greenhouse images contain structures of orchards or sparse residential areas. As a result, the mining algorithm treats those structures as if they are correlated with the greenhouses class. This explains the reason for the relatively lower performance on the greenhouses and puts the obstacles in the way of improving the general performance of the mining algorithm. Sample images for each class are shown in Figure 6.7. They are grouped according to their classification result by the graph mining algorithm.

Finally, we apply the SVM model learned from the training set to every tile in the Antalya image. The classification result is drawn in Figure 6.8. Similarly, we discover themes in the Antalya image using the LDA model whose parameters are estimated from the training dataset for 12 themes. The set of subgraphs used in the LDA model is the set $\mathcal{S}$ which is found by the graph mining algorithm. The LDA model gives insight into distributions of themes over the whole image. Hence, heterogeneous tiles contents are analyzed in a better manner. However, presenting all topics distribution in this study for the Antalya image is a difficult task. Therefore, we combine this distributions into one image in Figure 6.9 where each color represents a region where the corresponding theme dominates other

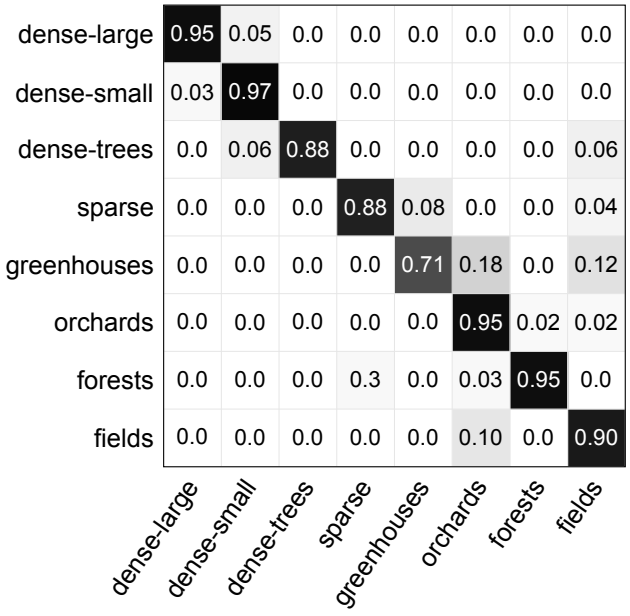


Figure 6.5: The confusion matrix of the graph mining algorithm using the parameters $N_\ell = 36, N_\theta = 200$ and $N_s = 9$. Class names are given in short: *sparse* and *dense* are used for sparse and dense residential areas, respectively. Also, *large* and *small* mean large and small buildings, respectively.

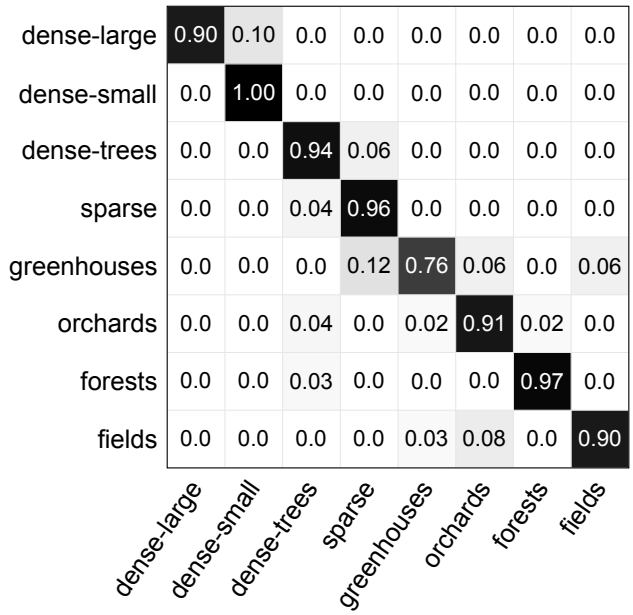


Figure 6.6: The confusion matrix of the bag-of-words model for 26 labels. Class names are given in short: *sparse* and *dense* are used for sparse and dense residential areas, respectively. Also, *large* and *small* mean large and small buildings, respectively.

(a) Correctly classified images (b) Misclassified images

Figure 6.7: Sample images from the dataset. The images at the left are correctly classified by the graph mining algorithm while the images at right-hand side are misclassified using the parameters $N_\ell = 36, N_\theta = 200$ and $N_s = 9$. The image classes from top to down are in the order: dense residential areas with large buildings, dense residential areas with small buildings, dense residential areas with trees, sparse residential areas, greenhouses, orchards, forests and fields.

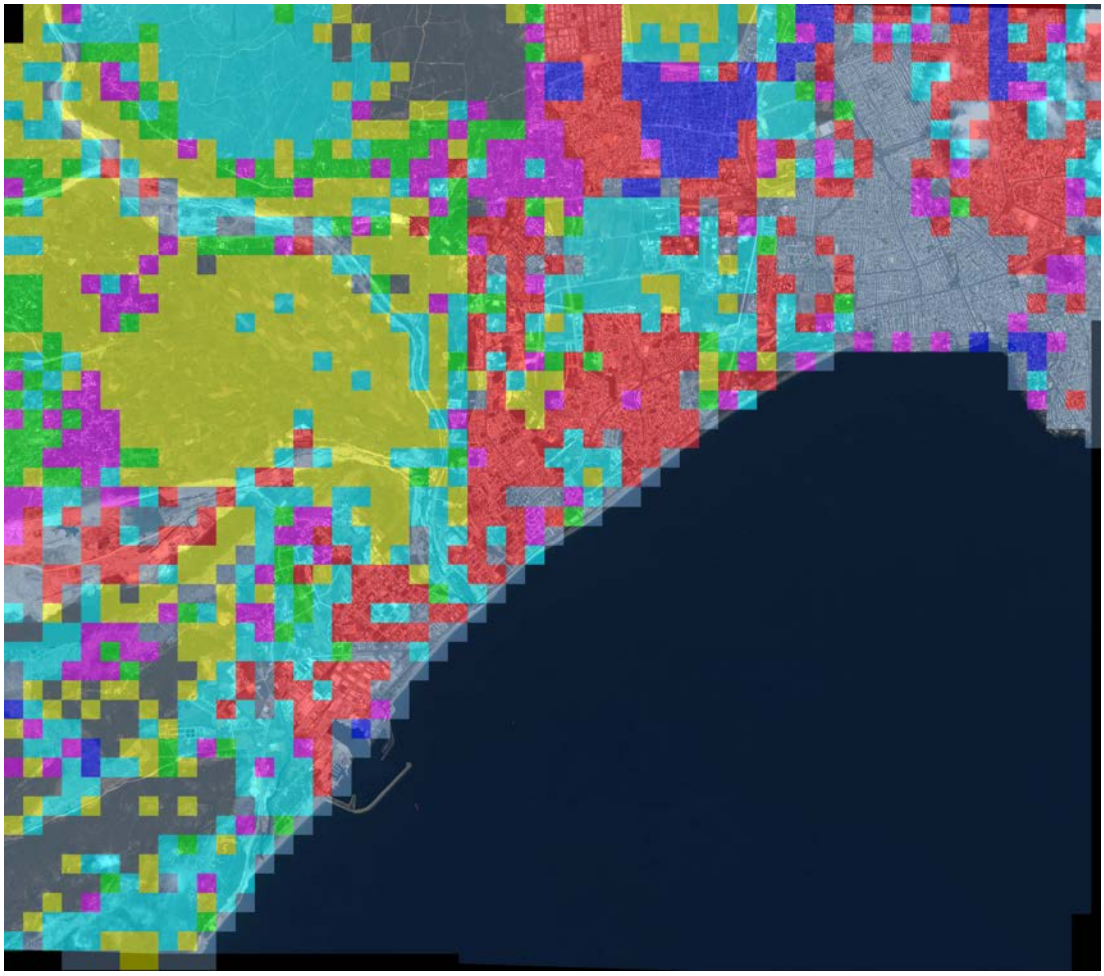


Figure 6.8: The classification of all tiles except sea using the SVM learned from the training set for the parameters $N_\ell = 36, N_\theta = 200$ and $N_s = 9$. Each color represents a unique class.

themes. The probability distributions of the most dominating 6 themes found by the LDA model trained for 16 themes are drawn in Figure 6.10. The further extensions of the LDA model remain as the future work for this study such as theme localization by analyzing the locations where the correlated subgraphs are found and testing with the model with other subgraph sets.

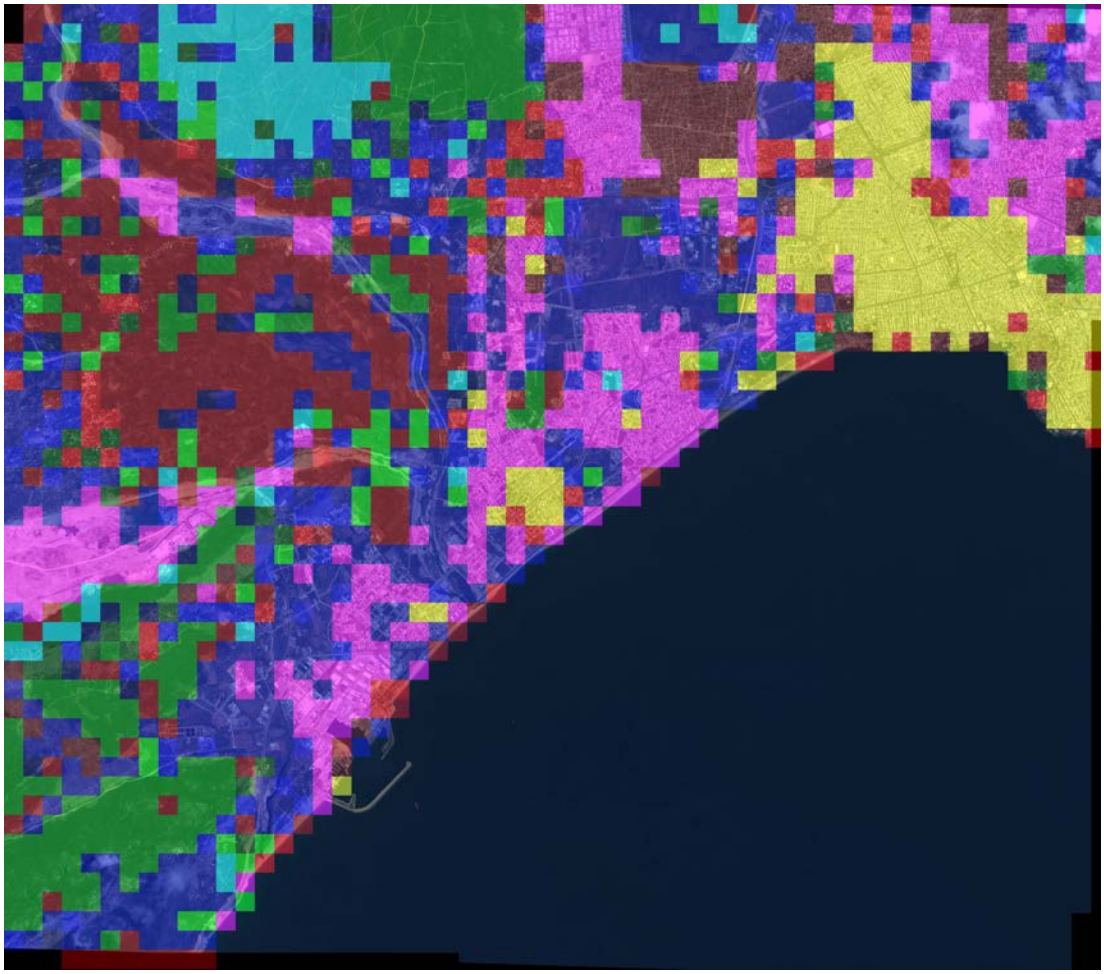


Figure 6.9: Every tile is labeled by a unique color which indicates the corresponding theme that dominates the other themes in that tile. The theme distributions are inferred from the LDA model for 12 themes. The subgraph set is the one mined in the previous experiments for the best parameters.

Figure 6.10: The most dominating 6 themes are shown, found by the LDA model trained for 16 themes. The intensity of red color represents the probability of the theme in an individual tile.

Chapter 7

Conclusions and Future Work

It would seem that perfection is attained
not when there is nothing more to add,
but when there is nothing left to take away.

"Wind, Sand and Stars" – Antoine de Saint-Exupéry

7.1 Conclusions

We emphasized the importance of high-level understanding of the image content through compound structures and we discussed the challenges of detecting compound structures. Accordingly, we described a new image content representation using the histogram of a subgraph set for classifying complex scenes such as dense and sparse urban areas. As the first step of this method, we transformed images to graphs where the nodes store local image content and the edges encode spatial information. We proposed a graph construction method where the patches, encoded by the graph nodes, are detected using maximally extremal stable regions and discriminative information about these regions are obtained by extracting features from these regions in relation to their surroundings. The features were quantized to form a codebook of local information that determined the node labels and the neighborhood relation between local patches were found from the

Voronoi tessellations of the patches.

In the second step of the method, we selected the subgraphs which are used in the histogram representation of images where the subgraphs encapsulate the local patches and their spatial arrangements within a specific structure. We described a graph mining algorithm to find the set of frequent and discriminative subgraphs which also has low redundancy. The algorithm first mines the frequent subgraphs in the image graph set. Then, it selects the discriminative subgraphs among the frequent ones with respect to the correlation between the subgraphs and the classes. We introduced a novel algorithm for extracting class-correlated patterns using the probabilistic model of subgraph frequency in an individual graph. Finally, the redundancy between the subgraphs in the set was resolved by choosing the most significant subgraphs considering the distances between them.

The third step of the method is the model learning from the vector space representations of the images. A multi-class support vector machine was employed for classifying the images. Furthermore, the latent Dirichlet allocation model was used for further classification of images. The LDA model provides the theme distribution representation of images computed from the subgraph histograms. Themes provide a better understanding of the images having heterogeneous content.

In experimental work, we evaluated the performance of the graph mining algorithm in image classification compared to the bag-of-words model. The dataset consists of tiles cut from an Ikonos image of Antalya image and each tile is labeled by one of the eight high-level semantic scene types. The classification accuracy of the graph mining algorithm shows the effectiveness of the proposed method in classification of complex scene types. We observed and discussed the effects of the parameters on the classification accuracy. We concluded that the graph mining algorithm is capable of discriminating images of different scene types successfully. Furthermore, the LDA model manages to discover interesting themes in the whole satellite image.

7.2 Future Work

The proposed image representation can be easily adapted to other application areas of computer vision. To illustrate, the application of subgraph histogram representation can be extended to image retrieval by defining a distance measure between subgraph histograms or borrowing a distance definition from the information retrieval literature. Given an input image, the most similar images in the dataset are the closest ones according to the distance function.

The LDA model offers new applications of the proposed image representation. Given a given graph set and a subgraph set, the LDA model finds the subgraph-theme probabilities β. A subgraph that is highly correlated with a theme can be used for localizing the theme distribution in a heterogeneous content. This enables high-level partitioning of heterogeneous images. Another application of the LDA model is unsupervised partitioning of the whole satellite image using subgraph histograms in case the labeled data are not available or the scene types are unknown. In such cases, the subgraph set contains all subgraphs generated by the graph language for a fixed-size. Given the number of themes, the LDA tries to discover themes from the whole image using the occurrence numbers of all subgraphs in tiles.

Finally, the set of subgraphs selected by the graph mining algorithm from an Ikonos image of Antalya can be used for classification of images cut from another satellite image which is retrieved from another satellite in a different spatial-resolution, by the help of a mapping function. The function maps the features extracted from the MSERs of the second image to the node labels determined for the first image. The parameters of the graph construction method should be adjusted for the spatial resolution of the second image, then the mapping function determines, after normalization, the closest cluster center of the first image to the features of an MSER in the second image.

Bibliography

[1] S. Aksoy. Modeling of remote sensing image content using attributed relational graphs. In D. Yeung, J. T. Kwok, A. Fred, F. Roli, and D. Ridder, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 4109, chapter 52, pages 475–483. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[2] S. Aksoy. *Signal and Image Processing for Remote Sensing*, chapter Spatial techniques for image classification, pages 491–513. Taylor & Francis, 2006.

[3] D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[4] E. Barbu, P. Hroux, S. Adam, and E. Trupin. Clustering document images using graph summaries. In P. Perner and A. Imiya, editors, *Machine Learning and Data Mining in Pattern Recognition*, volume 3587 of *Lecture Notes in Computer Science*, pages 194–202. Springer, 2005.

[5] C. Berge. *Hypergraphs*. North-Holland Mathematical Library, 1989.

[6] J. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, International Computer Science Institute, 1998.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[8] C. Borgelt. MoSS: Molecular substructure miner. `http://www.borgelt.net/moss.html`, 2009.

[9] J. Bourgain. On lipschitz embedding of finite metric spaces in hilbert space. *Israel Journal of Mathematics*, 52(1):46–52, March 1985.

[10] B. Bringmann. *Mining Patterns in Structed Data*. PhD thesis, Katholieke Universiteit Leuven, Celestijnenlaan 200 A – B-3001 Leuven, Belgium, 2009.

[11] H. Bunke, P. Dickinson, M. Kraetzl, M. Neuhaus, and M. Stettler. *Applied Pattern Recognition*, chapter Matching of Hypergraphs – Algorithms, Applications, and Experiments, pages 131–154. Springer Berlin / Heidelberg, 2008.

[12] H. Bunke and K. Riesen. Graph classification based on dissimilarity space embedding. In *Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 996–1007, Berlin, Heidelberg, 2008. Springer-Verlag.

[13] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[14] H. Cheng, X. Yan, J. Han, and C. wei Hsu. Discriminative frequent pattern analysis for effective classification. In *IEEE International Conference on Data Engineering*, pages 716–725, 2007.

[15] K. W. Church and W. A. Gale. Poisson mixtures. *Natural Language Engineering*, 1:163–190, 1995.

[16] M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engineering*, 17(8):1036 – 1050, aug. 2005.

[17] E. Dogrusoz and S. Aksoy. Modeling urban structures using graph-based spatial patterns. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 4826 –4829, 23-28 2007.

[18] M. Fiedler and C. Borgelt. Subgraph support in a single large graph. In *IEEE International Conference on Data Mining Workshops*, pages 399–404, Washington, DC, USA, 2007. IEEE Computer Society.

[19] R. Gaetano, G. Scarpa, and G. Poggi. Hierarchical texture-based segmentation of multiresolution remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 47(7):2129 –2141, july 2009.

[20] X. Gao, B. Xiao, D. Tao, and X. Li. Image categorization: Graph edit distance+edge direction histogram. *Pattern Recognition*, 41(10):3179 – 3191, 2008.

[21] Z. Harchaoui and F. Bach. Image classification with segmentation graph kernels. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[22] G. Ivncsy, R. Ivncsy, and I. Vajk. Graph mining-based image indexing. In *International Symposium of Hungarian Researchers on Computational Intelligence*, pages 313–323, Budapest, Hungary, Nov 2004.

[23] H. Jiang and C. W. Ngo. Image mining using inexact maximal common subgraph of multiple args. In *International Conference on Visual Information Systems*, 2003.

[24] Y. G. Jiang, J. Yang, C. W. Ngo, and A. G. Hauptmann. Representations of keypoint-based semantic concept detection: A comprehensive study. *IEEE Transactions on Multimedia*, 12(1):42–53, November 2009.

[25] M. Kuramochi and G. Karypis. Finding frequent patterns in a large sparse graph. *Data Mining and Knowledge Discovery*, 11(3):243–271, 2005.

[26] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, 2006.

[27] E. Levina and P. Bickel. The earth mover's distance is the mallows distance: some insights from statistics. In *IEEE International Conference on Computer Vision*, volume 2, pages 251–256 vol.2, 2001.

[28] F.-F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society.

[29] T. Li, T. Mei, I.-S. Kweon, and X.-S. Hua. Contextual bag-of-words for visual categorization. *IEEE Transactions on Circuits and Systems for Video Technology*, PP(99):1 –1, Jan 2010.

[30] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.

[31] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, 2002.

[32] R. Myers, R. Wison, and E. Hancock. Bayesian graph edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):628 –635, jun 2000.

[33] S. Nijssen, T. Guns, and L. De Raedt. Correlated itemset mining in roc space: a constraint programming approach. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 647–656, New York, NY, USA, 2009. ACM.

[34] S. Nijssen and J. N. Kok. Multi-class correlated pattern mining. In *Knowledge Discovery in Inductive Databases*, pages 165–187, 2005.

[35] S. Nowozin, K. Tsuda, T. Uno, T. Kudo, and G. H. Bakir. Weighted substructure mining for image analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[36] E. Pekalska and R. P. W. Duin. *The Dissimilarity Representation for Pattern Recognition: Foundations And Applications (Machine Perception and Artificial Intelligence)*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2005.

[37] K. Riesen and H. Bunke. Iam graph database repository for graph based pattern recognition and machine learning. In *Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 287–297, Berlin, Heidelberg, 2008. Springer-Verlag.

[38] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, November 2000.

[39] G. Scarpa, R. Gaetano, M. Haindl, and J. Zerubia. Hierarchical multiple markov chain model for unsupervised texture segmentation. *IEEE Transactions on Image Processing*, 18(8):1830 –1843, aug. 2009.

[40] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *IEEE International Conference on Computer Vision*, 2005.

[41] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[42] M. Stasolla and P. Gamba. Spatial indexes for the extraction of formal and informal human settlements from high-resolution sar images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 1(2):98 –106, june 2008.

[43] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[44] L. Vincent and E. Dougherty. *Digital Image Processing Methods*, chapter Morphological Segmentation for Textures and Particles, pages 43–102. CRC, 1994.

[45] D. Xin, Cheng, Hong, Yan, Xifeng, and J. Han. Extracting redundancy-aware top-k patterns. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 444–453, New York, NY, USA, 2006. ACM.

[46] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation patterns: from visual words to visual phrases. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1 –8, Jun 2007.

[47] D. Zamalieva. Unsupervised detection of compound strctures using image segmentation and graph-based texture analysis. Master's thesis, Bilkent University, August 2009.