Sparsity Analysis for Computer Vision Applications

CHENG BIN

NATIONAL UNIVERSITY OF SINGAPORE

2013

Sparsity Analysis for Computer Vision Applications

CHENG BIN

(B.Eng. (Electronic Engineering and Information Science), USTC)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2013

Acknowledgments

There are many people whom I wish to thank for the help and support they have given me throughout my Ph.D. study. My foremost thank goes to my supervisor Dr. Shuicheng Yan. I thank him for all the guidance, advice and support he has given me during my Ph.D. study at NUS. For the last four and half years, I have been inspired by his vision and passion to research, his attention and curiosity to details, his dedication to the profession, his intense commitment to his work, and his humble and respectful personality. During this most important period in my career, I thoroughly enjoyed working with him, and what I have learned from him will benefit me for my whole life.

I also would like to give my thanks to Dr. Bingbing Ni for all his kind help throughout all my Ph.D study. He is my brother forever. I also appreciate Dr. Loong Fah Cheong. His visionary thoughts and energetic working style have influenced me greatly.

I would also like to take this opportunity to thank all the students and staffs in Learning and Vision Group. During my Ph.D. study in NUS, I enjoyed all the vivid discussions we had and had lots of fun being a member of this fantastic group.

Last but not least, I would like to thank my parents for always being there when I needed them most, and for supporting me through all these years. I would especially like

to thank my girlfriend Huaxia Li, who with her unwavering support, patience, and love has helped me to achieve this goal. This dissertation is dedicated to them.

Contents

Summary

The research on sparse modeling has a long history. Recently research shows that sparse modeling appears to be biologically plausible as well as empirically effective in fields as diverse as computer vision, signal processing, natural language processing and machine learning. It has been proven to be an extremely powerful tool for acquiring, representing and compressing high-dimensional signals, and providing high performance for noise reduction, pattern classification, blind sourse separation and so on. In this dissertation, we study the sparse representations of high-dimensional signals for various learning and vision tasks, including graph learning, image segmentation and face recognition. The entire thesis is arranged into four parts.

In the first part, we investigate the graph construction by sparse modeling. An informative graph is critical for those graph-oriented algorithms designed for the purpose of data clustering, subspace learning, and semi-supervised learning. We model the graph construction problem, and propose a procedure to construct a robust and datum-adaptive ℓ^1-graph by encoding the overall behavior of the data set in sparse representation. The neighboring samples of a datum and the corresponding ingoing edge weights are simultaneously derived by solving an ℓ^1-norm optimization problem, where each datum is reconstructed by the linear combination of the remaining samples and noise item, with the objective of minimizing the ℓ^1 norm of both reconstruction coefficients and data

noise. It exhibits exceptionally performance in various graph-based applications.

We then study the label-to-region problem by sparse modeling in the second part. The ability to annotate images with related text labels at the semantic region-level is invaluable for boosting keyword based image search with the awareness of semantic image content. To address this label-to-region assignment problem, we propose to propagate the lables annotated at the image-level to those local semantic regions merged from the over-segmentation atomic image patches of the entire image set, by using a bi-layer sparse coding model. The underlying philosophy of bi-layer sparse coding is that an image or semantic region can be sparsely reconstructed via the atomic image patches belonging to the images with common labels, while the robustness in label propagation requires that these selected atomic patches come from very few images. Each layer of sparse coding produces the image label assignment to those selected atomic patches and merged candidate regions based on the shared image labels. Extensive experiments on three public image datasets clearly demonstrate the effectiveness of this algorithm.

In the third part, we implement the sparse modeling in face misalignment problem. Face recognition has been motivated by both its scientific values and potential applications in the practice of computer vision and machine learning. And face alignment is standard preprocessing step for recognition. Sometimes the practical system, or even manual face cropping, may bring considerable face misalignment problem. This discrepancy may inversely affect image similarity measurement, and consequently degrade face recognition performance. We develop a supervised sparse coding framework towards a practical solution to mislignment-robust face recognition. It naturally integrates the patch-based representation, supervised learning and sparse coding, and is superior to most conventional algorithms in term of algorithmic robustness.

To this end, we study the low-rank representation, an extension of sparse modeling,

and propose a multi-task low-rank affinity pursuit framework for image segmentation. Given an image described with multiple types of features, we aim at inferring a unified affinity matrix that implicitly encodes the segmentation of the image. This is achieved by seeking the sparsity-consistent low-rank affinities from the joint decompositions of multiple feature matrices into pairs of sparse and lowrank matrices, the latter of which is expressed as the production of the image feature matrix and its corresponding image affinity matrix. Experiments on the MSRC dataset and Berkeley segmentation dataset well validate the superiority of using multiple features over single feature and also the superiority of our method over conventional methods for feature fusion. Moreover, our method is shown to be very competitive while comparing to other state-of-the-art methods.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Sparse Representation

Recently, sparse signal representation has gained a lot of interests from various research areas in information science. It accounts for most or all of the information of a signal by a linear combination of a small number of elementary signals called atoms in a basis or an over-complete dictionary, and has increasingly been recognized as providing high performance for applications as diverse as noise reduction, compression, inpainting, compressive sensing, pattern classification, and so on. Suppose we have an underdetermined system of linear equations: $x = D\alpha$, where $x \in \mathbb{R}^m$ is the vector to be approximated, $\alpha \in \mathbb{R}^n$ is the vector for unknown reconstruction coefficients, and $D \in \mathbb{R}^{m \times n} (m < n)$ is the overcomplete dictionary with n bases. Generally, a sparse solution is more robust and facilitate the consequent identification of the test sample x. This motivates us to seek the sparest solution to $x = D\alpha$ by solving the following optimization problem:

$$\min_{\alpha} \|\alpha\|_0, \quad \textit{s.t.} \;\; x = D\alpha. \tag{1.1}$$

where $\|\cdot\|_0$ denotes the ℓ^0-norm, which counts the number of nonzero entries in a vector. One natural variation is to relax the equality constraint to allow some error tolerance $\epsilon \geq 0$, where the signal is contrminated with noise

$$\min_{\alpha} \|\alpha\|_0, \quad \textit{s.t.} \;\; \|D\alpha - x\|_2 \leq \epsilon. \tag{1.2}$$

However, solving this sparse representation problem directly is combinatorially NP-hard in general case, and difficult even to approximate. In the past several years, there have been exciting breakthroughs in the study of high dimensional sparse signals. Recent results [4][5] show that if the solution is sparse enough, the sparse representation can be recovered by the following convex ℓ^1-norm minimization [4],

$$\min_{\alpha} \|\alpha\|_1, \quad \textit{s.t.} \;\; x = D\alpha. \tag{1.3}$$

or

$$\min_{\alpha} \|\alpha\|_1, \quad \textit{s.t.} \;\; \|D\alpha - x\|_2 \leq \epsilon. \tag{1.4}$$

In the concrete sense, the ℓ^1-norm is the tighest convex relaxation for the ℓ^0-norm. And this optimization problem can be transformed into a general linear programming problem. There exists a globally optimal solution, and the optimization can be solved

efficiently by standard linear programming method [6]. In practice, there may exist noises on certain elements of x, and a natural way to recover these elements and provide a robust estimation of α is to formulate

$$x = D\alpha + \zeta = \begin{bmatrix} D & I \end{bmatrix} \begin{bmatrix} \alpha \\ \zeta \end{bmatrix}, \tag{1.5}$$

where $\zeta \in \mathbb{R}^m$ is the noise term. Then by setting $B = \begin{bmatrix} D & I \end{bmatrix} \in \mathbb{R}^{m \times (m+n)}$ and $\alpha' = \begin{bmatrix} \alpha \\ \zeta \end{bmatrix}$, we can solve the following ℓ^1-norm minimization problem with respect to both reconstruction coefficients and data noises,

$$\min_{\alpha'} \|\alpha'\|_1, \;\; \textit{s.t.} \;\; x = B\alpha', \tag{1.6}$$

Sparse representation has proven to be an extremely powerful tool for acquiring, representing, and compressing high dimensional signals. In the more general sense, sparsity constraints have emerged as a fundamental type of regularizer for many ill-conditioned or under-determined linear inverse problems. In the past several years, variations and extensions of sparsity promoting ℓ^1-norm minimization have been applied to many vision and machine learning tasks, such as face recognition [5, 7], human action recognition [8], image classification [9, 10, 11], background modeling [12], and bioinformatics [13].

1.2 Thesis Focus and Main Contributions

In this dissertation, we will explore several different areas in computer vision and machine learning based on sparse modeling.

During our research on sparse modeling, we did a lot of experiments and found that it has the following advantages:

1) Sparse modeling is much more robust than the Euclidean distance based modeling (shown in Figure 2.1);

2) Sparse modeling has the potential to connect kindred samples, and hence may potentially convey more discriminative information (shown in Figure 2.2).

These advantages make it very suitable for graph construction. So in the first work, we apply the sparse modeling to graph construction and derive various machine learning tasks upon the graph.

1) **Learning with L1-Graph for Image Analysis:** The graph construction procedure essentially determines the potentials of those graph-oriented learning algorithms for image analysis. In this work, we propose a process to build the so-called directed ℓ^1-graph, in which the vertices involve all the samples and the ingoing edge weights to each vertex describe its ℓ^1-norm driven reconstruction from the remaining samples and the noise. Then, a series of new algorithms for various machine learning tasks, e.g., data clustering, subspace learning, and semi-supervised learning, are derived upon the ℓ^1-graphs. Compared with the conventional k-nearest-neighbor graph and ϵ-ball graph, the ℓ^1-graph possesses the advantages: 1) greater robustness to data noise, 2) automatic sparsity, and 3) adaptive neighborhood for individual datum. Extensive experiments on three real-world datasets show the consistent superiority of ℓ^1-graph over those classic graphs in data clustering, subspace learning, and semi-supervised learning tasks.

In this work, we constructed the graph by sparse modeling and applied it to unsupervised learning. Then naturally how to combine the label information and extend the sparse coding to supervised learning became a very interesting problem for me. Also during the experiments, we found that sparse modeling works well on face recognition when faces are well aligned, while yields poor performance on misaligned face images. Addressing these two problems, we move to our second work as follows:

2) **Supervised Sparse Coding Towards Misalignment-Robust Face Recognition:** We address the challenging problem of face recognition under the scenarios where both training and test data are possibly contaminated with spatial misalignments. A *supervised* sparse coding framework is developed in this work towards a practical solution to misalignment-robust face recognition. Each gallery face image is represented as a set of patches, in both original and misaligned positions and scales, and each given probe face image is then uniformly divided into a set of local patches. We propose to *sparsely* reconstruct each probe image patch from the patches of all gallery images, and at the same time the reconstructions for all patches of the probe image are regularized by one term towards enforcing *sparsity* on the subjects of those selected patches. The derived reconstruction coefficients by ℓ_1-norm minimization are then utilized to fuse the subject information of the patches for identifying the probe face. Such a supervised sparse coding framework provides a unique solution to face recognition with all the following four characteristics: 1) the solution is model-free, without the model learning process, 2) the solution is robust to spatial misalignments, 3) the solution is robust to image occlusions, and 4) the solution is effective even when there exist spatial misalignments for gallery images. Extensive face recognition experiments on three benchmark face datasets demonstrate the advantages of the proposed framework over holistic

sparse coding and conventional subspace learning based algorithms in terms of robustness to spatial misalignments and image occlusions.

In this work, we used the patch reconstruction and dual sparsity for misaligned face recognition problem. These two methods are as important as the basis for all my following works.

Now we have applied the sparse modeling for two classical problems both on face images. During the research, we kept thinking that whether we can apply the sparse modeling to the real-world image anlaysis. So in the third work, based on patch reconstruction and dual sparsity, we explore the sparse modeling on real-world images as follows:

3) **Label to Region by Bi-Layer Sparsity Priors:** In this work, we investigate how to automatically reassign the manually annotated labels at the image-level to those contextually derived semantic regions. First, we propose a bi-layer sparse coding formulation for uncovering how an image or semantic region can be robustly reconstructed from the over-segmented image patches of an image set. We then harness it for the automatic label to region assignment of the entire image set. The solution to bi-layer sparse coding is achieved by convex ℓ^1-norm minimization. The underlying philosophy of bi-layer sparse coding is that an image or semantic region can be sparsely reconstructed via the atomic image patches belonging to the images with common labels, while the robustness in label propagation requires that these selected atomic patches come from very few images. Each layer of sparse coding produces the image label assignment to those selected atomic patches and merged candidate regions based on the shared image labels. The results from all bi-layer sparse codings over all candidate regions are then fused

to obtain the entire label to region assignments. Besides, the presenting bi-layer sparse coding framework can be naturally applied to perform image annotation on new test images. Extensive experiments on three public image datasets clearly demonstrate the effectiveness of our proposed framework in both label to region assignment and image annotation tasks.

The label-to-region problem can be considered as a variate of image segmentation. After this work, Prof. Yi MA presented a new extension on sparse modeling: Robust PCA, where an observed data matrix D can naturally be modeled as a low-rank contribution A plus a sparse contribution E. All the statistical applications, in which robust principle components are sought, of course fit the model. In the third work, we found that common regions may share some common features, which is very suitable for Robust PCA model. So in the fourth work, we combine Robust PCA and graph learning, and provide a new framework for region-based image segmentation.

4) **Multi-task Low-rank Affinity Pursuit for Image Segmentation:** This work investigates how to boost region-based image segmentation by pursuing a new solution to fuse multiple types of image features. A collaborative image segmentation framework, called multi-task low-rank affinity pursuit, is presented for such a purpose. Given an image described with multiple types of features, we aim at inferring a unified affinity matrix that implicitly encodes the segmentation of the image. This is achieved by seeking the sparsity-consistent low-rank affinities from the joint decompositions of multiple feature matrices into pairs of sparse and low-rank matrices, the latter of which is expressed as the production of the image feature matrix and its corresponding image affinity matrix. The inference process is formulated as a constrained nuclear norm and $\ell_{2,1}$-norm minimization problem, which is convex and can be solved efficiently with the Augmented Lagrange

Multiplier method. Compared to previous methods, which are usually based on a single type of features, the proposed method seamlessly integrates multiple types of features to jointly produce the affinity matrix within a single inference step, and produces more accurate and reliable segmentation results. Experiments on the MSRC dataset and Berkeley segmentation dataset well validate the superiority of using multiple features over single feature and also the superiority of our method over conventional methods for feature fusion. Moreover, our method is shown to be very competitive while comparing to other state-of-the-art methods.

1.3 Organization of the Thesis

The reminder of the thesis is organized as follows. Chapter 2 explores the sparse representation for signal space modeling and presents a graph construction procedure with explicit sparsity constraint. Chapter 3 discusses the face misalignment problem and develops a supervised sparse coding framework towards a practical solution to misalignment-robust face recognition. Chapter 4 introduces the label-to-region problem and provide a bi-layer sparse coding model to solve this problem. As all these applications are based on the sparse representation, in Chapter 5 we extend the model to low-rank representation and implement it in image segmentation. Finally, Chapter 6 summarizes this dissertation with discussions for future exploration.

Chapter 2

Learning with L1-Graph for Image Analysis

2.1 Introduction

An informative graph, directed or undirected, is critical for those graph-oriented algorithms designed for the purposes of data clustering, subspace learning, and semi-supervised learning. Data clustering often starts with a pairwise similarity graph and is then transformed into a graph partition problem [14]. The pioneering works on manifold learning, *e.g.*, ISOMAP [15], Locally Linear Embedding [16], and Laplacian Eigenmaps [17], all rely on graphs constructed in different ways. Moreover, most popular subspace learning algorithms, *e.g.*, Principal Component Analysis [18], Linear Discriminant Analysis [19], and Locality Preserving Projections [20], can all be explained within the graph embedding framework as claimed in [21]. Also, most semi-supervised learning algorithms are driven by certain graphs constructed over both labeled and unlabeled

data. Zhu et al. [22] utilized the harmonic property of Gaussian random field over the graph for semi-supervised learning. Belkin and Niyogi [23] instead learned a regression function that fits the labels at labeled data and also maintains smoothness over the data manifold expressed by a graph.

There exist two popular ways for graph construction, one of which is the k-nearest-neighbor method, and the other is the ϵ-ball based method, where, for each datum, the samples within its surrounding ϵ ball are connected, and then various approaches, *e.g.*, binary, Gaussian-kernel [17] and ℓ^2-reconstruction [16], can be used to further set the graph edge weights. Since the ultimate purposes of the constructed graphs are for data clustering, subspace learning, semi-supervised learning, etc., the following graph characteristics are desired:

1) **Robustness to data noise**. The data noises are inevitable especially for visual data, and the robustness is a desirable property for a satisfactory graph construction method. The graph constructed by k-nearest-neighbor or ϵ-ball method is founded on pair-wise Euclidean distance, which is very sensitive to data noise. It means that the graph structure is easy to change when unfavorable noises come in.

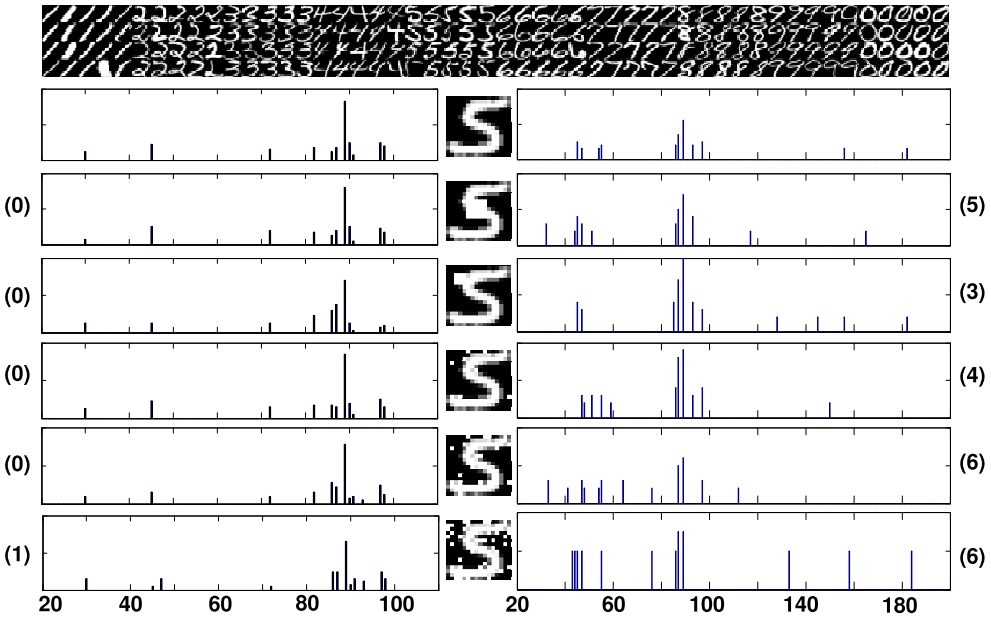
2) **Sparsity**. Recent research on manifold learning [17] shows that sparse graph characterizing locality relations can convey valuable information for classification purpose. Also for applications with large scale data, a sparse graph is the inevitable choice due to the storage limitation.

3) **Datum-adaptive neighborhood**. Another observation is that the data distribution probability may vary greatly at different areas of the feature space, which results in distinctive neighborhood structure for each datum. Both k-nearest-neighbor and ϵ-ball methods however use a fixed global parameter to determine the neighborhoods

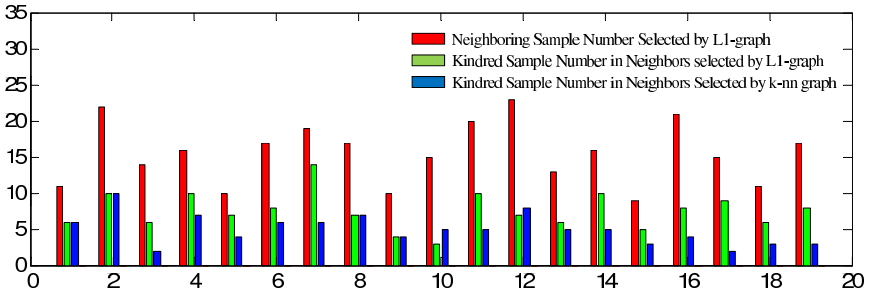
for all the data, and hence fail to offer such datum-adaptive neighborhoods.

We present in Section-2.2 a procedure to construct robust and datum-adaptive ℓ^1-graph by utilizing the overall contextual information instead of only pairwise Euclidean distance as conventionally. The neighboring samples of a datum and the corresponding ingoing edge weights are simultaneously derived by solving an ℓ^1-norm optimization problem, where each datum is reconstructed by the linear combination of the remaining samples and noise item, with the objective of minimizing the ℓ^1 norm of both reconstruction coefficients and data noise. Compared with the graphs constructed by k-nearest-neighbor and ϵ-ball methods, the ℓ^1-graph has the following three advantages. First, ℓ^1-graph is robust owing to the overall contextual ℓ^1-norm formulation and the explicit consideration of data noises. Figure 2.1(a) shows the graph robustness comparison between ℓ^1-graph and k-nearest-neighbor graph. Second, the sparsity of the ℓ^1-graph is automatically determined instead of manually as in k-nearest-neighbor and ϵ-ball methods. Finally, the ℓ^1-graph is datum-adaptive. As shown in Figure 2.1(b), the number of neighbors selected by ℓ^1-graph is adaptive to each datum, which is valuable for applications with unevenly distributed data.

This ℓ^1-graph is then utilized in Section-2.3 to instantiate a series of graph-oriented algorithms for various machine learning tasks, *e.g.*, data clustering, subspace learning, and semi-supervised learning. Owing to the above three advantages over classical graphs, ℓ^1-graph brings consistent performance gain in all these tasks as detailed in Section-2.4.



(a) Neighbor robustness comparison of ℓ^1-graph and k-nn graph



(b) Datum-adaptive neighbor numbers selected by sparse ℓ^1-graph, and kindred neighbor numbers for ℓ^1-graph and k-nn graph

Figure 2.1: Robustness and adaptiveness comparison for neighbors selected by ℓ^1-graph and k-nn graph. (a) Illustration of basis samples (1st row), reconstruction coefficient distribution in ℓ^1-graph (left), samples to reconstruct (middle, with added noises from the third row on), and similarity distribution of the k nearest neighbors selected with Euclidean distance (right) in k-nn graph. Here the horizontal axes indicate the index number of the training samples. The vertical axes of the left column indicate the reconstruction coefficient distribution for all training samples in sparse coding, and those of right column indicate the similarity value distribution of k nearest neighbors. Note that the number in parenthesis is the number of neighbors changed compared with results in the second row, and ℓ^1-graph shows much more robust to image noises. (b) Neighboring samples comparison between ℓ^1-graph and k-nn graph. The red bars indicate the numbers of the neighbors selected by ℓ^1-graph automatically and adaptively. The green bars indicate the numbers of kindred samples among the k neighbors selected by ℓ^1-graph. And the blue bar indicate the numbers of kindred samples within the k nearest neighbors measured by Euclidean distance in k-nn graph. Note that the results are obtained on USPS digit database [1] and the horizontal axis indicates the index of the reference sample to reconstruct.

2.2 Rationales on ℓ^1-graph

For a general data clustering or classification problem, the training sample set is assumed being represented as a matrix $X = [x_1, x_2, \ldots, x_N], x_i \in \mathbb{R}^m$, where N is the sample number and m is the feature dimension. For supervised learning problems, the class label of the sample x_i is then assumed to be $l_i \in \{1, 2, \ldots, N_c\}$, where N_c is the total number of classes.

2.2.1 Motivations

The ℓ^1-graph is motivated by the limitations of classical graph construction methods [17][16] in robustness to data noise and datum-adaptiveness, and recent advances in sparse coding [4][24][5]. Note that a graph construction process includes both sample neighborhood selection and graph edge weight setting, which are assumed in this work to be unsupervised, without harnessing any data labelinformation.

The approaches of k-nearest-neighbor and ϵ-ball are very popular for graph construction in literature. Both of them determine the neighboring samples based on *pairwise* Euclidean distance, which is however very sensitive to data noises and one noisy feature may dramatically change the graph structure. Also when the data are not evenly distributed, the k nearest neighbors of a datum may involve faraway inhomogeneous data if the k is set too large, and the ϵ-ball may involve only single isolated datum if ϵ is set too small. Moreover, the optimum of k (or ϵ) is datum-dependent, and one single global parameter may result in unreasonable neighborhood structure for certain datum.

The research on sparse coding or sparse representation has a long history. Recent research shows that sparse coding appears to be biologically plausible as well as em-

pirically effective for image processing and pattern classification [5]. Olshausen et al. [25] employed the Bayesian models and imposed ℓ^1 priors for deducing the sparse representation, and Wright et al. [5] proposed to use sparse representation for direct face recognition. In this work, beyond the sparse coding for individual test datum, we are interested in the overall behavior of the whole sample set in sparse representation, and then present the general concept of ℓ^1-graph, followed by its applications in various machine learning tasks, *e.g.*, data clustering, subspace learning, and semi-supervised learning.

2.2.2 Robust Sparse Representation

Much interest has been shown in computing linear sparse representation with respect to an overcomplete dictionary of the basis elements. Suppose we have an underdetermined system of linear equations: $x = D\alpha$, where $x \in \mathbb{R}^m$ is the vector to be approximated, $\alpha \in \mathbb{R}^n$ is the vector for unknown reconstruction coefficients, and $D \in \mathbb{R}^{m \times n}(m < n)$ is the overcomplete dictionary with n bases. Generally, a sparse solution is more robust and facilitate the consequent identification of the test sample x. This motivates us to seek the sparest solution to $x = D\alpha$ by solving the following optimization problem:

$$\min_{\alpha} \|\alpha\|_0, \quad \textit{s.t.} \;\; x = D\alpha. \tag{2.1}$$

where $\|\cdot\|_0$ denotes the ℓ^0-norm, which counts the number of nonzero entries in a vector. But It is well known that the sparsest representation problem is NP-hard in general case, and difficult even to approximate. However, recent results [4][5] show that if the solution is sparse enough, the sparse representation can be recovered by the following convex ℓ^1-norm minimization [4],

$$\min_{\alpha} \|\alpha\|_1, \quad \textit{s.t.} \;\; x = D\alpha. \tag{2.2}$$

This problem can be solved in polynomial time by standard linear programming method [6]. In practice, there may exist noises on certain elements of x, and a natural way to recover these elements and provide a robust estimation of α is to formulate

$$x = D\alpha + \zeta = \begin{bmatrix} D & I \end{bmatrix} \begin{bmatrix} \alpha \\ \zeta \end{bmatrix}, \tag{2.3}$$

where $\zeta \in \mathbb{R}^m$ is the noise term. Then by setting $B = \begin{bmatrix} D & I \end{bmatrix} \in \mathbb{R}^{m \times (m+n)}$ and $\alpha' = \begin{bmatrix} \alpha \\ \zeta \end{bmatrix}$, we can solve the following ℓ^1-norm minimization problem with respect to both reconstruction coefficients and data noises,

$$\min_{\alpha'} \|\alpha'\|_1, \quad \textit{s.t.} \;\; x = B\alpha', \tag{2.4}$$

This optimization problem is convex and can be transformed into a general linear programming problem. There exists a globally optimal solution, and the optimization can be solved efficiently using many available ℓ^1-norm optimization toolboxes like [26]. Note that the ℓ^1 norm optimization toolbox in [26] may convert the original constrained optimization problem into an unconstrained one, with an extra regularization coefficient which can be tuned for optimum in practice but essentially does not exist in original problem formulation.

2.2.3 ℓ^1-graph Construction

An ℓ^1-graph summarizes the overall behavior of the whole sample set in sparse representation. The construction process is formally stated as follows.

1) **Inputs:** The sample data set denoted as the matrix $X = [x_1, x_2, ..., x_N]$, where $x_i \in \mathbb{R}^m$.

2) **Robust sparse representation:** For each datum x_i in the sample set, its robust sparse coding is achieved by solving the ℓ^1-norm optimization problem

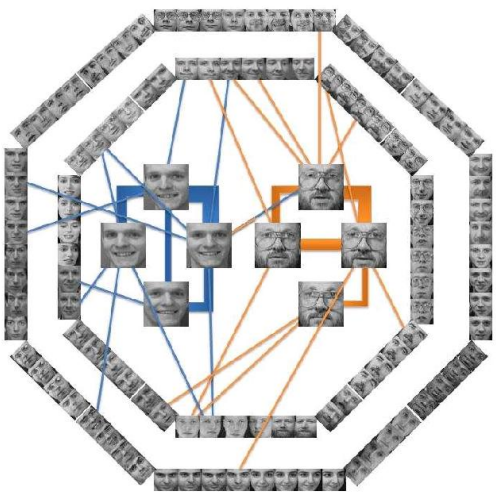
$$\min_{\alpha^i} \|\alpha^i\|_1, \quad \textit{s.t.} \quad x_i = B^i \alpha^i, \tag{2.5}$$

where matrix $B^i = [x_1, ..., x_{i-1}, x_{i+1}, ..., x_N, I] \in \mathbb{R}^{m \times (m+N-1)}$ and $\alpha^i \in \mathbb{R}^{m+N-1}$.

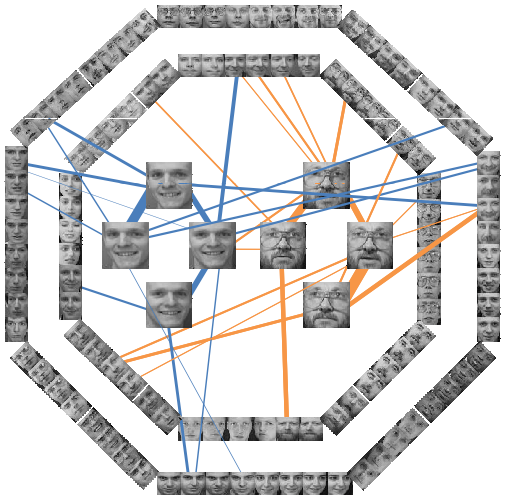
3) **Graph weight setting:** Denote $\mathrm{G} = \{X, W\}$ as the ℓ^1-graph with the sample set X as graph vertices and W as the graph weight matrix, and we set $W_{ij} = \alpha^i_j$ (nonnegativity constraints may be imposed for α^i_j in optimization if for similarity measurement) if $i > j$, and $W_{ij} = \alpha^i_{j-1}$ if $i < j$.

Figure 2.2 depicts partial of the ℓ^1-graphs based on the data from the YALE-B face database [27] and USPS digit database [1] respectively. An interesting observation from Figure 2.2 is that, besides being robust and datum-adaptive, the ℓ^1-graph has the potential to connect kindred samples, and hence may potentially convey more discriminative information, which is valuable for its later introduced applications in data clustering, subspace learning, and semi-supervised learning. Taking the face image as an example, the intuition behind the observed discriminating power of ℓ^1 graph is that, if one expects to reconstruct a face image with all other face images as bases, the most efficient way in terms of the number of relevant bases is to use similar images or images from the same subject, which leads to a sparse solution and coincides with the empirical observations in [5] for robust face recognition with sparse representation.

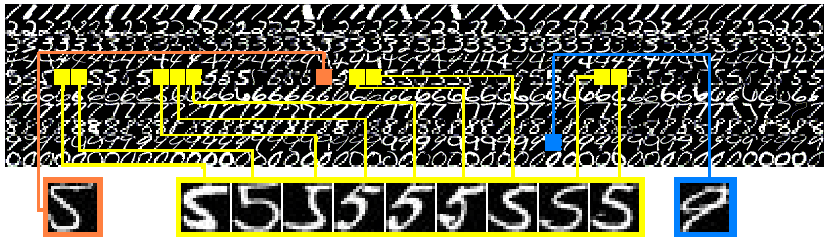
Discussions: 1) Note that the formulation in (2.4) is based on the assumption that the feature dimension, m, is reasonably large, otherwise the sparsity of noises shall make no



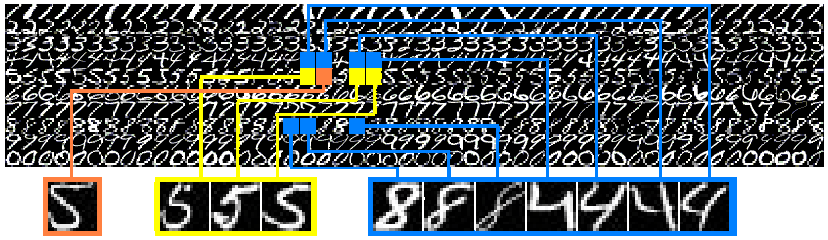
(a) Example of ℓ^1-graph



(b) Example k-nearest-neighbor graph



(i) Neighbors selected by L1-graph robustly and contextually



(ii) Pair-distance based k nearest neighbors with many outliers

(c) Example ℓ^1-graph and k-NN graph

Figure 2.2: Visualization comparison of (a) the ℓ^1-graph and (b) the k-nn graph, where the k for each datum is automatically selected in the ℓ^1-graph. Note that the thickness of the edge line indicates the value of the edge weight (Gaussian kernel weight for k-nn graph). For ease of display, we only show the graph edges related to the samples from two classes and in total 30 classes from the YALE-B database are used for graph construction. (c) Illustration on the positions of a reference sample (red), its kindred neighbors (yellow), and its inhomogeneous neighbors (blue) selected by (i) ℓ^1-graph and (ii) k-nearest-neighbor method based on samples from the USPS [1].

sense. It means that (2.4) is not applicable for simple 2-dimensional or 3-dimensional toy data. 2) In implementation, the data normalization, *i.e.*, $||x_i||_2 = 1$, is critical for deriving semantically reasonable coefficients. 3) The k-nearest-neighbor graph is flexible in terms of the selection of similarity/distance measurement, but the optimality is heavily data dependent. In this work, we use the most conventional Euclidean distance for selecting the k nearest neighbors. 4) For certain extreme cases, *e.g.*, if we simply duplicate each sample and generate another new dataset of double size, ℓ^1-graph may only connect these duplicated pairs, and thus fail to convey valuable information. A good observation is that these extreme cases are very rare for those datasets investigated in general research.

2.3 Learning with ℓ^1-graph

An informative graph is critical for those graph-oriented learning algorithms. Similar to classical graphs constructed by k-nearest-neighbor or ϵ-ball method, ℓ^1-graph can be integrated with various learning algorithms for various tasks, *e.g.*, data clustering, subspace learning, and semi-supervised learning. In this section, we briefly introduce how to benefit from ℓ^1-graph for these tasks.

2.3.1 Spectral Clustering with ℓ^1-graph

Data clustering is the classification of samples into different groups, or more precisely, the partition of samples into subsets, such that the data within each subset are similar to each other. The spectral clustering [14] is among the most popular algorithms for this task, but there exists one parameter δ [14] for controlling the similarity between a data pair. Intuitively the contribution of one sample to the reconstruction of another sample

is a good indicator of similarity between these two samples, we decide to use the reconstruction coefficients to constitute the similarity graph for spectral clustering. As the weights of the graph are used to indicate the similarities between different samples, they should be assumed to be non-negative. Using the ℓ^1-graph, the algorithm can automatically select the neighbors for each datum, and at the same time the similarity matrix is automatically derived from the calculation of these sparse representations. The detailed spectral clustering algorithm based on ℓ^1-graph is listed as follows.

1) Symmetrize the graph similarity matrix by setting the matrix $W = (W + W^T)/2$.

2) Set the graph Laplacian matrix $L = D^{-1/2}WD^{-1/2}$, where $D = [d_{ij}]$ is a diagonal matrix with $d_{ii} = \sum_j w_{ij}$.

3) Find $c_1, c_2, \cdots, c_K$, the eigenvectors of L corresponding to the K largest eigenvalues, and form the matrix $C = [c_1, c_2, \cdots, c_K]$ by stacking the eigenvectors in columns.

4) Treat each row of C as a point in $\mathbb{R}^K$, and cluster them into K clusters via the K-means method.

5) Finally, assign x_i to the cluster j if the i-th row of the matrix C is assigned to the cluster j.

2.3.2 Subspace Learning with ℓ^1-graph

Similar to the graph construction process in Locally Linear Embedding (LLE), the ℓ^1-graph characterizes the neighborhood reconstruction relationship. In LLE, the graph is constructed by reconstructing each datum with its k nearest neighbors or the samples

within the ϵ-ball based on the ℓ^2-norm. LLE and its linear extension, called Neighborhood Preserving Embedding (NPE) [28], both rely on the global graph parameter (k or ϵ). Following the idea of NPE algorithm, ℓ^1-graph can be used to develop a subspace learning algorithm as follows.

The general purpose of subspace learning is to search for a transformation matrix $P \in \mathbb{R}^{m \times d}$ (usually $d \ll m$) for transforming the original high-dimensional datum into another low-dimensional one. ℓ^1-graph uncovers the underlying sparse reconstruction relationship of each datum, and it is desirable to preserve these reconstruction relationships in the dimensionality reduced feature space. Note that in the dimension reduced feature space, the reconstruction capability is measured by ℓ^2 norm instead of ℓ^1 norm for computational efficiency. Then the pursue of the transformation matrix can be formulated as the optimization

$$\min_{P^T X X^T P = I} \sum_{i=1}^{N} \|P^T x_i - \sum_{j=1}^{N} W_{ij} P^T x_j\|^2, \tag{2.6}$$

where W_{ij} is determined by the constructed ℓ^1-graph. This optimization problem can be solved with generalized eigenvalue decomposition approach as

$$X M X^T p_{m+1-j} = \lambda_j X X^T p_{m+1-j}, \tag{2.7}$$

where $M = (I - W)^T (I - W)$, and p_{m+1-j} is the eigenvector corresponding to the j-th largest eigenvalue λ_j as well as the $(m + 1 - j)$-th column vector of the matrix P.

The derived transformation matrix is then used for dimensionality reduction as

$$y_i = P^T x_i, \tag{2.8}$$

where y_i is the corresponding low-dimensional representation of the sample x_i and finally the classification process is performed in this low-dimensional feature space with reduced computational cost.

2.3.3 Semi-supervised Learning with ℓ^1-graph

As shown in Figure 2.1 and Figure 2.2, the ℓ^1-graph is robust to data noises and datum-adaptive, also empirically has the potential to convey more discriminative information compared with conventional graphs based on k-nearest-neighbor or ϵ-ball method. These properties make ℓ^1-graph a good candidate for propagating the label information over the graph. Semi-supervised learning recently has attracted much attention, and was widely used for both regression and classification purposes. The main idea of semi-supervised learning is to utilize unlabeled data for improving the classification and generalization capability on the testing data. Commonly the unlabeled data are used as an extra regularization term to the objective functions from traditional supervised learning algorithms.

In this work, the unlabeled data are used to enlarge the vertex number of the ℓ^1-graph, and further enhance the robustness of the graph. Finally the ℓ^1-graph based on both labeled and unlabeled data is used to develop semi-supervised learningalgorithm. Here, we take Marginal Fisher Analysis (MFA) [21] as an example for the supervised part in semi-supervised learning. Similar to the philosophy in [29], the objective for ℓ^1-graph based semi-supervised learning is defined as

$$\min_P \frac{\gamma S_c(P) + (1-\gamma)\sum_{i=1}^N \|P^T x_i - \sum_{j=1}^N W_{ij} P^T x_j\|^2}{S_p(P)},$$

where $\gamma \in (0,1)$ is a threshold for balancing the supervised term and ℓ^1-graph regular-

ization term, and the supervised part is defined as

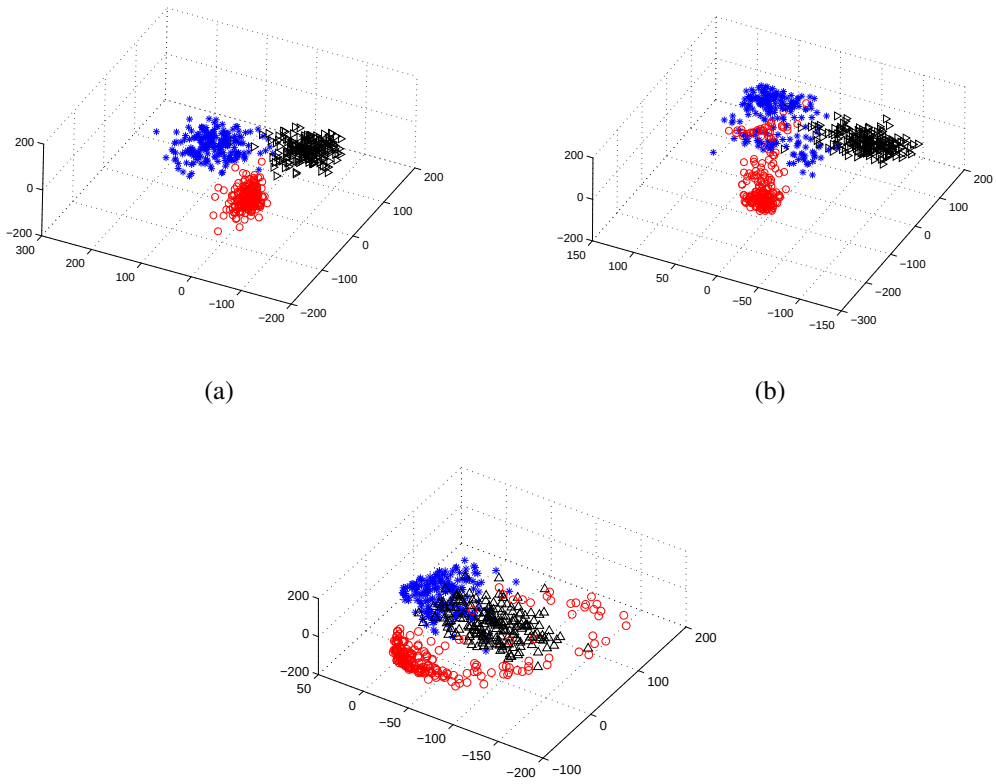
$$S_c(P) = \sum_i \sum_{j \in N_{k_1}^+(i)} ||P^T x_i - P^T x_j||^2, \tag{2.9}$$

$$S_p(P) = \sum_i \sum_{(i,j) \in P_{k_2}(l_i)} ||P^T x_i - P^T x_j||^2, \tag{2.10}$$

where S_c indicates the intra-class compactness, which is represented as the sum of distances between each point and its neighbors of the same class and $N_{k_1}^+(i)$ is the index set of the k_1 nearest neighbors of the sample x_i in the same class, S_p indicates the separability of different classes, which is characterized as the sum of distances between the marginal points and their neighboring points of different classes and $P_{k_2}(l)$ is a set of data pairs that are the k_2 nearest pairs among the set $\{(i,j), l_i = l, l_j \neq l\}$, and W is the weight matrix of the ℓ^1-graph. Similar to (2.6), the optimum can be obtained via the generalized eigenvalue decomposition method, and the derived projection matrix P is then used for dimensionality reduction and consequent data classification.

2.4 Experiments

In this section, we systematically evaluate the effectiveness of ℓ^1-graph in three learning tasks, namely, data clustering, subspace learning, and semi-supervised learning. For comparison purpose, the classical k-nearest-neighbor graph and ϵ-ball graph with different graph weighting approaches are implemented as evaluation baselines. Note that for all k-near-neighbor graph and ϵ-ball graphs related algorithms, the reported results are based on the tuned best k and ϵ among all proper values.



(a) (b)

(c)

Figure 2.3: Visualization of the data clustering results from (a) ℓ^1-graph, (b) LE-graph, and (c) PCA algorithm for three clusters (handwritten digits 1, 2 and 3 in the USPS database). The coordinates of the points in (a) and (b) are obtained from the eigenvalue decomposition in the 3^{rd} step of Section-2.3.1. Different colors of the points indicate different digits. For better viewing, please see the color pdf file.

2.4.1 Data Sets

For all the experiments, three databases are used. The USPS handwritten digit database [1] includes 10 classes (0-9 digit characters) and 11000 samples in total. We randomly select 200 samples each digit character for the experiments, and all of these images are normalized to the size of 32-by-32 pixels. The forest covertype database [30] was collected for predicting forest cover type from cartographic variables. It includes seven classes and 581012 samples in total. We randomly select 100 samples for each type in

the following experiments. The Extended YALE-B database [27] contains 38 individuals and around 64 near frontal images under different illuminations per individual, where each image is manually cropped and normalized to the size of 32-by-32 pixels. All the images were taken against a dark homogeneous background with the subjects in an upright and frontal position.

2.4.2 Spectral Clustering with ℓ^1-graph

In this part of experiments, for a comprehensive evaluation, the ℓ^1-graph based spectral clustering algorithm is compared with the spectral clustering based on the Gaussian-kernel [14] graph, LE-graphs (used in Laplacian Eigenmaps [17] algorithm), LLE-graphs (ℓ^2-norm based and used in LLE [16]), and also the K-means clustering results based on the derived low-dimensional representations from Principal Component Analysis (PCA) [18]. And two metrics, the accuracy (AC) and the normalized mutual information (NMI) [31], are used for performance evaluation. Suppose that L is the clustering result label vector and $\hat{L}$ is the known sample label vector, AC is defined as

$$AC = \frac{\sum_{i=1}^{N} \delta(\hat{L}(i), Map_{(L,\hat{L})}(i))}{N} \tag{2.11}$$

where N denotes the total number of samples, $\delta(a, b)$ equals to 1 if and only if $a = b$, $Map_{(L,\hat{L})}$ is the best mapping function that permutes X to match Y, where X and Y are the index sets involving all values in L and $\hat{L}$ respectively. The Kuhn-Munkres algorithm is used to obtain the best mapping [6]. On the other hand, the mutual information between X and Y is defined as

$$MI(X,Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log_2(\frac{p(x,y)}{p(x)p(y)}), \tag{2.12}$$

where $p(x)$, $p(y)$ denote the marginal probability distribution functions of X and Y respectively, and $p(x,y)$ is the joint probability distribution function of X and Y. Suppose $H(X)$ and $H(Y)$ denote the entropies of $p(x)$ and $p(y)$. $MI(X,Y)$ varies between 0 and $max(H(X),H(Y))$. So we use normalized mutual information NMI as the second metric, namely,

$$NMI(X,Y) = \frac{MI(X,Y)}{max(H(X),H(Y))}. \tag{2.13}$$

It is obvious that the normalized mutual information NMI takes values in $[0,1]$. Unlike AC, NMI is invariant with the permutation of labels, namely, NMI does not require the matching X and Y in advance.

The visualization comparison of the data clustering results (digit characters 1-3 from the USPS database) based on ℓ^1-graph and those based on LE-graph and *K*-means are depicted in Figure 2.3, which shows that the data are much better separated in ℓ^1-graph. The quantitative comparison results on clustering accuracy are listed in Table 2.1-2.3 for these three databases respectively. From the listed results, three observations can be made: 1) the clustering results from ℓ^1-graph based spectral clustering algorithm are consistently much better than those from all other evaluated algorithms for both metrics; 2) (k-nn + LLE)-graph based spectral clustering algorithm is relatively more stable compared with other ones; and 3) ϵ-ball based algorithms show to be generally worse, in both accuracy and robustness, than the corresponding k-nn based graphs, and thus for the consequent experiments, we only report the results from k-nn graphs instead. Note

Table 2.1: Clustering accuracies (normalized mutual information/NMI and accuracy/AC) for spectral clustering algorithms based on ℓ^1-graph, Gaussian-kernel graph (G-graph), LE-graphs, and LLE-graphs, as well as PCA+K-means on the USPS digit database. Note that 1) the values in the parentheses are the best algorithmic parameters for the corresponding algorithms and for the parameters for AC are set as those with the best results for NIM, and 2) the cluster number K also indicates the class number used for experiments, that is, we use the first K classes in the database for the corresponding data clustering experiments.

USPS	Metric	ℓ^1-graph	G-graph	LE-graph		LLE-graph		PCA+
Cluster #				k-nn	ϵ-ball	k-nn	ϵ-ball	K-means
$K=2$	NMI	**1.000**	0.672(110)	0.858(7)	0.627(3)	0.636(5)	0.717(4)	0.608(10)
	AC	**1.000**	0.922	0.943	0.918	0.917	0.932	0.905
$K=4$	NMI	**0.977**	0.498(155)	0.693(16)	0.540(6)	0.606(5)	0.465(7)	0.621(20)
	AC	**0.994**	0.663	0.853	0.735	0.777	0.668	0.825
$K=6$	NMI	**0.972**	0.370(120)	0.682(5)	0.456(6)	0.587(5)	0.427(9)	0.507(4)
	AC	**0.991**	0.471	0.739	0.594	0.670	0.556	0.626
$K=8$	NMI	**0.945**	0.358(150)	0.568(7)	0.371(4)	0.544(12)	0.404(7)	0.462(17)
	AC	**0.981**	0.423	0.673	0.453	0.598	0.499	0.552
$K=10$	NMI	**0.898**	0.346(80)	0.564(6)	0.424(5)	0.552(16)	0.391(4)	0.421(10)
	AC	**0.873**	0.386	0.578	0.478	0.537	0.439	0.433

Table 2.2: Clustering accuracies (normalized mutual information/NMI and accuracy/AC) for spectral clustering algorithms based on ℓ^1-graph, Gaussian-kernel graph (G-graph), LE-graphs, and LLE-graphs, as well as PCA+K-means on the forest covertype database.

COV	Metric	ℓ^1-graph	G-graph	LE-graph		LLE-graph		PCA+
Cluster #				k-nn	ϵ-ball	k-nn	ϵ-ball	K-means
$K=3$	NMI	**0.792**	0.651(220)	0.554(16)	0.419(6)	0.642(20)	0.475(6)	0.555(5)
	AC	**0.903**	0.767	0.697	0.611	0.813	0.650	0.707
$K=4$	NMI	**0.706**	0.585(145)	0.533(13)	0.534(6)	0.622(20)	0.403(5)	0.522(13)
	AC	**0.813**	0.680	0.608	0.613	0.782	0.519	0.553
$K=5$	NMI	**0.623**	0.561(240)	0.515(12)	0.451(5)	0.556(10)	0.393(7)	0.454(15)
	AC	**0.662**	0.584	0.541	0.506	0.604	0.448	0.486
$K=6$	NMI	**0.664**	0.562(200)	0.545(6)	0.482(6)	0.602(20)	0.465(7)	0.528(8)
	AC	**0.693**	0.585	0.564	0.523	0.632	0.509	0.547
$K=7$	NMI	**0.763**	0.621(130)	0.593(9)	0.452(6)	0.603(11)	0.319(6)	0.602(17)
	AC	**0.795**	0.642	0.629	0.498	0.634	0.394	0.631

Table 2.3: Clustering accuracies (normalized mutual information/NMI and accuracy/AC) for spectral clustering algorithms based on ℓ^1-graph, Gaussian-kernel graph (G-graph), LE-graphs, and LLE-graphs, as well as PCA+K-means on the Extended YALE-B database. Note that the G-graph performs extremely bad in this case, a possible explanation of which is that the illumination difference dominates the clustering results in G-graph based spectral clustering algorithm.

YALE-B	Metric	ℓ^1-graph	G-graph	LE-graph		LLE-graph		PCA+
Cluster #				k-nn	ϵ-ball	k-nn	ϵ-ball	K-means
$K = 10$	NMI	**0.738**	0.07(220)	0.420(4)	0.354(16)	0.404(3)	0.302(3)	0.255(180)
	AC	**0.758**	0.175	0.453	0.413	0.450	0.383	0.302
$K = 15$	NMI	**0.759**	0.08(380)	0.494(4)	0.475(20)	0.438(5)	0.261(5)	0.205(110)
	AC	**0.762**	0.132	0.464	0.494	0.440	0.257	0.226
$K = 20$	NMI	**0.786**	0.08(290)	0.492(2)	0.450(18)	0.454(4)	0.269(3)	0.243(110)
	AC	**0.793**	0.113	0.478	0.445	0.418	0.241	0.238
$K = 30$	NMI	**0.803**	0.09(50)	0.507(2)	0.417(24)	0.459(7)	0.283(4)	0.194(170)
	AC	**0.821**	0.088	0.459	0.383	0.410	0.236	0.169
$K = 38$	NMI	**0.776**	0.11(50)	0.497(2)	0.485(21)	0.473(8)	0.319(4)	0.165(190)
	AC	**0.785**	0.081	0.443	0.445	0.408	0.248	0.138

that all the results listed in the tables are from the best tuning of all possible algorithmic parameters, *e.g.*, kernel parameter for G-graph, the number of neighboring samples and ϵ for LE-graphs and LLE-graphs, and the retained feature dimensions for PCA. To further compare the ℓ^1-norm and ℓ^2-norm in graph edge weight deduction, we show the clustering accuracies on USPS based on ℓ^1-graph and (k-nn + LLE)-graphs with variant k in Figure 2.4, which shows ℓ^1-graph is consistently better than ℓ^2-norm based graph construction for all k's, and the performance of the latter first increases, and then drops very slowly after k is large enough.

2.4.3 Subspace Learning with ℓ^1-graph

The experiments on classification based on subspace learning are also conducted on the above three databases. To make the comparison fair, for all the evaluated algorithms we

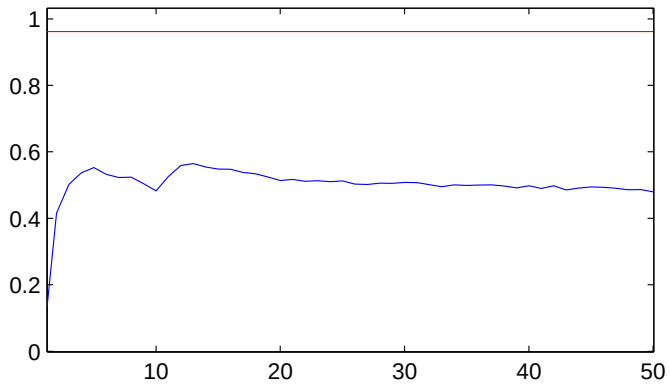


Figure 2.4: Comparison clustering accuracies of the ℓ^1-graph (red line, one fixed value) and (k-nn + LLE)-graphs (blue curve) with variant k's on the USPS dataset and K=7. It shows that ℓ^1-norm is superior over ℓ^2-norm in deducing informative graph weights.

first apply PCA as preprocessing step by retaining 98% energy.

To extensively evaluate the algorithmic performance on the USPS database, we randomly sampled 10, 20, 30 and 40 images from each digit as training data. Similarly, for the forest covertype database, we randomly sampled 5, 10, 15 and 20 samples from each class as training data, and for the Extended YALE-B database, we randomly sampled 10, 20, 30 , 40 and 50 training images for each individual. All the remaining data are used for testing purpose. Here we use the error rate to measure the classification performance, defined as

$$\text{error rate} = 1 - \frac{\sum_{i=1}^{N_t} \delta(\hat{y}_i, y_i)}{N_t} \tag{2.14}$$

where $\hat{y}_i$ is the predicted sample label and y_i is the given sample label, N_t is the total number of testing samples,and $\delta(\hat{y}_i, y_i)$ equals 0 if $\hat{y}_i \neq y_i$, otherwise equals 1. The best performance of each algorithm overall possible parameters, *i.e.*, graph parameters and feature dimension retained, is reported along with the corresponding feature dimension. The popular unsupervised subspace learning algorithms PCA, NPE and LPP, and the

Table 2.4: USPS digit recognition error rates (%) for different subspace learning algorithms. Note that the numbers in the parentheses are the feature dimensions retained with the best accuracies.

USPS	Unsupervised				Supervised
Train #	PCA	NPE	LPP	ℓ^1-graph-SL	Fisherfaces
10	37.21(17)	33.21(33)	30.54(19)	21.91(13)	**15.82**(9)
20	30.59(26)	27.97(22)	26.12(19)	18.11(13)	**13.60**(9)
30	26.67(29)	23.46(42)	23.19(26)	16.81(15)	**13.59**(7)
40	23.25(25)	20.86(18)	19.92(32)	14.35(19)	**12.29**(7)

Table 2.5: Forest cover recognition error rates (%) for different subspace learning algorithms.

COV	Unsupervised				Supervised
Train #	PCA	NPE	LPP	ℓ^1-graph-SL	Fisherfaces
5	33.23(17)	28.80(6)	35.09(12)	**23.36**(6)	23.81(6)
10	27.29(18)	25.56(11)	27.30(16)	**19.76**(15)	21.17(4)
15	23.75(14)	22.69(16)	23.26(34)	**17.85**(7)	19.57(6)
20	21.03(29)	20.10(10)	20.75(34)	**16.44**(6)	18.09(6)

Table 2.6: Face recognition error rates (%) for different subspace learning algorithms on the Extended YALE-B database.

YALE-B	Unsupervised				Supervised
Train #	PCA	NPE	LPP	ℓ^1-graph-SL	Fisherfaces
10	44.41(268)	23.41(419)	24.61(234)	**14.26**(112)	**13.92**(37)
20	27.17(263)	14.62(317)	14.76(281)	**5.30**(118)	9.46(37)
30	20.11(254)	9.40(485)	8.65(246)	**3.36**(254)	12.45(34)
40	16.98(200)	5.84(506)	5.30(263)	**1.93**(143)	3.79(37)
50	12.68(366)	3.78(488)	3.02(296)	**0.75**(275)	1.64(37)

supervised algorithm Fisherfaces [19] are evaluated for comparison with ℓ^1-graph based subspace learning, which is essentially unsupervised. For NPE and LPP, we used their unsupervised versions for fair comparison. For LPP, we use the *cosine* metric in graph construction for a better performance. The detailed comparison experimental results for classification are listed in Table 2.4-2.6 for these three databases, from which we can observe: 1) on the forest covertype and Extended YALE-B databases, ℓ^1-graph based unsupervised subspace learning algorithm generally performs better than the supervised

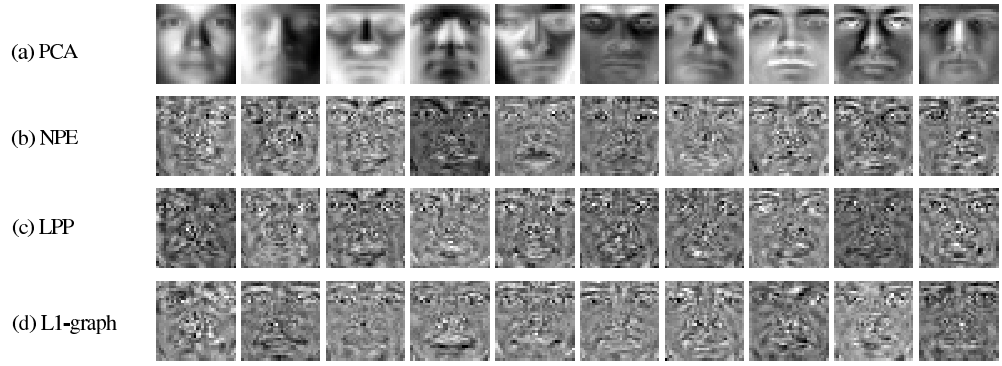


Figure 2.5: Visualization comparison of the subspace learning results. They are the first 10 basis vectors of (a) PCA, (b) NPE, (c) LPP, and (d) ℓ^1-graph calculated from the face images in YALE-B database.

algorithm Fisherfaces, and on the USPS database, Fisherfaces shows a little better than the former; 2) the ℓ^1-graph based subspace learning algorithm is much superior over all the other evaluated unsupervised subspace learning algorithms; and 3) NPE and LPP show to be better than PCA. Note that for all the classification experiments in this chapter, we used the classical nearest neighbor classifier [19][28][20] for fairly comparing the discriminating power of the derived subspaces from different subspace learning algorithms. The visualization comparison of the subspaces learnt based on ℓ^1-graph and those based onPCA, LPP and NPE are depicted in Figure 2.5, from which we can observe bases from PCA show to be most similar to real faces since PCA is motivated for direct data reconstruction.

2.4.4 Semi-supervised Learning with ℓ^1-graph

The semi-supervised learning is driven by the philosophy that the unlabeled data can also convey useful information for the learning process. We also use the above three databases for evaluating the effectiveness of the semi-supervised algorithm based on ℓ^1-graph by comparing with semi-supervised learning algorithms based on Gaussian-kernel

graph, LE-graph and LLE-graph. For all the semi-supervised learning algorithms, the supervised part is based on the Marginal Fisher Analysis [21] algorithm. And the error rate is also used to measure the performances. For a fair comparison, the parameters k_1, k_2, and γ are tuned for all proper combinations, and the result reported is based on the best parameter combination. The detailed comparison experiment results for semi-supervised leaning algorithms based on different graphs, the original supervised algorithm and the baseline of PCA, are shown in Table 2.7-2.9, from which we can have two observations: 1) the ℓ_1-graph based semi-supervised learning algorithm generally achieves the highest classification accuracy compared to semi-supervised learning based on those traditional graphs, and 2) semi-supervised learning can generally bring accuracy improvement compared to the counterparts without harnessing extra information from the unlabeled data.

Table 2.7: USPS digit recognition error rates (%) for different semi-supervised, supervised and unsupervised learning algorithms. Note that the numbers in the parentheses are the feature dimensions retained with the best accuracies.

USPS	Semi-supervised			Supervised	Unsupervised
Train #	ℓ^1-graph	LLE-graph	LE-graph	MFA [21]	PCA
10	**25.11**(33)	34.63(9)	30.74(33)	34.63(9)	37.21(17)
20	**26.94**(41)	41.38(39)	30.39(41)	41.38(39)	30.59(26)
30	**23.25**(49)	36.55(49)	27.50(49)	44.34(47)	26.67(29)
40	**19.17**(83)	30.28(83)	23.55(83)	35.95(83)	23.35(25)

2.5 Conclusion

In machine learning, the graph construction procedure essentially determines the potentials of those graph-oriented learning algorithms for image analysis. We address the graph construction problem as one of finding the sparse representation of each datum

Table 2.8: Forest cover recognition error rates (%) for different semi-supervised, supervised and unsupervised learning algorithms. Note that the numbers in the parentheses are the feature dimensions retained with the best accuracies.

COV	Semi-supervised			Supervised	Unsupervised
Train #	ℓ^1-graph	LLE-graph	LE-graph	MFA [21]	PCA
5	**22.50**(9)	29.89(5)	25.81(7)	29.89(5)	33.23(17)
10	**17.45**(10)	24.93(10)	22.74(8)	24.93(10)	27.29(18)
20	**15.00**(8)	19.17(10)	17.38(9)	19.17(10)	23.75(14)
30	**12.26**(8)	15.32(8)	13.81(10)	16.40(8)	21.03(29)

Table 2.9: Face recognition error rates (%) for different semi-supervised, supervised and unsupervised learning algorithms on the Extended YALE-B database. Note that the numbers in the parentheses are the feature dimensions retained with the best accuracies.

YALE-B	Semi-supervised			Supervised	Unsupervised
Train #	ℓ^1-graph	LLE-graph	LE-graph	MFA [21]	PCA
5	**21.63**(51)	33.47(51)	33.47(51)	33.47(51)	61.34(176)
10	**9.56**(61)	18.39(33)	18.39(33)	18.39(33)	44.41(268)
20	**5.05**(57)	14.30(29)	11.26(53)	14.30(29)	27.17(263)
30	**2.92**(73)	9.15(70)	7.37(71)	11.06(70)	20.11(254)

with respect to the dictionary composed of the remaining data samples. The sparse representation coefficients, which have been empirically shown to be informative for classification purposes, are used directly to determine the edge weights between the current datum and all the remaining data samples. Such a graph construction procedure is based on the assumption that natural highdimensional signals lie in a union of low-dimensional linear subspaces. A series of new algorithms for various machine learning tasks, e.g., data clustering, subspace learning, and semi-supervised learning, are then derived based on this new graph. Compared with with the conventional k-nearest-neighbor graph and the ϵ-ball graph, we demonstrate that our new graph possesses three advantages: (1) robustness to noise; (2) automatic sparsity; and (3) adaptive neighborhood selection. Extensive experiments on diverse real-world datasets show the consistent su-

periority of our new graph over those classical graphs in clustering, subspace learning, and semi-supervised learning tasks.

Chapter 3

Supervised Sparse Coding Towards Misalignment-Robust Face Recognition

3.1 Introduction

Face recognition has been motivated by both its scientific values and potential applications in the practice of computer vision and machine learning. This problem has been extensively studied and much progress has been achieved during the past decades. As a standard preprocessing step for face recognition, face alignment and cropping are generally applied in automatic face recognition systems, and face images are typically aligned according to the positions of corresponding eyes [32], [33], [34]. The main purpose of face alignment is to build the semantic correspondences between the pixels of different images and eventually to classify by matching the pixels with identical semantic meaning.

Unfortunately, the images may not be accurately aligned, and the pixels for the same

facial landmarks may not be strictly matched. Practical systems, or even manual face cropping, may bring considerable image misalignments, including translations, scaling, and rotation. These transformations can consequently make discrepant the semantics of two pixels in different images but at the same position. This discrepancy may inversely affect image similarity measurement, and consequently degrade face recognition performance. Thus it is a challenging problem to recognize faces under scenarios with spatial misalignments, where the margins between subjects tend to be more ambiguous.

In the literature, there exist some attempts to analyze and tackle this type of problems. Shan $et\ al.$ [35] showed that the effect of spatial misalignments can be alleviated to some extent by adding virtual gallery samples with artificial spatial misalignments. Yang $et\ al.$ [36] proposed a solution to improve algorithmic robustness to image misalignments with ubiquitously supervised subspace learning. Xu $et\ al.$ [37] proposed a solution based on the so-called Spatially constrained Earth Mover's Distance (SEMD), which is more robust against spatial misalignments than the traditional distance measures ($e.g.$, Euclidean distance). Recently, Wang $et\ al.$ [38] provided a novel and efficient algorithm for face recognition under scenarios with spatial misalignments by solving a constrained ℓ_1-norm optimization problem, which minimizes the error between the misalignment-amended image and the image reconstructed from the given subspace along with its principal complementary subspace. However, the spatial misalignment problem is still far from being solved, since: 1) most of these methods focus on the global features of face images, yet typically, the global features are much more sensitive to spatial misalignments compared with local features; and 2) the only patch-based method proposed in our previous work [37] towards misalignment-robust face recognition is however not robust to image occlusions.

In this chapter, we present a supervised sparse coding framework for face recogni-

tion under the scenarios with possible spatial misalignments for both gallery and probe images. As spatial misalignments often lead to large divergence among images from the same subject, the global features, $e.g.$, a vector concatenating gray-level values of all pixels, may lack of enough discriminating power for recognition purpose. Instead, if an image is considered as a set of orderless local patches, then this bag-of-patch representation shall be less sensitive to spatial misalignments compared with global features. In this work, each gallery image is partitioned into local patches at both original and misaligned positions as well as scales. To mitigate the affect of noise in extracting patches at misaligned positions and scales, we throw away those patches which may bring in noise near the image borders for the gallery images. The classification of a probe image is achieved with collective sparse codings of all the uniformly partitioned patches of the probe image from all the patches of all gallery images, and the solution is obtained via ℓ_1-norm optimization with the enforcement of sparsity on both patch level and subject level. More specifically, the patches from a probe image should be reconstructed from as few patches as possible, and also from as few subjects as possible. The final subject decision can be then be determined based on the reconstruction coefficient sums over different subjects.

The rest of this chapter is organized as follows. In Section 2, we first introduce the motivations of the supervised sparse patch coding framework and the details on ℓ_1-norm based sparse coding for general classification purpose. Then the details on supervised sparse patch coding framework for misalignment-robust face recognition are elaborated in Section 3. Section 4 demonstrates the experiment results. Finally, some concluding remarks are presented in Section 5.

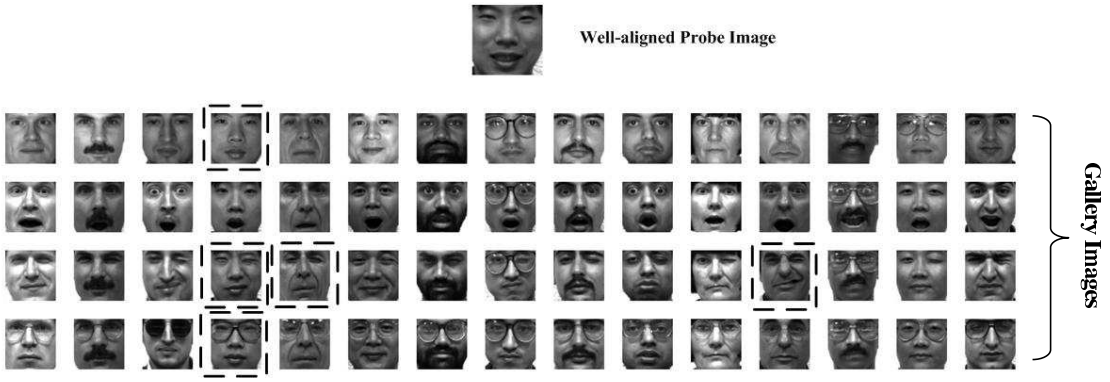
3.2 Motivations and Background

3.2.1 Motivations

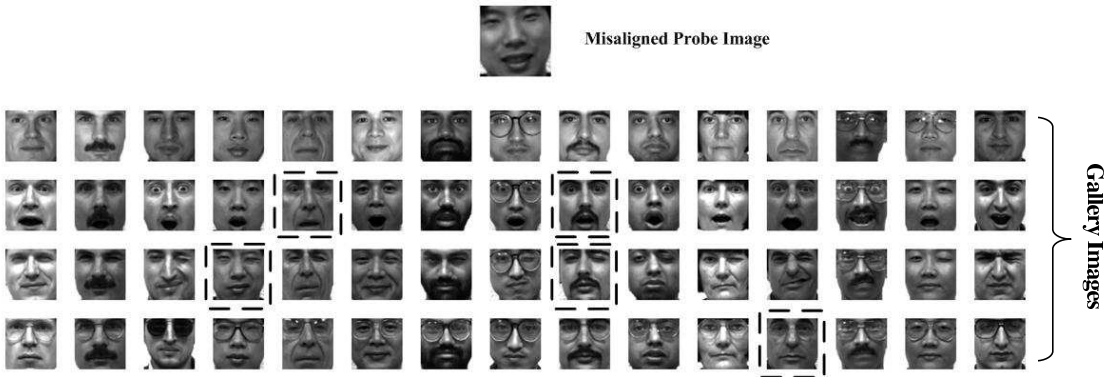
For face recognition task, the face images generally need first be aligned and cropped out from the original images, which may contain background objects, and one naive way is to fix the locations of two eyes in a fixed-size image rectangle. For practical systems, however, the positions of the two eyes may need be automatically located by face alignment algorithms [39] or eye detectors [40], so it is inevitable that there may exist localization errors, namely spatial misalignments. These spatial misalignments include four components, translations in horizontal and vertical directions (T_x, T_y), scaling (S), and rotation (θ). When spatial misalignment occurs, the usage of global features typically leads to substantially different data distribution compared with data without such spatial misalignments. Figure 3.1 shows such a demonstration, where the 5 nearest neighbors of a misaligned face image are considerably different from those of well-aligned face image, if measured based on Euclidean distance and with global features. This observation motivates us to utilize orderless local patch based image representation, which is generally more robust to spatial misalignments compared with global features.

Recently, Wright $et\ al.$ [5] exploited the classification potentials of sparse representation/coding in face recognition problem. In [5], each probe image is sparsely reconstructed from an over-complete dictionary, whose bases are the gallery samples and bases for noises, by solving a general ℓ_1-norm optimization problem. This solution is learning free, and robust to image occlusions, it is however intuitively sensitive to spatial misalignments.

Motivated by above observations, we propose a *supervised* sparse *patch* coding



(a) 5 nearest neighbors (marked with rectangles) of a probe face image without spatial misalignment based on Euclidean distance.



(b) 5 nearest neighbors (marked with rectangles) of a probe face image with misalignment based on Euclidean distance.

Figure 3.1: The neighboring samples comparison between the well-aligned and misaligned face images. It is observed that the neighboring samples may change substantially when the spatial misalignment occurs. The face images are from the ORL [2] dataset and each column includes the gallery images from one subject.

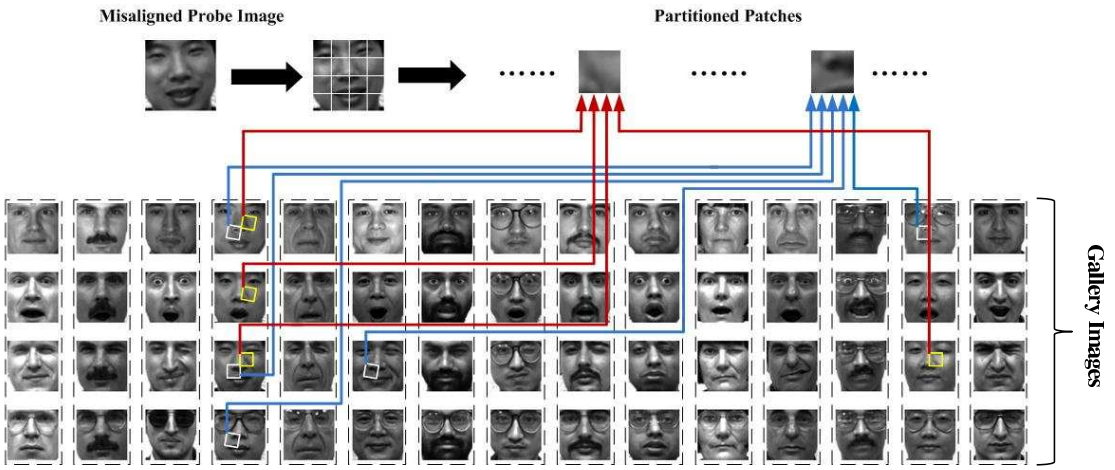


Figure 3.2: Collective patch reconstruction from SSPC. The first line is the misaligned probe image and its partitioned patches. These patches are sparsely reconstructed with gallery patches selected by SSPC, which are marked with rectangles in gallery images.

(SSPC) framework for enhancing the general sparse coding towards misalignment-robust face recognition. Th general idea of SSPC is to integrate the local-patch based image representation, supervised learning philosophy and sparse coding towards four algorithmic characteristics: 1) the solution is model free and no learning process is required, 2) the solution is robust to spatial misalignments, 3) the solution is robust to image occlusions, and 4) the solution is effective even when there exist spatial misalignments for gallery images. Figure 3.2 shows an exemplary result from SSPC, from which we can observe that the patches from the misaligned image in Figure 3.1(b) are mainly reconstructed from patches within the images from the identical subject.

3.2.2 Review on Sparse Coding for Classification

The research on sparse coding has a long history. Recent research shows that sparse coding appears to be biologically plausible as well as empirically effective for image processing and pattern classification [5]. In this subsection, we give a brief review on

sparse coding within the context of face recognition, which serves as the foundation for our proposed supervised sparse patch coding framework.

Here, the given n_k gallery images from the k-th subject are represented as a matrix,

$$X_k = [\mathbf{x}_{1,k}, \mathbf{x}_{2,k}, \cdots, \mathbf{x}_{n_k,k}] \in \mathbb{R}^{m \times n_k}, \tag{3.1}$$

where $\mathbf{x}_{i,k}$ means the i-th image of the k-th subject. The sample matrix X is then defined as the entire gallery set by concatenating the $n = \sum_{k=1}^{N_c} n_k$ gallery samples from N_c subjects,

$$X = [X_1, X_2, \cdots, X_{N_c}] = [\mathbf{x}_{1,1}, \mathbf{x}_{2,1}, \cdots, \mathbf{x}_{n_{N_c},N_c}]. \tag{3.2}$$

Denote y as the feature representation of a probe image. For face recognition, if there exists only illumination variation for images from the same subject, the images from this subject can then be represented with a low-dimensional subspace [41]. If sufficient gallery images are available for each subject in this case, it is possible to represent y as a linear combination of the column vectors of A_k, where k indicates the index of the subject the image y belongs to, namely,

$$y = X_k \alpha_k, \tag{3.3}$$

where $\alpha_k \in \mathbb{R}^{n_k}$ is the coefficient vector. However the subject index for image y is unknown, and thus we turn to reconstruct y as

$$y = X \alpha_0, \tag{3.4}$$

with the expectation that α_0 is sparse and the non-zero elements right correspond to the

subject k.

A natural formulation to seek the sparsest solution for $y = X\alpha_0$ is,

$$\alpha_0 = \arg\min_{\alpha} ||\alpha||_0, \quad \textit{s.t.} \quad X\alpha = y, \tag{3.5}$$

where $||\cdot||_0$ denotes the ℓ_0-norm, which counts the number of nonzero elements in a vector. However, the problem of finding the sparest solution of an under-determined system of linear equations is NP-hard, and difficult even to approximate. Actually, in general case, no known procedure to find the sparest solution is significantly more efficient than exhaustively evaluating all subsets of the entries for α.

Fortunately, recently development in theories on sparse representation reveals that if the solution α_0 is sparse enough, the solution from the ℓ_0-norm minimization can be recovered by the solution to the following ℓ_1-norm minimization problem,

$$\alpha_1 = \arg\min_{\alpha} ||\alpha||_1, \quad \textit{s.t.} \quad X\alpha = y. \tag{3.6}$$

This optimization problem is convex and can be transformed into a general linear programming problem. There exists a globally optimal solution, which can be solved efficiently using the classical ℓ_1-norm optimization toolboxes like [26].

Furthermore, real world images may be noisy, and thus it may be impossible to express y exactly as a sparse superposition of the column vectors of X. To explicitly account for those often sparse noises, the sparse coding formulation in Eq. (4.1) is rewritten as follows,

$$y = X\alpha + \epsilon, \tag{3.7}$$

where $\epsilon \in \mathbb{R}^m$ is a noise vector. The sparse solution can again be recovered by solving the following ℓ_1-norm minimization problem,

$$\min_{\alpha'} \|\alpha'\|_1, \;\; \textit{s.t.} \;\; y = X'\alpha', \tag{3.8}$$

where $X' = [X, \; I]$ and $\alpha' = [\alpha^T, \; \epsilon^T]^T$. It imposes the sparse constraints on both reconstruction coefficients and possible noises. Similarly, this problem can be solved by classical ℓ_1-norm optimization toolboxes.

3.3 Misalignment-Robust Face Recognition by Supervised Sparse Patch Coding

In this section, we introduce the details on supervised sparse patch coding framework for misalignment-robust face recognition. We follow the terminologies used in Section 2.

3.3.1 Patch Partition and Representation

The proposed framework starts with the image partition step. Here we use the gray-level values to describe the appearance of an image patch. Each gallery image $\mathbf{x}_{i,k}$ is uniformly partitioned into an ensemble of non-overlapping $w \times h$ patches, denoted as $\mathbf{x}_{i,k} = \{\mathbf{x}_{i,k}^j; j = 1, 2, \cdots, N_p\}$, where $\mathbf{x}_{i,k}^j \in \mathbf{R}^d$ ($d = w \times h$) is a d-dimensional feature vector and N_p is the number of patches belonging to one image. As aforementioned, for practical systems, there may exist spatial misalignments when cropping the face images out. The possible spatial misalignments are simplified using eight parameters in

this work: translations in both forward and backward horizontal directions (T_{fx}, T_{bx}), translations in both up and down vertical directions (T_{uy}, T_{dy}), scaling up and down (S_u, S_d), and left-hand and right-hand rotation (R_l, R_r). Then the virtual patches with these eight types of possible misalignments are obtained as an augmented gallery patch set, $\{\mathbf{x}_{i,k}^{j,p}; p = 0, T_{fx}, T_{bx}, T_{uy}, T_{dy}, S_u, S_d, R_l, R_r\}$. Note that when $p = 0$, $\mathbf{x}_{i,k}^{j,p}$ denotes the patch from gallery image without misalignments. To mitigate the affect of possible noises in virtual patches, we throw away the patches which may bring in noise near the image borders. For each $j = 1, 2, \cdots, N_p$, a patch set A^j is defined as follows,

$$A^j = [\mathbf{x}_{1,1}^{j,0}, \mathbf{x}_{1,1}^{j,T_{fx}}, \ldots, \mathbf{x}_{1,1}^{j,R_r}, \mathbf{x}_{2,1}^{j,0}, \mathbf{x}_{2,1}^{j,T_{fx}}, \ldots, \mathbf{x}_{n_{N_c},N_c}^{j,R_r}], \tag{3.9}$$

which includes all the related patches from the gallery set related to the j-th position in the image plane.

For a probe image y, we instead only partition it into uniform patches, and if we concatenate the representations of the patches into a long vector, it shall be right y if the elements of y are listed according to the order of the patches. Unlike general sparse coding in [5], which reconstruct y from the gallery images directly, we do the reconstructions for patches of y instead. Denote y^j as the feature vector for the j-th patch of the image y, then we assume that $y^j = A^j\alpha^j$, where α^j is the j-th sub-vector of the overall reconstruction vector α, namely the patch y^j is reconstructed from all the related patches of the gallery set related to the j-th position. The collective reconstructions for all the patches of the image y can then be represented as,

$$y = A\alpha, \tag{3.10}$$

where the matrix A is defined as,

$$A = \begin{bmatrix} A^1 & 0 & 0 & \dots & 0 \\ 0 & A^2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & \dots & A^{N_p} \end{bmatrix}.$$

3.3.2 Dual Sparsities for Collective Patch Reconstructions

The ultimate of sparse coding in this work is to propagate the subject information of the patches from the gallery images to the probe image y. Let $\beta_{i,k}^{j,p}$ denote the confidence weight of gallery patch $\mathbf{x}_{i,k}^{j,p}$. If the reconstruction coefficients $\alpha \Rightarrow \left\{\alpha_{i,k}^{j,p}\right\}$ for the probe image is obtained, we may have the following quantity γ_k to measure the overall confidence weight for each subject as,

$$\gamma_k = \sum_{i=1}^{n_k} \sum_{j=1}^{N_p} \sum_{p} \beta_{i,k}^{j,p} \alpha_{i,k}^{j,p}. \tag{3.11}$$

Then, we have a subject confidence vector γ=$[\gamma_1, \gamma_2, \cdots, \gamma_{N_c}]^T$. In our implementation, we set

$$\beta_{i,k}^{j,p} = \begin{cases} 1, & p = 0 \\ 1 - \epsilon, & \text{otherwise} \end{cases}$$

where $\epsilon = 0.02$ in this work. The underlying philosophy is that if the selected patch is a virtual patch, it shall convey less confidence to its associated subject compared with the original patches.

Intuitively, an optimal decision should come from a γ with one element as one and others as zeros, which motivates us to impose an extra sparse constraint on γ to achieve

more confident decision. Along with the sparse constraint on α^j as in general sparse coding [5], we then have a formulation for patch-based face recognition with dual sparsities.

To simplify Eq. (4.7), we define a set of matrices $B_{j,k}$ as,

$$B_{j,k} = \begin{matrix} [\beta_{1,k}^{j,0} & \beta_{1,k}^{j,T_{fx}} & \cdots & \beta_{1,k}^{j,R_r} & \beta_{2,k}^{j,0} & \beta_{2,k}^{j,T_{fx}} \\ & & \cdots & \beta_{2,k}^{j,R_r} & \cdots & \beta_{n_k,k}^{j,R_r}] \end{matrix},$$

and the matrix B_j is then defined as,

$$B_j = \begin{bmatrix} B_{j,1} & 0 & 0 & \dots & 0 \\ 0 & B_{j,2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & \dots & B_{j,N_c} \end{bmatrix}.$$

Let the matrix B be defined as,

$$B = [B_1, B_2, \cdots, B_{N_p}], \tag{3.12}$$

then we can rewrite Eq. (4.7) in a simple form as,

$$\gamma = B\alpha. \tag{3.13}$$

Based on above notations, we formally express the supervised sparse patch coding framework as the following optimization problem,

$$\begin{aligned} [\hat{\alpha}_1, \hat{\epsilon}_1, \hat{\gamma}_1] \quad &= \quad \arg\min_{\alpha,\epsilon,\gamma} ||\alpha||_1 + ||\epsilon||_1 + ||\gamma||_1, \\ &\textit{s.t.} \quad y = A\alpha + \epsilon,\ \gamma = B\alpha. \end{aligned} \tag{3.14}$$

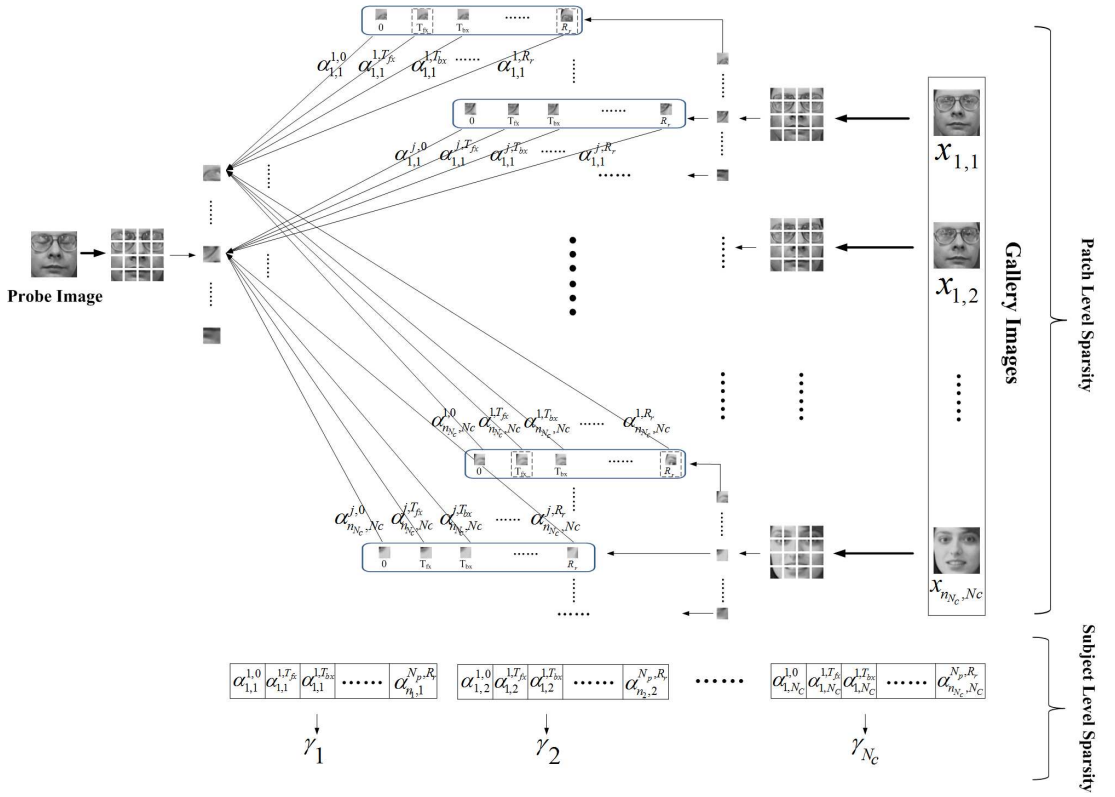


Figure 3.3: Exemplary illustration of the supervised sparse patch coding framework for uncovering how a face image can be robustly reconstructed from those gallery image patches. Note that the patches with broken lines shall be thrown away because they may bring in noises for those virtual patches.

Let

$$y' = \begin{bmatrix} y \\ 0 \end{bmatrix}, \alpha' = \begin{bmatrix} \alpha \\ \epsilon \\ \gamma \end{bmatrix}, A' = \begin{bmatrix} A & I & 0 \\ B & 0 & -I \end{bmatrix},$$

and then we can reformulate the supervised sparse patch coding framework as the ℓ_1-norm optimization problem below,

$$\hat{\alpha}'_1 = \arg\min_{\alpha'} ||\alpha'||_1, \quad \textit{s.t.} \quad y' = A'\alpha'. \tag{3.15}$$

This final formulation is right a general ℓ_1-norm minimization problem, and can thus be easily solved with ℓ_1-norm optimization toolboxes. It is predictable that the derived $\hat{\alpha}_1$ and $\hat{\gamma}_1$ shall be sparse if the system $y' = A'\alpha'$ is sufficiently under-determined. Figure 3.3 illustrates an exemplary explanation of the entire supervised sparse patch coding framework.

One interesting byproduct of this framework is its robustness against partial occlusion, although our main purpose is misalignment-robust face recognition. When partial occlusions occur in a gallery image, the strength of the occluded patches may be suppressed by that of the dominant *good* patches in collective patch reconstruction process, and the minimization of the $\|\epsilon\|_1$ shall naturally uncover the occluded area by the relatively large elements in the derived ϵ.

3.3.3 Related Work Discussions

Yang $et\ al.$ [36] proposed a solution to improve algorithmic robustness to image misalignments by ubiquitously supervised subspace learning. This method can deal with the cases where both probe and gallery images are misaligned. Our formulation in this work is different from [36] in several aspects: 1) the work [36] is based on global features, instead of local patch representations; 2) the work [36] cannot handle image occlusion issue; and 3) our proposed formulation is based on local patch representations, which are much less sensitive to spatial misalignments, and can be used under scenarios with both spatial misalignments and image occlusions.

Wang $et\ al.$ [38] provided a novel and efficient algorithm for face recognition under scenarios with spatial misalignments by solving a constrained ℓ_1-norm optimization problem, which minimizes the error between the misalignment-amended image and the image reconstructed from the given subspace along with its principal complementary subspace. This algorithm can deal with image occlusions, but it is still limited in the following aspects: 1) similar to [36], it is based on global features, instead of local patch representations; and 2) the work [38] cannot handle the cases where both probe and gallery images are misaligned, while our algorithm is workable under these scenarios.

Xu $et\ al.$ [37] proposed a solution based on the so-called Spatially constrained Earth Mover's Distance (SEMD), which is more robust against spatial misalignments than traditional distance measures ($e.g.$, Euclidean distance). This algorithm is patch-based as our proposed algorithm, however, it is sensitive to image occlusions, and thus not robust as our proposed algorithm.

As the main focus of this work is to strengthen traditional sparse coding algorithm for handling spatial misalignment issue, and the solutions in [36] [38] are limited for

subspace learning algorithms while [37] is sensitive to image occlusions, our experiments shall focus on the comparisons with traditional sparse coding algorithm and the intuitively reasonable and general solutions based on virtual samples [35].

3.4 Experiments

In this section, we systematically evaluate the superiority of our proposed supervised sparse patch coding (SSPC) framework over conventional sparse coding in term of robustness to spatial misalignments for face recognition task. Also the misalignment-robust counterparts with virtual misaligned samples for Principal Component Analysis (PCA) [18], Linear Discriminant Analysis (LDA) [19], Locality Preserving Projections (LPP) [20], and Neighborhood Preserving Embedding (NPE) [28] are evaluated to validate the effectiveness of our proposed SSPC framework.

3.4.1 Data Sets

Three popular face datasets, ORL [2], Yale [41], and Extended Yale-B [42], are used for performance evaluation. The ORL face database contains 10 different images of each of 40 distinct subjects. All the images were taken against a dark homogeneous background with the subjects in an upright and frontal position. The Yale face database contains 165 grayscale images of 15 individuals with 11images per subject, one per different facial expression or configuration: center-light, with/without glasses, happy, left-light, normal, right-light, sad, sleepy, surprised, and wink. The images are also manually cropped. The Extended YALE-B database contains 38 individuals and around 64 near frontal images under different illuminations per individual. All the images were taken against a dark

homogeneous background with the subjects in an upright and frontal position. Note that all the images in these three datasets are normalized to 28-by-28 pixels.

3.4.2 Experiment Setups

Face recognition experiments are conducted on above three benchmark face datasets under two scenarios with or without spatial misalignments. Three groups of experiments are designed under different misalignment setups for gallery and probe sets:

1) Face recognition on probe images with spatial misalignments, and gallery images without spatial alignments;

2) Face recognition on probe images with spatial misalignments, and gallery images also with spatial misalignments;

3) Face recognition on probe images with spatial misalignments and occlusions, and gallery images with spatial misalignments;

For each dataset, we conduct experiments with various configurations for gallery and probe sets for the sake of statistical importance, denoted as $'GaPb'$ for which a images of each subject are randomly selected for gallery set and the remaining b images of each subject are used for probe set. More specifically, for ORL dataset, we randomly select 2, 3, 4 images from each subject as gallery data. For the Yale database, we randomly select 3, 4, 5 images from each subject as gallery data. For YaleB database, we randomly select 10, 20, 30 gallery images for each subject. All the remaining data are used for probe set. Random artificial misalignments are added to the gallery and/or probe samples. As aforementioned, in our algorithm, to mitigate the affect of noises in extracting patches

at misaligned positions and scales, we throw away those patches near the image borders for the gallery images. At the same time, if the patches are too small, their representative capability shall be greatly degraded, and thus in the following experiments, we set the number of patches to be 4-by-4 for each probe image.

The unsupervised sparse coding based on global features is implemented for comparison. For a more comprehensive evaluation, SSPC is compared with those popular subspace learning algorithms including PCA, LDA, LPP and NPE, implemented in two versions with and without virtual samples. The nearest neighbor approach is used for final classification after dimensionality reduction for these algorithms. All possible dimensions of the final low-dimensional representation are evaluated and the best results are reported. Here we use the mixed spatial misalignments to simulate the misalignments brought by the automatic face alignment process. In the mixed spatial misalignment configuration, a rotation $R \in [-5^o, +5^o]$, a scaling $S \in [0.95, 1.05]$, a horizontal shift $T_x \in [-1, +1]$, and a vertical shift $T_y \in [-1, +1]$ are randomly added to the images, which are then assumed to be spatially misaligned.

3.4.3 Experiment Results

Only Probe Images are Misaligned

In these experiments, we assume that gallery images are well aligned while the probe images are spatially misaligned. To better understand the effect of virtual samples [35], the experiment results from the original gallery set and the gallery set containing virtual samples are both reported, denoted as "o/w" in the result tables. The detailed comparison experiment results are listed in Table 3.1-3.3 for these three datasets, from which

we can have the following observations: 1) the recognition results from SSPC framework are consistently much better than those from all the competing algorithms; 2) the results from supervised sparse coding are a little better than those from unsupervised sparse coding; 3) the results from the original gallery set show to be generally worse than the corresponding results from the gallery set with virtual samples, and thus for the consequent experiments, we only report the results from the gallery set with virtual samples; and 4) on the ORL and Yale datasets, the performances of unsupervised sparse coding are not very good because the numbers of gallery samples are very small for each subject, while on the YaleB dataset, the performances of unsupervised sparse coding improve greatly, and are generally better than those of LDA. This shows that the conventional sparse coding algorithm is good under scenarios with large-scale dataset.

Table 3.1: Face recognition error rates (%) for different algorithms on ORL dataset. Here only probe images are spatially misaligned.

ORL #	PCA(o/w)	LPP(o/w)	NPE(o/w)	LDA(o/w)	Unsupervised Sparse Coding(o/w)	Supervised Sparse Coding(o/w)	SSPC N_p=4*4
G2P8	54.69/32.46	63.37/29.72	42.26/27.19	33.28/16.53	34.27/20.03	33.53/19.61	**12.95**
G3P7	36.90/21.83	56.98/19.36	35.87/18.86	19.34/8.93	24.84/13.84	24.31/13.41	**6.27**
G4P6	31.76/15.97	50.92/12.22	28.43/14.12	16.87/5.93	19.72/8.33	19.43/8.01	**3.93**

Table 3.2: Face recognition error rates (%) for different algorithms on Yale dataset. Here only probe images are spatially misaligned.

Yale #	PCA(o/w)	LPP(o/w)	NPE(o/w)	LDA(o/w)	Unsupervised Sparse Coding(o/w)	Supervised Sparse Coding(o/w)	SSPC N_p=4*4
G3P8	52.50/39.07	60.18/31.84	51.57/37.96	38.81/29.72	45.00/29.98	44.56/29.44	**18.61**
G4P7	50.16/35.87	56.08/24.98	48.36/29.31	34.18/22.96	40.11/25.33	38.68/24.97	**13.01**
G5P6	52.35/34.56	54.25/22.14	47.64/28.15	30.86/19.26	38.51/22.06	37.12/21.48	**9.62**

Table 3.3: Face recognition error rates (%) for different algorithms on YaleB dataset. Here only probe images are spatially misaligned.

YaleB #	PCA(o/w)	LPP(o/w)	NPE(o/w)	LDA(o/w)	Unsupervised Sparse Coding(o/w)	Supervised Sparse Coding(o/w)	SSPC N_p=4*4
G10P40	39.84/28.60	38.82/17.62	37.23/17.53	33.99/14.83	30.97/22.81	30.11/22.66	**6.30**
G20P30	37.06/21.36	32.51/14.61	32.28/13.65	30.22/7.32	17.86/11.39	17.03/11.31	**4.25**
G30P20	31.02/16.02	30.06/11.64	29.36/11.45	29.04/6.02	16.17/9.07	15.91/8.95	**1.86**

Both Gallery and Probe Images are Misaligned

In these experiments, we further consider the scenario where spatial misalignments exist in both gallery and probe sets. We simulate this scenario by adding random artificial misalignments to all the gallery and probe images. The gallery/probe split setups are the same as those for the former experiments. The detailed comparison experiment results are listed in Table 3.4-3.6 for these three datasets, from which we can observe that our SSPC again significantly outperforms all the other competing algorithms, when gallery and probe images are both contaminated by spatial misalignments.

Table 3.4: Face recognition error rates (%) for different algorithms on ORL dataset. Here both gallery and probe images are misaligned.

Yale #	PCA	LPP	NPE	LDA	Unsupervised Sparse Coding	Supervised Sparse Coding	SSPC N_p=4*4
G2P8	39.24	35.07	34.72	23.61	27.01	26.66	**20.23**
G3P7	27.63	23.65	23.01	12.70	17.79	17.07	**9.83**
G4P6	21.71	17.75	17.13	8.59	12.96	12.50	**6.06**

Table 3.5: Face recognition error rates (%) for different algorithms on YALE dataset. Here both gallery and probe images are misaligned.

Yale #	PCA	LPP	NPE	LDA	Unsupervised Sparse Coding	Supervised Sparse Coding	SSPC N_p=4*4
G3P8	43.33	32.96	39.91	31.94	31.23	30.91	**21.50**
G4P7	37.88	26.14	32.59	23.39	27.84	27.41	**15.45**
G5P6	34.20	24.07	29.26	22.22	26.77	26.42	**11.73**

Table 3.6: Face recognition error rates (%) for different algorithms on YaleB dataset. Here both gallery and probe images are misaligned.

YaleB #	PCA	LPP	NPE	LDA	Unsupervised Sparse Coding	Supervised Sparse Coding	SSPC N_p=4*4
G10P40	37.35	31.54	29.41	17.13	24.82	24.53	**7.89**
G20P30	29.41	21.85	21.35	8.12	12.59	12.45	**6.82**
G30P20	25.44	19.39	17.97	7.18	11.14	10.87	**4.21**

Both Gallery and Probe Images are Misaligned, and Probe Images are with Occlusions

In these experiments, we show that the proposed SSPC is robust to partial occlusions as well as misalignments. The gallery images are misaligned while the probe images are not only misaligned, but also occluded. Here, an 8-by-8 artificial occlusion area is generated at a random position for each probe image. Figure 3.4 shows the exemplary face images with partial occlusions. The detailed comparison experiment results are listed in Table 3.7-3.9 for these three datasets. From the listed results, we can observe that the recognition results from our algorithm are slightly worse than those from gallery images without image occlusions, while the other algorithms all greatly suffer from the affection of image occlusions. Another observation is that sparse coding related algorithms are generally better than subspace related algorithms in this scenario, which validates the capability of general sparse coding in handling image occlusions.

Figure 3.4: Exemplary face images with partial image occlusions. Original image are displayed in the first row. An 8-by-8 occlusion area is randomly generated as shown in the second row, and the bottom row shows the occluded face images.

Table 3.7: Face recognition error rates (%) for different algorithms on ORL dataset. Here the probe images suffer from both misalignments and occlusions, and the gallery images are misaligned.

Yale #	PCA	LPP	NPE	LDA	Unsupervised Sparse Coding	Supervised Sparse Coding	SSPC N_p=4*4
G2P8	62.43	65.29	59.69	63.30	62.13	61.46	**24.17**
G3P7	57.02	61.86	58.10	63.10	58.06	57.73	**13.57**
G4P6	54.54	59.94	55.93	58.16	54.56	54.17	**9.26**

Table 3.8: Face recognition error rates (%) for different algorithms on YALE dataset. Here the probe images suffer from both misalignments and occlusions, and the gallery images are misaligned.

Yale #	PCA	LPP	NPE	LDA	Unsupervised Sparse Coding	Supervised Sparse Coding	SSPC N_p=4*4
G3P8	48.98	47.96	50.37	43.70	39.45	39.26	**23.24**
G4P7	45.71	44.65	44.44	44.66	36.07	35.66	**22.54**
G5P6	47.40	43.82	42.09	43.21	35.69	35.16	**17.90**

Table 3.9: Face recognition error rates (%) for different algorithms on YaleB dataset. Here the probe images suffer from both misalignments and occlusions, and the gallery images are misaligned.

YaleB #	PCA	LPP	NPE	LDA	Unsupervised Sparse Coding	Supervised Sparse Coding	SSPC N_p=4*4
G10P40	57.52	57.23	56.05	55.26	45.18	44.89	**19.36**
G20P30	46.83	46.55	44.18	37.12	30.43	30.21	**8.38**
G30P20	45.32	39.08	38.95	33.77	25.86	25.58	**5.16**

3.5 Conclusion

In this chapter, we developed the SSPC, supervised sparse patch coding, framework towards a robust solution to the challenging face recognition task with considerable spatial misalignments and possible image occlusions. In this framework, each image is represented as a set of local patches, and the classification of a probe image is achieved with the collective sparse reconstructions of the patches of the probe image from the patches of all the gallery images with the consideration of both spatial misalignments and the extra sparse enforcement on subject confidences. SSPC naturally integrates the patch-based representation, supervised learning and sparse coding, and thus is superior to most conventional algorithms in term of algorithmic robustness.

Chapter 4

Label to Region by Bi-Layer Sparsity Priors

4.1 Introduction

Keywords based queries have been found to be the most efficient and effective for Internet image search. Beyond simply harnessing the indirect surrounding texts of web images for query matching, the more desirable technique is to annotate the images with their associated semantic concepts/labels. To achieve reliable and visible content-based image retrieval, it is critical to obtain the correspondence between the image labels and their precise regions within an image. In practice, it is very tedious to manually annotate the image labels to the corresponding image regions, and a more feasible alternative is to annotate the labels at the image-level. Therefore, it is interesting and practically valuable to investigate how to automatically reassign the labels annotated at the image-level to those contextually derived image regions, ie, the label to region assignment (LRA) problem.

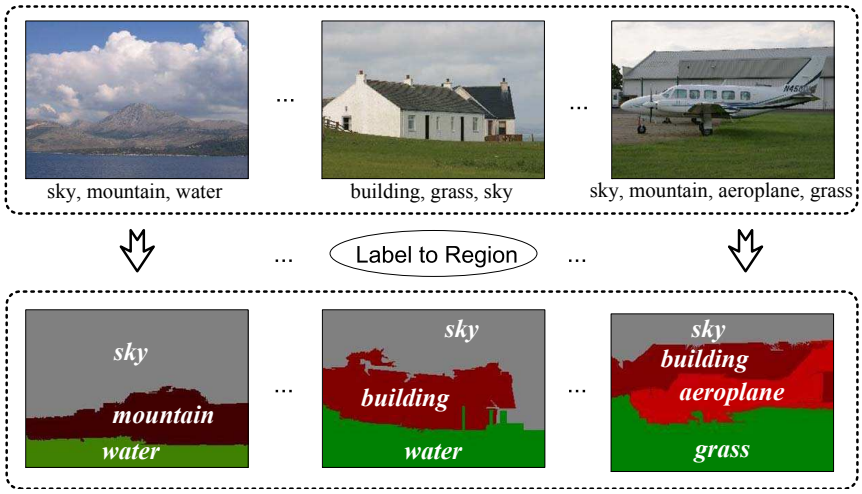


Figure 4.1: Exemplar illustration of the label-to-region assignment task. Note that: 1) no data with ground-truth label-to-region relations are provided as priors for this task, and 2) the inputs include only the image-level labels, with no semantic regions provided.

Although the LRA problem has not been essentially studied before, there are some related works in computer vision community, known as simultaneous object recognition and image segmentation. These algorithms can be roughly divided into two categories. The first category focuses on unsupervised learning techniques [43, 44, 45, 46, 47, 48]. Leibe *et al.* [47] propose to perform object localization, namely image segmentation along with object classification, by using an implicit shape model, which was further extended by Chen *et al.* in [48] to learn explicit shape model from single image. Both of them focused on single object category or assumed there is no overlapping between multiple objects in the training images. Winn *et al.* proposed in [45] to learn object classes based on the results of automatic image segmentation. A recent extension is presented by Cao *et al.* in [46], which applied the spatially coherent latent topic model to conduct multi-label image segmentation and classification. However these algorithms can only handle images either with single major object or with clean background and without occlusions between objects. In contrast, in this chapter we aim to process more challenging images containing multiple objects and with possible inter-object occlusions.

The second category is generally founded on supervised learning techniques. The typical efforts are the classifier-based methods [49, 50, 51, 52, 53], which usually first learn image classifiers to characterize concepts (or keywords) based on the training images, and then identify the images belonging to the specific category. These algorithms are very limited when encountering cases with semantically overlapped labels or imbalanced data from different semantic labels, which will heavily impair the discriminative power of these algorithms. There are also approaches which focus on learning the correlation between the visual features and semantic concepts, including CMRM [54] and its extended versions [55, 56, 57]. In addition, some works [58, 59, 60] are proposed to additionally harness the label correlation for label ranking and choosing the proper keywords as semantic annotations, and most of them use the image-to-image visual similarities to predict the image labels. However, there are usually multiple semantic concepts within one image and two different images containing a common object may contain different other objects at the same time. For example, in Figure 4.1, the image with objects "cow", "sky" and "mountain" may be visually different from the images with only "sky" or " mountain". Therefore, it is not reliable to directly compare the features of two images that may contain different number of objects from different categories.

Compared with the above efforts for simultaneous image annotation and parsing, LRA instead elicits a more challenging problem, characterized by: 1) the optimal partition of the input images to semantic regions and the correspondence between the annotated labels and image regions are unknown, which makes most state-of-the-art classifier based methods [49, 50, 51, 52, 53] inapplicable; and 2) all the spatially connected objects within an image need be assigned with individual labels, which may challenge those conventional unsupervised learning algorithms as aforementioned. Figure 4.1 illustrates the problem inputs, ie, images annotated with labels at the image-level, and the problem outputs, ie, semantic regions with labels, for the label-to-region assignment task.

To address the LRA problem, we propose to propagate the labels annotated at the image-level to those local semantic regions merged from the over-segmented atomic image patches of the entire image set. Generally, one label of an image only characterizes a single local semantic region, and two images with common labels often share similar semantic regions. Inversely, if two local semantic regions from different images are visually similar, these two images are likely to share certain common label. Thus, if the region-to-region correspondences have been given for all the image pairs, we can assign the common image labels to those corresponded local regions, which then translates the label-to-region assignment problem into a problem to uncover the region-level correspondence. In practice, these semantic regions corresponding to certain labels cannot be directly obtained, but those smaller-size spatially coherent image patches are easy to derive with classical image over-segmentation approach. One semantic region generally comprises of multiple such atomic patches, but it is infeasible to uncover the patch-to-region relations by merging those visually similar patches, due to the underlying large within-region variations. In this work, we instead propose to first construct the so-called candidate regions, initially grouping from those local spatially coherent patches, and then use those atomic patches from other input images to reconstruct the candidate regions, with the hypothesis that those selected atomic patches for reconstruction shall come from few semantically similar regions. Finally the cross-image patch-to-region correspondences are used for the ultimate label-to-region assignment purpose. Note that: (1) we cannot directly use visual similarity between the candidate region and atomic patch to select patches for the reconstruction purpose, since an atomic patch is only part of a region and their similarity cannot convey the inclusion relations; and (2) an intuitive way to improve the accuracy of cross-image region-to-region correspondence is to enforce the usage of atomic patches from few images for this reconstruction, from which those selected atomic patches from one image may have high possibility to form

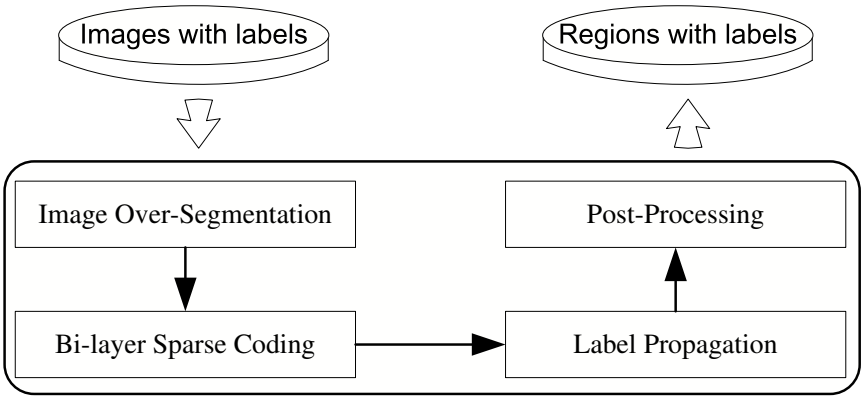


Figure 4.2: Sketch of our proposed solution to automatic label-to-region assignment task. This solution contains four steps: 1) patch extraction with image over-segmentation algorithm; 2) image reconstruction via bi-layer sparse coding, 3) label propagation between candidate region and selected image patches based on the coefficients from bi-layer sparse coding, and 4) post-processing for deriving both semantic regions and associated labels.

a semantic region.

More specifically, the reconstruction of a semantic region from a set of image patches is achieved by the proposed bi-layer sparse coding formulation. The basic philosophy is that an image or semantic region can be sparsely reconstructed via the image patches belonging to the images with common image labels. We additionally introduce another type of constraints, namely, to select patches from as few images as possible, which brings the second layer of sparsity to improve the fidelity in label-to-region assignment. Based on the sparse reconstruction coefficients, we assign the common image labels to the selected patches, and then further fuse all the assignment results to distribute the image labels to those contextually derived semantic regions merged from multiple atomic patches. The proposed label-to-region assignment process has the following characteristics: 1) the bi-layer sparse coding aims to enforce the usage of merged patches within an image to reconstruct the reference image or semantic region, which ensures the reliability of label propagation; 2) the process does not require exact image object/concept parsing, which is still far from satisfactory on real world images; and 3) no generative

model for each label/concept is learnt, and thus it is scalable to applications with large label set. In addition, the proposed bi-layer sparse coding formulation can also be directly applied on new test image to perform multi-label image annotation. Figure 4.2 illustrates the overall sketch of this idea.

The remainder of this chapter is organized as follows. We first formulate the label-to-region task within the bi-layer sparse coding framework in Section 4.2 and introduces how to use the bi-layer sparse coding for direct image annotation in Section 4.3. The detailed comparison experiments are then demonstrated in Section 4.4. Section 4.5 presents the conclusive remarks along with discussion for future work.

4.2 Label to Region Assignment by Bi-layer Sparsity Priors

4.2.1 Overview of Problem and Solution

The ability to annotate images with related text labels at the semantic region-level is valuable for boosting keyword based image search with the awareness of semantic image content. However it is tedious if not impossible to manually annotate labels at the region-level for large-scale image set. We therefore study in this work on how to utilize the cross-image label contexts to automatically reassign the image labels to those contextually merged image patches in a group manner. As illustrated in Figure 4.4, the image $\mathbf{y}$ comprises an ensemble of image patches, each of which may partially characterize one image label, *e.g.*, tree, building, etc. Two images annotated with common labels are likely to contain some similar patches. However it is generally difficult to directly

Figure 4.3: Exemplar image with over-segmentation result, where different colors indicate different patches.

derive those semantically similar patch pairs between two images. Thus instead we use a group of atomic patches to reconstruct an image or semantic region, and then harness the reconstruction coefficients for propagating the image labels to those localized image patches. Meanwhile, to reduce the influence of image noises and robustly derive label-to-region relations, we propose to enforce that the selected atomic patches should come from as few images as possible for the reconstruction purpose. Consequently, we obtain a bi-layer sparse coding framework, where each image is reconstructed using a few localized atomic patches from a few related images. Note that in this work, we only consider such localized labels, ie, the so-called *flat* labels.

4.2.2 Over-Segmentation and Representation

As aforementioned, the main purpose of this work is to propagate the semantic labels annotated at the image-level to image regions merged from image patches. Each homogeneous patch comprises of the pixels that are spatially coherent and perceptually similar with respect to certain appearance features, such as intensity, color and texture, etc.

Our proposed solution starts with an initial image over-segmented by a reliable segmentation algorithm into multiple homogeneous atomic patches. Here we choose to use

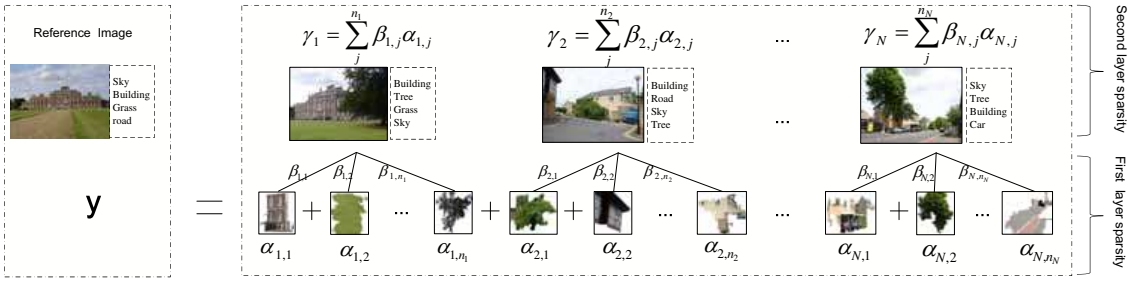


Figure 4.4: Illustration of bi-layer sparse coding formulation for uncovering how an image can be contextually and robustly reconstructed from those over-segmented atomic image patches.

the graph-based segmentation algorithm in [61], which incrementally merges smaller-size patches with similar appearances and with small minimum spanning tree weights. This method is of nearly linear computational complexity in the number of neighboring pixels. In this work we use a modified version of the algorithm in [61] to obtain coherent patches with homogeneous appearances. Note that our proposed solution is general and not tied to any specific image segmentation algorithm. We use the color features to describe the appearance of an image patch and partition an image into roughly homogeneous patches as in [61]. To ensure that each atomic patch contains only one single image label, we propose to stop merge if the patch size is larger than a predefined threshold. In this work, we first resize all the images into the resolution of 320×240 pixels and set 600 pixels as the threshold to stop further merging. Thus, for each image, we generally obtain about $40 \sim 50$ atomic patches. Based on Intel Xeon X5450 workstation with 3.0GHz CPU and 16GB memory, it takes less than 0.2 second to segment one image. Figure 4.3 shows an exemplary result of an over-segmented image.

The goal of the image over-segmentation step is to enforce that the segmented patch is involved within an object/concept, and these over-segmented patches shall be merged to constitute semantic regions. This way of using the image patches makes our algorithm less vulnerable to the quality of the image segmentation step.

Based on the image over-segmentation results, we can obtain the feature representations for those atomic patches. Let $\mathbf{X} = \{\mathbf{x}_i, z_i; i{=}1, \cdots, N\}$ denote the annotated image set, where N is the total image number, $z_i \in \mathbb{R}^{N_c}$ indicates the label vector and the binary $z_{i,c}$ takes 1 if the ith image contains the cth label and 0 otherwise. N_c is the total number of image labels. As aforementioned, each image $\mathbf{x}_i$ contains an ensemble of atomic patches, denoted as $X_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,n_i}]$, where $x_{i,n_i} \in \mathbb{R}^m$ is an m-dimensional feature descriptor and n_i is the number of patches belonging to the ith image. We then arrange all the patch representations as column vectors of the matrix $A = [X_1, X_2, \cdots, X_N] \in \mathbb{R}^{m \times \sum_{i=1}^{N} n_i}$. Then, for any specific image $Y = [y_1, \cdots, y_{n_y}] \in \mathbb{R}^{m \times n_y}$, the target of sparse image coding is to represent the full or partial sum of the column vectors as the linear combination of the column vectors in matrix A. In other words, an image or its initially merged candidate region is reconstructed from a set of over-segmented image patches.

We describe each atomic patch by using Bag-of-Words (BOW) features. The generation of visual words comprises of three steps: i) we apply the Difference-of-Gaussian filter on the gray-scale image to detect a set of salient points; ii) we then compute the Scale-Invariant-Feature-Transform (SIFT) [62] features over the local areas defined by the detected salient points; and iii) we perform the vector quantization on SIFT region descriptors to construct the visual vocabulary by K-Means clustering approach. In this work we generate 500 clusters, and thus the dimension of the BOW feature vector is $m = 500$.

4.2.3 I: Sparse Coding for Candidate Region

The core component of the solution to label-to-region assignment task is to found out the semantically-similar region-pair from two images that contains common labels/concepts.

We can then derive the label information for the region-pair from the shared images labels. A dilemma is that the over-segmentation step only produces smaller-size patches instead of semantic regions. In this work, we propose a sparse coding framework to implicitly uncover the semantic region correspondence by explicitly uncovering how an image or its candidate region can be reconstructed from the over-segmented patches of other input images. Mathematically, denote y as the feature representation of an image or its candidate region merged from over-segmented patches. If sufficient training samples are available for each label, it is possible to represent y as a sparse and linear combination of the patch representations from other input images, namely,

$$y = A\,\alpha_0 \in \mathbb{R}^m, \tag{4.1}$$

where α_0 is the coefficient vector whose entries are expected to be zeros except for those samples associated with common label(s) with y. For ease of representation, we use A again here for all the patch representations from all other input images.

Theoretically, α_0 can be obtained by solving the linear system of equations $y = A\alpha$, but when m ¡ $\sum_{i=1}^{N} n_i$, there exist infinite number of possible solutions. A possible way to select a solution is to minimize the ℓ^2-norm of the solution, namely,

$$\hat{\alpha}_2 = \arg\min_{\alpha} ||\alpha||_2, \quad s.t. \quad A\,\alpha = y. \tag{4.2}$$

The solution $\hat{\alpha}_2$, although is easy to obtain, is dense and thus not informative for reconstructing y. Essentially, the sparser the recovered α_0 is, the easier it will be to accurately determine the correspondence of y and the selected patches. Thus, it is reasonable to

seek the sparsest solution to $y = A\alpha$ by solving the following optimization problem:

$$\hat{\alpha}_0 = \arg\min_{\alpha} ||\alpha||_0, \quad \textit{s.t.} \quad A\,\alpha = y, \tag{4.3}$$

where $||\cdot||_0$ denotes the ℓ^0 norm, which counts the number of nonzero elements in a vector. However, this problem is NP-hard. Fortunately, recently development in theories on sparse representation reveals that if the solution $\hat{\alpha}_0$ is sparse enough, the solution from the ℓ^0-norm minimization can be recovered by the solution to the following ℓ^1-norm minimization problem:

$$\hat{\alpha}_1 = \arg\min_{\alpha} ||\alpha||_1, \quad \textit{s.t.} \quad A\,\alpha = y. \tag{4.4}$$

This optimization problem is convex and can be transformed into a general linear programming problem. There exists a globally optimal solution, which can be solved efficiently using the classical ℓ^1-norm optimization toolboxes, like [**?**].

Furthermore, the real world images are often noisy, and thus it may be impossible to express y exactly as a sparse superposition of the column vectors of A. To explicitly account for those often sparse noises, we rewrite the sparse coding formulation in Eq. (4.1) as follows:

$$y = A\alpha + \epsilon, \tag{4.5}$$

where $\epsilon \in \mathbb{R}^m$ is a noise vector. The sparse solution can again be recovered by solving the following robust ℓ^1-norm minimization problem:

$$[\hat{\alpha}_1, \hat{\epsilon}_1] = \arg\min_{\alpha,\epsilon} ||\alpha||_1 + ||\epsilon||_1, \quad \textit{s.t.} \quad y = A\,\alpha + \epsilon, \tag{4.6}$$

which simultaneously imposes the sparse constraints on both reconstruction coefficients and noises. Similarly, this problem can also be solved by classical ℓ_1-norm optimization toolboxes.

4.2.4 II: Sparsity for Patch-to-Region

The ultimate of sparse coding in this work is to build pairwise semantic region correspondence, which is then used for label propagation from image-level to region-level. Generally each semantic region comprises of serval over-segmented patches, and thus it is natural to enforce the possibility of merging atomic patches into semantic regions within individual image, which motivates an extra layer of sparsity, called the sparsity for patch-to-region.

Let $\beta_{i,j}$ denote the normalized importance weight of the jth patch for the ith input image, and we bring another set of coefficients, γ=$[\gamma_1, \gamma_2, \cdots, \gamma_N]^T$, to measure the total importance weights for individual image,

$$\gamma_i = \sum_{j=1}^{n_i} \beta_{i,j}\alpha_{i,j}, \tag{4.7}$$

where $\beta_{i,j}$ is calculated according to the size of the jth atomic patch and normalized by the image size of the ith image, and the index for α is rearranged according to the patch index within each image. Here, we define a matrix $B \in R^{N\times\sum_i n_i}$ using $\beta_{i,j}$ as:

$$B = \begin{bmatrix} \beta_{1,1} & \dots & \beta_{1,n_1} & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & \dots & \beta_{N,1} & \dots & \beta_{N,n_N} \end{bmatrix},$$

and then we can rewrite Eq. (4.7) as

$$\gamma = B\alpha. \tag{4.8}$$

Finally, we obtain the following optimization problem:

$$\begin{aligned} [\hat{\alpha}_1, \hat{\epsilon}_1, \hat{\gamma}_1] \quad &= \quad \arg\min_{\alpha,\epsilon,\gamma} ||\alpha||_1 + ||\epsilon||_1 + ||\gamma||_1, \\ &\textit{s.t.} \quad y = A\alpha + \epsilon,\ \gamma = B\alpha. \end{aligned} \tag{4.9}$$

Let

$$y' = \begin{bmatrix} y \\ 0_{N\times 1} \end{bmatrix}, \alpha' = \begin{bmatrix} \alpha \\ \epsilon \\ \gamma \end{bmatrix}, A' = \begin{bmatrix} A, I_{m\times m}, 0_{m\times N} \\ B, 0_{N\times m}, -I_{N\times N} \end{bmatrix},$$

and then we can reformulate the bi-layer sparse coding as the ℓ^1-norm optimization below,

$$\hat{\alpha}'_1 = \arg\min_{\alpha'} ||\alpha'||_1, \quad \textit{s.t.} \quad y' = A'\alpha'. \tag{4.10}$$

The derived $\hat{\alpha}_1$ and $\hat{\gamma}_1$ are both sparse, and thus y is reconstructed from a set of sparsely selected column vectors in A, which belong to few images. This result is in accordance with the real observations and the entire algorithm is called "Bi-layer Sparse Coding". Figure 4.4 illustrates the exemplary explanation of the bi-layer sparse coding formulation. Figure 4.5 shows the comparison results on the distribution of the reconstruction coefficient from bi-layer and one-layer sparse codings. We can observe that

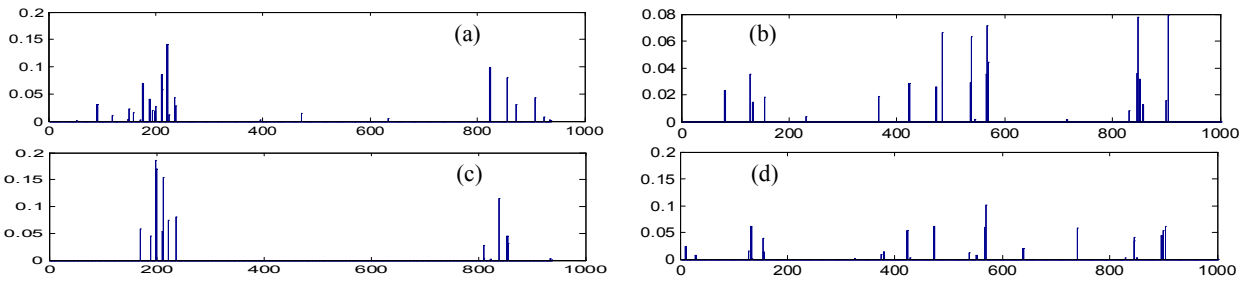


Figure 4.5: Two exemplar comparison results for bi-layer sparsity (a, c) vs. one-layer sparsity (b, d). The subfigures are obtained based on 20 samples randomly selected from the MSRC dataset used in the experiment part. The horizontal axis indicates the index for the atomic image patch and the vertical axis shows the values of the corresponding reconstruction coefficients (We only plot the positive ones for ease of display).

based on the bi-layer sparse coding, the selected patches tend to gather within a few images. Figure 4.6 displays some examples on how a candidate region within an image is reconstructed from the over-segmented atomic patches guided by bi-layer sparsity priors.

4.2.5 Contextual Label-to-Region Assignment

In this subsection, we further introduce how to utilize the bi-layer sparse coding for label-to-region assignment, namely, the simultaneous semantic region merging from atomic patches and region label assignment. The proposed procedure is motivated by the observation that, if the image or candidate region representation y from image $\mathbf{x}$ is reconstructed by using the patch $x_{i,j}$ of the image $\mathbf{x}_i$ with the coefficient $\alpha_{i,j}$, then the patch $x_{i,j}$ is likely to contain the content for the labels shared by the image $\mathbf{x}$ and $\mathbf{x}_i$. Moreover, the larger the reconstruction coefficient α_k is, the more likely the patch $x_{i,j}$ contains the shared labels. This observation naturally leads to a bi-directional label propagation between the selected atomic patches and the reference image or candidate regions. The fusion of the results from all such reconstructions yields the procedure for

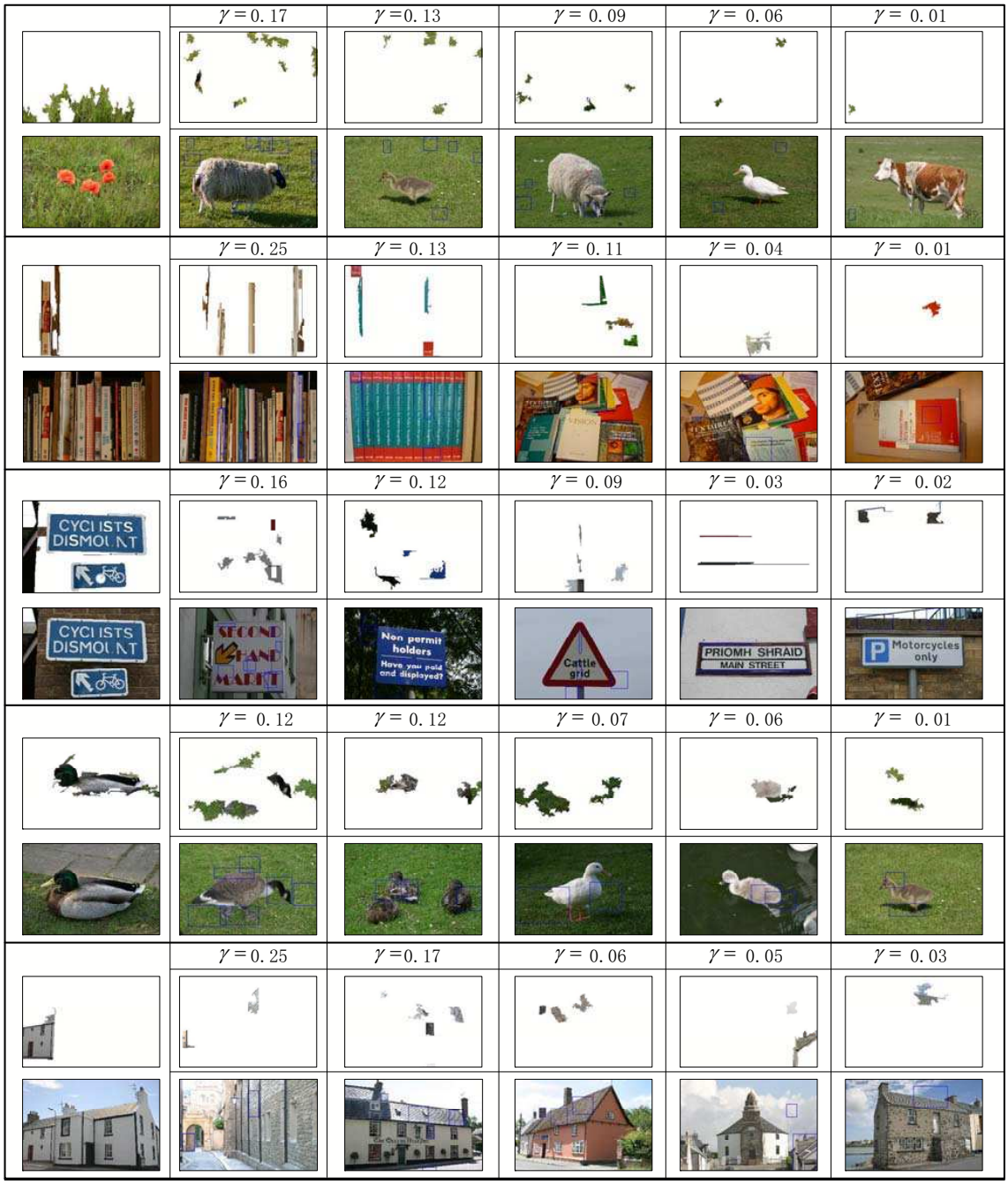


Figure 4.6: Exemplary results of bi-layer sparse coding for sparse image reconstruction from the MSRC database. For each row, the left subfigure shows the initially merged candidate region and its parent image, and the right subfigure shows the top few selected images and their selected patches.

label-to-region assignment.

The procedure contains four iterative steps:(1) choose an image $\mathbf{x}_i$ and its label vector z_i from the input image set, then collect and arrange the atomic patches of the remaining images in matrix A; (2) derive the bi-layer sparse solution $\hat{\alpha}$ of the equation $y_i = A\alpha$ by ℓ^1-norm minimization, where y_i is the representation for a candidate region (merged from over-segmented patches) of the reference image $\mathbf{x}_i$; (3) assign each selected atomic patch with the common labels shared by the image $\mathbf{x}_i$ and the image that the patch belongs to; and (4) assign the labels to the candidate region based on the labels of the selected patches and the coefficient vector α.

These four steps iterate by choosing each input image as reference image in turn, and the label vector $z_{i,c}$ of each atomic patch $x_{i,c}$ is obtained by cumulatively summing the label vector propagated to it in each iteration. Note that each candidate region comprises of several atomic patches, and the patch-level label vectors for the involved patches are updated in a cumulative way. In practice, after choosing an input image as reference image, we use a simple algorithm described in [61] to merge the spatially coherent and perceptually similar atomic patches to form the relatively larger-size candidate regions. We stop the merging if the region size is larger than a constant threshold, which is set as the 6000 pixels in this work. Figure 4.6(the first column) shows some candidate regions for performing the construction.

Algorithm 1 details the procedure for label-to-region assignment, where the inputs are the image set with annotated image labels and the outputs are the merged semantic regions with assigned image labels. Here, we would like to highlight some aspects of this label-to-region assignment procedure as follows:

1) The first step calls the image segmentation algorithm to obtain the over-segmented

Algorithm 1 . Procedure for Label-to-Region Assignment (LRA)

1: **Input:** Image set $\mathbf{X} = \{\mathbf{x}_i; i = 1, \ldots, N\}$ with label vector $z_i \in R^{N_c}$ for image $\mathbf{x}_i$;
Output: Semantic region set $\{O_j^i\}$ and their associated labels for the entire image set;

2: Partition each image $\mathbf{x}_i$ into a set of atomic patches with representations as $\{x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}\}$;

3: For $i = 1, 2, \ldots, N$,
  For $j = 1, 2, \ldots, n_i$,
    $z_{i,j}$= size of the (i, j)th patch / size of the ith image;

4: For each $i = 1, 2, \ldots, N$

  4.1: Set $A = [x_{1,1}, \ldots, x_{i-1,n_{i-1}}, x_{i+1,1}, \ldots, x_{N,n_N}]$;

  4.2: Group the atomic patches of image $\mathbf{x}_i$ to form candidate regions with representations as $\{g_k\}$, where g_k denotes the representation for the k-th candidate region.

  4.3: For each representation $y \in \{g_1, g_2, \ldots\}$,

    4.3.1: Solve the sparse solution $\hat{\alpha}$ of the system of equations $A\alpha = y$, according to the Eq. (4.10);

    4.3.2: Label propagation from the candidate region to the selected patches of the remaining images:
    For $j = 1, 2, \ldots, N$,
      For $k = 1, 2, \ldots, n_j$,

        i. if $j < i$, $z_{j,k} \leftarrow z_{j,k} + \hat{\alpha}_{j,k}(z_j \wedge z_i)$;

        ii. if $j \geq i$, $z_{j+1,k} \leftarrow z_{j+1,k} + \hat{\alpha}_{j,k}(z_{j+1} \wedge z_i)$;

    4.3.3: Label propagation from the remaining images to the candidate region y of the reference image:
    For each patch x within the candidate region,

        i. $z_x \leftarrow z_x + \sum_{j,j\neq i} \sum_k \hat{\alpha}_{j,k} \hat{\beta}_{j,k}(z_j \wedge z_i)$

5: Post-processing by calling the Algorithm 2 to merge atomic patches into semantic regions and obtain their labels.

patches for each input image. Note that as the generated patch is atomic, each patch generally corresponds to at most one label.

2) The second step initializes the label vector of atomic patch using the annotated

Algorithm 2 . Post-processing after Label Propagation

1: **Input:** Image label vector z_i, and the patch-level label vector $z_{i,j}$, $i = 1, \ldots, N, j = 1, \ldots, n_i$;
Output: Merged regions with semantic labels;

2: For $i = 1, 2, \ldots N$,

2.1: Calculate the number of image labels for the image $\mathbf{x}_i$, denoted as K_i;

2.2: Cluster the atomic patches in the label vector space, namely divide all the patch label vectors $\{z_{i,j}\}$ into K_i clusters, denoted as $\{O_1^i, \ldots, O_{K_i}^i\}$;

2.3: For each cluster $O_c^i \in \{O_1^i, \ldots, O_{K_i}^i\}$

2.3.1: Let z_m denote the weighted label vector for each cluster, calculated as $z_m = \sum_{z_{i,j} \in O_c^i} z_{i,j}$;

2.3.2: For each patch in O_c^i, set its label vector as z_m;

2.4: Merge those patches with the same label vector to form a semantic region, and the label is set as the one with the largest value in the label vector and without overlapping label with other region.

labels of its parent image, which have been manually annotated. The experiments empirically show the gain in algorithmic robustness achieved from this initialization.

3) The iterative procedure implements the one-vs-else label propagation scheme. Note that $\wedge$ denotes the and operator between two vectors.

4) For the post-processing step, the label assignment to region is implemented by selecting the region with the largest value in the label vector first, and then sequently performing the region annotation.

Finally, suppose the label vector of each atomic patch has been derived by Algorithm 1, we adopt the K-means clustering approach over the the label vectors of all patches to generate the label-to-region assignment results, whose overall procedure is summarized in Algorithm 2.

4.3 Direct Image Annotation by Bi-layer Sparse Coding

In this section, we show how the proposed bi-layer sparse coding formulation can be used for direct image annotation on new test images by propagating the labels from a set of training images with the annotated labels. For a given test image with the patch representations as $Y = [y_1, \cdots, y_{n_y}]$, we set the reference representation $y = \sum_i y_i$ or by merging several patches to form a candidate region. We then determine the sparse reconstruction coefficient matrix $\hat{\alpha}_1$ and $\hat{\gamma}_1$ by solving the problem in Eq. (4.10). The label vector of the test image can then be obtained as:

$$z_y = Z\,\bar{\gamma}, \tag{4.11}$$

where Z is the label matrix for all the training images. The labels with the largest values in z_y (or the sum of all obtained z_y's) are considered as the final annotations of the test image.

Compared with classical works for image annotation, the proposed bi-layer sparse coding based image annotation algorithm has the following characteristics: 1) the propagation process is robust and less sensitive to the image noises owing to the bi-layer sparse coding formulation; and 2) the proposed algorithm is scalable to large-scale, even web-scale, image retrieval by first selecting a set of visually related images and then performing bi-layer sparse coding over those roughly selected images.

4.4 Experiments

In this section, we systematically evaluate the effectiveness of our proposed bi-layer sparse coding formulation for both label-to-region assignment and image annotation tasks.

4.4.1 Data Sets

Three publicly available datasets, MSRC [63], COREL-4K [64] and NUS-WIDE [65], are used for all the experiments in this work. The MSRC dataset contains 591 images from 23 categories and provides the region-level ground-truths. There are about 3 labels on average for each image. We remove the classes which have less than 10 positive samples and images that are annotated with only one single label. This gives rise to 350 images and 18 categories: "building","grass", "tree","cow","horse", "sheep","sky","mountain", "aeroplane", "water","flower", "sign", "bird", "book", "chair", "road", "cat", and "dog". The COREL-4K dataset contains 4002 images of 11 categories chosen from the Corel Stock Photo CDs and each image is annotated with about 3.5 labels on average. As the original COREL dataset only provide image-level labels, we randomly select 100 images and manually annotate the region-level groundtruth for evaluation. This subset, named COREL-100, contains images from 7 categories of: "grass", "cow","snow", "Sky", "bear", "ground", "water". The third dataset, NUS-WIDE, was recently collected by the National University of Singapore (NUS), which contains a total of $269,648$ images in 81 categories and has about 2 labels per image on average. To avoid semantic overlapping (e.g. "animals" and "cow"), we choose the following 24 categories: "airport", "dog", "flags", "boats", "building", "mountain", "ocean", "road", "street", "sky", "sign", "tiger", "grass", "window","tower", "tree", "railroad", "sun", "train", "water",

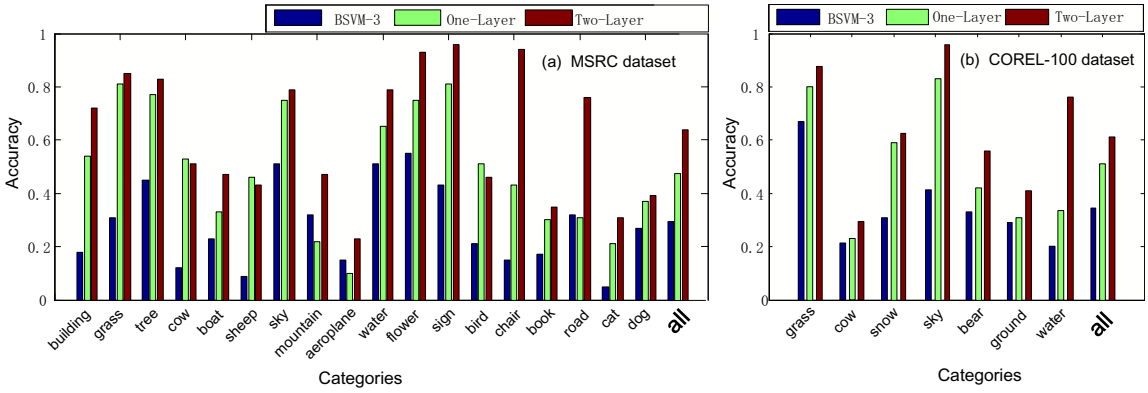


Figure 4.7: Detailed label-to-region accuracies for (a) MSRC dataset and (b) COREL-100 dataset. The horizontal axis shows the abbreviated name of each class and the vertical axis represents the label-to-region assignment accuracy.

”flowers”, ”plane”, ”snow”, and ”fire”. Finally we select the images with at least 5 annotated labels and obtain a subset with 1380 images.

These three datasets provide the image-level annotation labels for all images and hence can all be used for the experiments on imageannotation task. MSRC and COREL-100 additionally provide the region-level annotations, and thus they can both be used for the exam of label-to-region assignment task.

All the experiments are performed on an Intel Xeon X5450 workstation with 3.0GHz CPU and 16GB memory. The code is implemented on MATLAB platform. The Algorithm 1 can process 350 images (each of which is segmented into about 40 50 atomic patches) within 50 minutes. For the image annotation task, our method can reconstruct and predict a new test image (320×240 pixels) within 10 seconds (using the patch set collected from 350 images).

4.4.2 Exp-I: Label-to-Region Assignment

Parameters, Benchmarks and Metrics

We implement the proposed label to region assignment algorithm using the ℓ^1-Magic package [26]. It first translates Eq. (4.10) into a linear programming problem and then adopts the primal-dual algorithm to perform the optimization. In the implementation, we set the tolerance factor as 0.003 and the maximum number of primal-dual iterations as 50.

Another two free parameters are the maximal patch size $M1$ and maximal region size $M2$, both of which are used to control the segmentation algorithms [61]. The selection of these two parameters essentially makes a tradeoff between algorithmic performance and efficiency. Basically, the decrease of the patch size or region size shall increase the the computational cost or the iteration number for the reconstruction step in Algorithm 1, but may potentially increase the algorithmic performance. Therefore, we empirically set the two parameters as $M1 = 300$ pixels and $M2 = 6000$ pixels respectively in all tests.

Two algorithms are implemented as baselines for comparison to evaluate the effectiveness of the proposed bi-layer sparse coding formulation in label to region assignment task. One is the classical binary Support Vector Machine (BSVM), which translates the *m*-class multi-label classification problem into *m* binary classification problems. For each classifier, the image is considered as positive sample if it contains the specific concept/label, otherwise it is set as negative sample. In the training stage, we choose equally number of positive and negative samples and remove the overabundant ones to balance the training of SVM. In the testing stage, we first apply each classifier on the atomic

patch to obtain the probability of the patch to be positive sample. The results from the m classifiers are then fused to generate the m-dimensional label confidence vectors, which are further processed by Algorithm 2 to obtain the labels of those merged regions. Note that the training and testing procedures work at two different levels of the images, and the goal is to eventually obtain the semantic annotations at the region-level. For a fair and reliable comparison, we implement this baseline algorithm on over-segmented patches with different allowed maximal sizes, including 150, 200, 400 and 600 pixels. The binary SVM is implemented based on the lib-SVM library [66] and the Gaussian Radial Basis Function kernel is used by setting the kernel parameter as 1.

The second baseline algorithm is a simplification of the proposed solution, called one-layer sparse coding, which is used to demonstrate the improvement brought by the proposed bi-layer sparse coding formulation. The overall procedure is similar to Algorithm 1, except that the system of equation to perform the optimization is the Eq. (4.6) in Section 2.4. We set the parameters the same as that for the algorithm based on bi-layer sparse coding formulation.

The label-to-region performance is evaluated in both qualitative and quantitative ways. The quantitative label-to-region assignment accuracy measures as the percentage of pixels with agreement between the assigned label and ground truth.

Results and Analysis

Table 4.1 shows the accuracy comparison between the baseline algorithms and our proposed algorithm on the MSRC and COREL datasets. The detailed comparison results for individual classes are illustrated in Figure 4.7. In our implementation, all the images are resized to the resolution of 320×240 pixels, and the modified segmentation

Table 4.1: Label-to-region assignment accuracy comparison on MSRC and COREL-100 datasets. The SVM-based algorithm is implemented with different values for the parameter of maximal patch size, namely, SVM-1: 150 pixels, SVM-2: 200 pixels, SVM-3: 400 pixels, and SVM-4: 600 pixels.

Dataset	SVM-1	SVM-2	SVM-3	SVM-4	One-layer	Bi-Layer
MSRC	0.22	0.20	0.24	0.23	0.47	**0.63**
COREL-100	0.29	0.32	0.33	0.31	0.51	**0.61**

algorithm [61] is applied to obtain the initial ensemble of atomic patches. From these results, we can have the following observations. (1) The proposed algorithm achieves much higher accuracies of 0.63 and 0.61 on the MSRC and COREL-100 dataset respectively as compared to SVM-based baseline. This clearly demonstrates the effectiveness of the bi-layer sparse coding for relating the image or candidate region with the atomic pathes. 2) Bi-layer coding based algorithm outperforms the one-layer based algorithm over both two datasets. This is because the Bi-layer sparse coding formulation enforces the usage of merged patches for the reconstruction of the image or candidate regions. It greatly improves the quality of construction for label propagation and thus boosts the accuracy for label-to-region assignment. (3) On analysis of detailed class-level results shown in Figure 4.7, it is noted that our algorithm seems to be less effective for handling the categories for foreground objects, such as dogs, cows, and cats compared to background regions such as streets, trees and sky etc. This is because the labels of object classes have lower weights (normalized region size) as compared to the background regions. Although we can learn individual classifiers for these specific objects as in [46, 43, 50, 51, 52, 53] and then apply them to detect and localize objects in images, these algorithms generally require training data with ground-truths at the region-level and thus are not applicable for this general prior-free label-to-region assignment task.

Note that we do not compare our solution to label-to-region assignment task with

those typical algorithms for simultaneously classifying and localizing objects in images, for three reasons: a) our proposed solution works under the assumption that no region-level label annotation is provided, which is however the general prerequisite for most typical algorithms; b) most typical algorithms are tailored to specific objects and can only work on cases with limited number of object categories; and 3) for each image label, those typical algorithms need to learn an individual detector, which is thus very time consuming and more difficult to be applied for large-scale applications.

Some example results on label-to-region assignment are displayed in Figure 4.8 and 4.9 for the MSRC and COREL-100 datasets, respectively. These results over various conditions well validate the effectiveness of our proposed solution. Note that our proposed algorithm is scalable to large-scale applications, though we do not report results on larger-size dataset (mainly due to the tedious annotation for providing ground-truths, which is also the main motivation of this work) here. The algorithm is essentially fast if we utilize the priors to remove the input images without common labels with the reference image, and also the entire algorithm is suitable for parallel computation. Accordingly, more samples with abundant labels are able to provide more contextual information, which shall further boost the overall performance of Algorithm 1.

4.4.3 Exp-II: Image Annotation on Test Images

Benchmarks and Metrics

Three popular algorithms are implemented as benchmark baselines for the image annotation task. (1) Binary SVM [67] (**BSVM**), which translates the m-class multi-label classification problem into m binary classification problems. We use the same setting

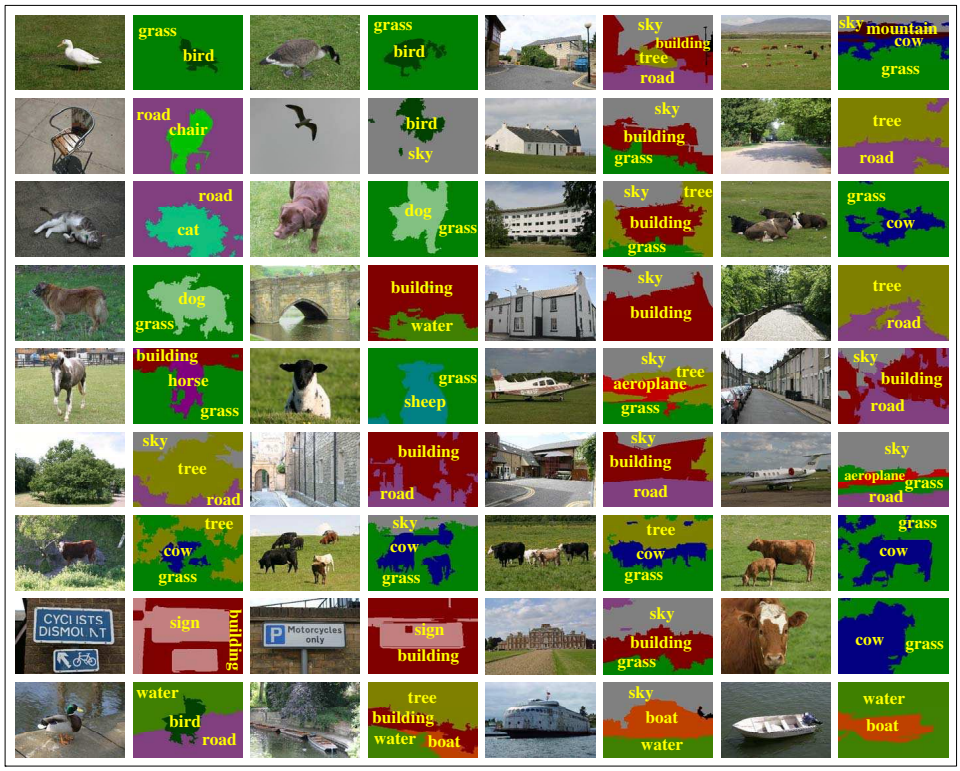


Figure 4.8: Example results on label-to-region assignment. The images are from the MSRC dataset. The original input images are shown in the columns $1, 3, 5, 7$ and the corresponding labeled images are shown in the columns $2, 4, 6, 8$. Each color in the result images denotes one class of localized region.

Figure 4.9: Example results on label-to-region assignment from the COREL dataset.

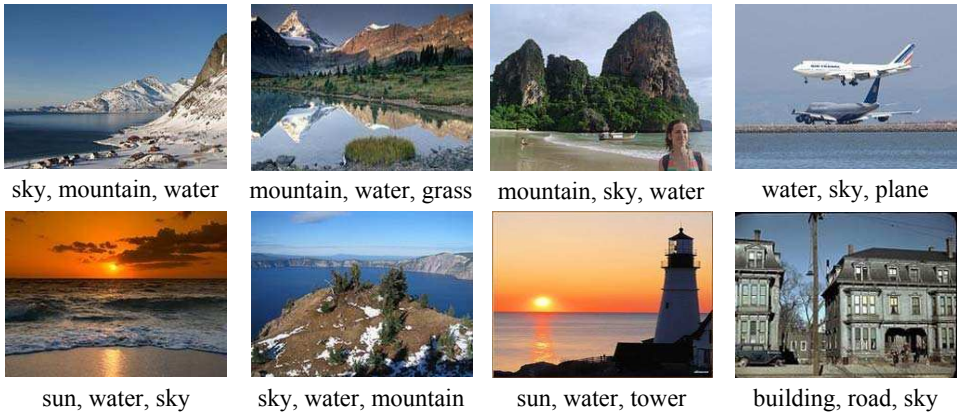


Figure 4.10: Some example results on image annotation from the NUS-WIDE dataset.

as that for the label-to-region assignment experiments. (2) The Correlated Label Propagation algorithm (**CLP**) proposed by Feng *et al.* in [58]. It provides several choices for the kernel type in the objective function, and in this work, we use the exponential function with parameter $\alpha = 0.6$ (not the α in Eq. (4.10)), which usually achieves the best performance as reported in [58]. (3) The KNN based multi-label learning algorithm (**MLKNN**) proposed by Zhuang *et al.* [59]. We set the number of nearest neighbors to 15 and keep other parameters the same as in [59].

CLP and MLKNN are the state-of-the-art multi-label annotation algorithms in literature. They have been reported to outperform most other multi-label annotating algorithms, such as rank-SVM [68] and boost.MH [69]. Thus we do not plan to further implement the latter two in this work. We evaluate and compare among the four algorithms over three datasets, MSRC, COREL-4K and NUS-WIDE, each of which is evenly split into training and testing subset.

The image annotation performance is measured by mean average precision, which is widely used for evaluating the performances of ranking related tasks.

Table 4.2: Image label annotation MAP (Mean Average Precision) comparisons among four algorithms on three different datasets.

Dataset	BSVM [67]	CLP [58]	MLKNN [59]	**Ours**
MSRC	0.30	0.54	0.65	**0.70**
COREL-4k	0.41	0.53	0.61	**0.72**
NUS-WIDE	0.56	0.64	0.63	**0.76**

Results and Analysis

The comparison results on image annotation performance are reported in Table 4.2, where each row shows the achieved MAP (mean average precision) over different datasets. From these results, we can derive the following observations. (1) The proposed method based on bi-layer sparse formulation outperforms the three baselines over all datasets. (2) The latter three methods all use the label contextual information, and achieve much higher performance than the SVM-based algorithm. This also accords with the motivation of our proposed solution, which takes the advantage of the label contextual information for label propagation. Figure 4.10 illustrates some exemplar image annotation results from the NUS-WIDE dataset. The images are challenging due to the large intra-class varieties and the usually inter-class occlusions.

4.5 Conclusion

In this chapter, we proposed a novel sparse coding technique for addressing the problem of label-to-region assignment, which only requires image-level label annotations. With the popularity of the photo sharing websites, the community-contributed images with rich tag information are becoming much easier to obtain, it is predicated that the key-

word query based semantic image search can greatly benefit from applying our proposed technique for label-to-region assignment on these tagged images.

Our proposed solution for both label-to-region assignment and image annotation tasks is applicable for handling large-scale dataset. For the label-to-region assignment task: 1) the images can first be clustered according to the image label information, and the proposed solution can then be applied within each cluster; and 2) the priors can be utilized to remove the input images without common labels with the reference image for the sparse reconstruction of the reference image or its candidate regions. For the image annotation task, we can first roughly select a sufficiently large set of visually similar images of the reference image, and then apply the bi-layer sparse coding formulation on these selected images for image annotation.

Chapter 5

Multi-task Low-rank Affinity Pursuit for Image Segmentation

5.1 Introduction

The task of *image segmentation* [70] is widely accepted as a crucial function for high-level image understanding. As pointed out by [71, 72, 14], a successful image segmentation algorithm can significantly reduce the complexity of object segmentation and recognition, which form the core of high-level vision. Hence, it has been widely studied in computer vision [73, 74, 75, 76, 61, 77, 78, 79, 80]. Generally, image segmentation is a comprehensive task which is related with several cues, $e.g.$ regions, contours and textons [81]. In particular, in this chapter we are interested in region-based methods [82], which aim at partitioning an image into homogenous regions by grouping together the basic image elements (e.g., superpixels) with similar appearances.

Many efforts have been devoted to this topic (e.g., [83, 84, 14, 85]). However, some

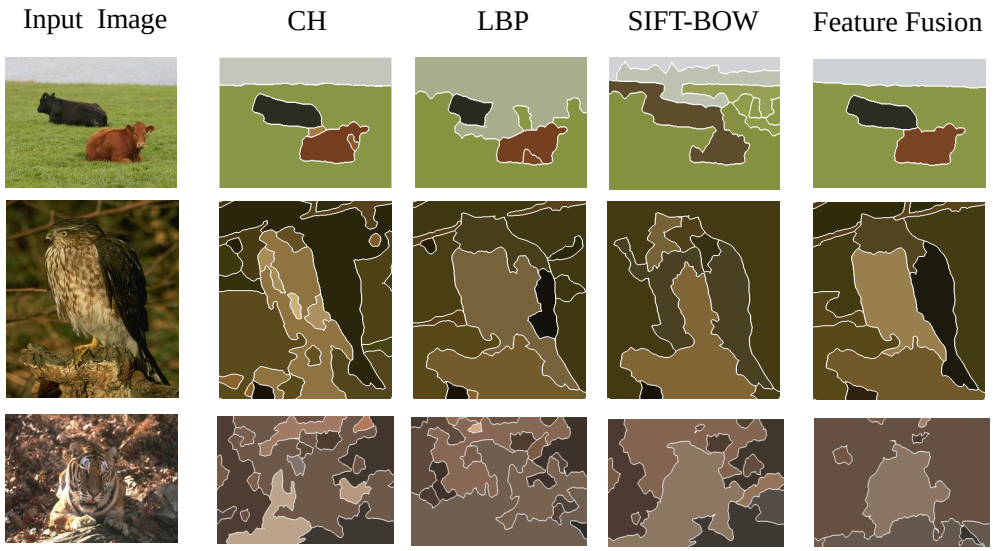


Figure 5.1: Illustration of the necessity and superiority of fusing multiple types of features. From left to right: the input images; the segmentation results produced by CH; the results produced by LBP; the results produced by SIFT based bag-of-words (SIFT-BOW); the results produced by integrating CH, BLP and SIFT-BOW. These examples are from our experiments.

critical problems remain unsolved. Most existing region-based methods focused on exploring the criteria, such as the widely used normalized cut (NCut) [14] and the recently established minimum description length (MDL) [84], for seeking the optimal segmentation. The feature space is however usually predetermined by mildly choosing a feature descriptor such as the color histograms (CH). However, as a data clustering problem, image segmentation performance heavily depends on the choice of the feature space. What is more, it is hard to find a single feature descriptor that can generally work well for various images with diverse properties, since each feature descriptor generally has its own advantages and limitations: the CH descriptor is very informative for describing color, but inappropriate for describing other visual information; the local binary pattern (LBP) [86] descriptor can defend the change of light conditions, but may cause some loss of information; the scale invariant feature transform (SIFT) [87] descriptor can be invariant to some image transformations, but some useful information of the original image may

also be lost. Hence, it is crucial to establish a good solution that can integrate multiple types of image features for more accurate and reliable segmentation. Figure 5.1 further illustrates the necessity and superiority of fusing multiple types of features.

Although image segmentation may intuitively benefit from the integration of multiple features, to the best of our knowledge, there is no previous work that intensively explores the fusion of multiple features in region-based segmentation. This is mainly due to the fact that it is actually not easy to well handle the multiple features of various properties. In machine learning community, the methods towards this issue are also quite limited. Possibly, the multi-view spectral clustering technique established by Zhou et al. [88] is an optional choice. Namely, one could first construct an undirected (or directed) graph by inferring an affinity matrix from each type of image features, resulting in a multi-view graph (there are multiple affinities between each pair of nodes), and then obtain the segmentation results by combing those multiple affinity matrices [88]. However, this option may not fully capture the advantages of multiple features, because the affinity matrices are still computed from different features *individually*, and thus the cross-feature information is not well considered during the inference process.

To make effective use of multiple features, in this chapter we introduce the so-called multi-task low-rank affinity pursuit (MLAP) method, which aims at inferring a unified affinity matrix from multiple feature spaces, and thus producing accurate and reliable segmentation results. Like the traditional methods such as NCut [14], we also treat image segmentation as a graph partitioning problem. That is, an image is represented as an undirected graph with each node corresponding to a *superpixel* [89]. Then the segmentation can be done by partitioning the nodes of the graph into groups. Unlike existing methods, which usually adopt a single feature space, each node (superpixel) in our method is described by multiple features with different properties. To integrate those multiple

features and make effective use of the cross-feature information, our MLAP method infers a unified affinity matrix by seeking the sparsity-consistent low-rank affinities from the joint decompositions of multiple feature matrices into pairs of sparse and low-rank matrices, the latter of which is the production of image feature matrix and its corresponding low-rank affinity matrix. The inferring process is formulated as a constrained nuclear norm and $\ell_{2,1}$-norm minimization problem, which is convex and can be solved efficiently with augmented Lagrange multiplier (ALM) [90] method. Provided with the affinity matrix encoding the similarities among superpixels, the final segmentation result can be simply obtained by applying the NCut algorithm to the inferred affinities. Compared with existing methods, the contributions of this work mainlyinclude:

- We propose a method for learning a unified affinity matrix from multiple feature spaces and so performing image segmentation collaboratively. Since the cross-feature information has been well considered, such a joint inference scheme can produce more accurate and reliable results than those methods directly combining multiple affinity matrices, each of which is learnt individually.

- We introduce a simple yet effective new image segmentation algorithm that achieves comparable performance with the state-of-the-art methods, as demonstrated on the MSRC database [63] and the Berkeley dataset [74].

5.2 Image Segmentation by Multi-task Low-rank Affinity Pursuit

5.2.1 Problem Formulation

For efficiency, superpixels other than image pixels are used as basic image elements. Using the over-segmentation algorithm in [89], a given image is partitioned into subregions, each of which is called a superpixel. In this way, the problem of segmenting the image is cast into clustering the superpixels into groups. By choosing an appropriate feature descriptor to describe each superpixel, the image segmentation problem can be formulated as follows.

Problem 5.2.1 *Let $X = [x_1, x_2, \cdots, x_N]$ be a feature matrix, each column of which is a feature vector x_i corresponding to a superpixel P_i. Then the task is to segment the superpixels into groups according to their features represented by X.*

A weak point of the above definition is that only one type of features is considered. To boost the performance, the problem definition based on multiple types of features can be formulated as below.

Problem 5.2.2 *Let $X_1, X_2, \cdots, X_K$ be K feature matrices for K types of features, where the columns in different matrices with the same index correspond to the same superpixel. Then the task is to segment the superpixels into groups by integrating the feature matrices $X_1, \cdots, X_K$.*

5.2.2 Multi-task Low-rank Affinity Pursuit

For easy of understanding, Problem 5.2.1 is explored first. Accordingly, the case for the multiple types of features targeting on Problem 5.2.2 will be further examined with a well-established solution.

Single-Feature Case (Problem 5.2.1)

According to the explorations in [91], the superpixel data from natural image usually has a structure of low-rank subspace, i.e., the task of segmenting the superpixels into homogeneous regions could be cast as segmenting the feature vectors into their respective subspaces. Hence, the task stated in Problem 5.2.1 may be handled by the subspace segmentation algorithms, e.g., the Sparse Subspace Clustering (SSC) algorithm presented in [92] and the Low-Rank Representation (LRR) algorithm presented in [93]. LRR based modeling is chosen for single-feature case owing to its effectiveness and robustness. Namely, for a matrix $X = [x_1, x_2, \cdots, x_N]$ with each x_i representing the i-th superpixel, the affinities among superpixels are computed by solving the following LRR problem:

$$\begin{aligned} \min_{Z_0, E_0} \quad & \|Z_0\|_* + \lambda \|E_0\|_{2,1}, \\ \text{s.t.} \quad & X = XZ_0 + E_0, \end{aligned} \tag{5.1}$$

where $\|\cdot\|_*$ denotes the nuclear norm, also known as the trace norm or Ky Fan norm (sum of the singular values), $\|\cdot\|_{2,1}$ is the $\ell_{2,1}$-norm [93, 94] for characterizing noise and the parameter $\lambda > 0$ is used to balance the effects of the two parts.

According to [94, 93], the optimal solution Z_0^* (with respect to the variable Z_0) to

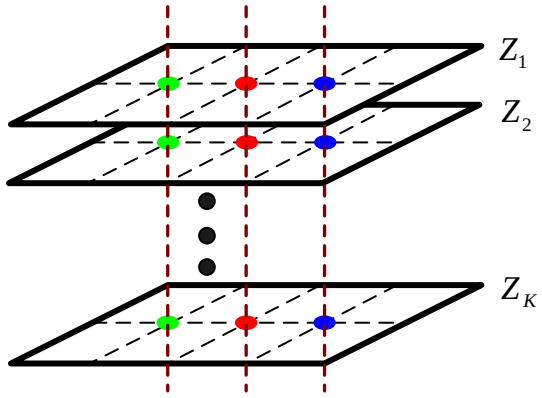


Figure 5.2: Illustration of the $\ell_{2,1}$-norm regularization defined on Z. Generally, this technique is to enforce the matrices $Z_i, i = 1, 2, \cdots, K$, to have sparsity consistent entries.

problem (5.1) naturally forms an affinity matrix that represents the pairwise similarities among superpixels. Namely, the affinity $\mathbb{S}_{ij}$ between two superpixels P_i and P_j could be computed by $\mathbb{S}_{ij} = |(Z_0^*)_{ij}| + |(Z_0^*)_{ji}|$, where $(\cdot)_{ij}$ denotes the (i, j)-th element of a matrix. Provided with such symmetric affinities, the NCut method in [14] can be applied for producing the final image segmentation results.

Multi-feature Case (Problem 5.2.2)

The above LRR can only be used to a certain type of visual features and not directly applicable for multi-feature cases. For multiple feature integration, an intuitive approach is to directly combine the affinity matrices individually inferred by LRR. The combination can be done by simply adding together multiple affinities or utilizing the multi-view spectral clustering technique presented in [88] to produce the final segmentation results. However, the inference of the individual affinity matrix does not well utilize the cross-feature information, which is crucial to produce accurate and reliable results.

For effectively fusing multiple features, we propose a new solution of multi-task

low-rank affinity pursuit (MLAP) that aims at *jointly* inferring a collection of affinity matrices $Z_1, Z_2, \cdots, Z_K$, where each $N \times N$ matrix Z_i corresponds to the i-th feature matrix X_i. Here, our consideration for formulating the inference process is two-side: to inherit the advantages of LRR, the affinity matrices should be encouraged to be of low-rank; to make effective use of the cross-feature information, the affinity matrices may be enforced to be sparsity-consistent. By considering both sides, the affinity matrices $Z_1, Z_2, \cdots, Z_K$ in MLAP are inferred by solving the following convex optimization problem:

$$\begin{aligned} \min_{\substack{Z_1,\cdots,Z_K \\ E_1,\cdots,E_K}} \quad & \sum_{i=1}^{K}(\|Z_i\|_* + \lambda\|E_i\|_{2,1}) + \alpha\|Z\|_{2,1}, \\ \text{s.t.} \quad & X_i = X_i Z_i + E_i, i = 1, \cdots, K, \end{aligned} \tag{5.2}$$

where $\alpha > 0$ is a parameter and the $K \times N^2$ matrix Z is formed by concatenating $Z_1, Z_2, \cdots, Z_K$ together as the following:

$$Z = \begin{bmatrix} (Z_1)_{11} & (Z_1)_{12} & \cdots & (Z_1)_{NN} \\ (Z_2)_{11} & (Z_2)_{12} & \cdots & (Z_2)_{NN} \\ \vdots & \vdots & \ddots & \vdots \\ (Z_K)_{11} & (Z_K)_{12} & \cdots & (Z_K)_{NN} \end{bmatrix}.$$

The $\ell_{2,1}$-norm regularization defined on Z plays a key role in our MLAP method: it is the minimization of $\|Z\|_{2,1}$ that enforces the affinities $(Z_l)_{ij}, l = 1, 2, \cdots, K$, to have consistent magnitudes, all either large or small, as shown in Figure 5.2. That is, the

Algorithm 3 Image Segmentation by MLAP

Input: An image and the required parameters.

1. Separate the image into superpixels by using the algorithm in [89].
2. Compute K feature matrices by extracting K types of features to describe each superpixel.
3. Obtain the sparsity-consistent low-rank affinity matrices $Z_1, \cdots, Z_K$ by solving problem (5.2), and define the edge weights of an undirected graph according to (5.3).
4. Use NCut to segment the nodes of the graph into a pre-specified number of groups.

Output: A map that encodes the segmentation result.

fusion of multiple features is "seamlessly" performed by minimizing the $\ell_{2,1}$-norm of Z. Without this regularization term, the formulation (5.2) will reduce to a "trivial" method that is equal to applying LRR to each feature matrix X_i individually.

Let $(Z_1^*, Z_2^*, \cdots, Z_k^*)$ be the optimal solution to problem (5.2). To obtain a unified affinity matrix, we only need a simple step to quantify the columns of the matrix Z:

$$\mathbb{S}_{ij} = \frac{1}{2}(\sqrt{\sum_{l=1}^{K}(Z_l)_{ij}^2} + \sqrt{\sum_{l=1}^{K}(Z_l)_{ji}^2}). \tag{5.3}$$

Note here that $\sqrt{\sum_{l=1}^{K}(Z_l)_{ij}^2}$ is right the ℓ_2-norm of the $((i-1)n+j)$-th column of Z used in (2) and thus (3) should not be considered as late fusion of Z_i's. Same as single feature case, the NCut method can be applied on such affinity matrix to produce the final image segmentation results. Algorithm 3 summarizes the entire image segmentation algorithm of MLAP.

5.2.3 Optimization Procedure

Problem (5.2) is convex and can be optimized in polynomial time. We first convert it into the following equivalent problem:

$$\begin{aligned} \min_{\substack{J_1,\cdots,J_K,\\ S_1,\cdots,S_K\\ Z_1,\cdots,Z_K\\ E_1,\cdots,E_K}} \quad & \sum_{i=1}^{K}(\|J_i\|_* + \lambda\|E_i\|_{2,1}) + \alpha\|Z\|_{2,1}, \\ \text{s.t.} \quad & X_i = X_iS_i + E_i, \\ & Z_i = J_i, \\ & Z_i = S_i, i = 1,\cdots,K. \end{aligned} \tag{5.4}$$

This problem can be solved with the augmented Lagrange multiplier (ALM) method [90], which minimizes the following augmented Lagrange function:

$$\begin{aligned} \alpha\|Z\|_{2,1} + \sum_{i=1}^{K}(\|J_i\|_* + \lambda\|E_i\|_{2,1}) + \sum_{i=1}^{K}(\langle W_i, Z_i - J_i\rangle + \\ \langle Y_i, X_i - X_iS_i - E_i\rangle + \langle V_i, Z_i - S_i\rangle + \\ \frac{\mu}{2}\|X_i - X_iS_i - E_i\|_F^2 + \frac{\mu}{2}\|Z_i - J_i\|_F^2 + \frac{\mu}{2}\|Z_i - S_i\|_F^2), \end{aligned}$$

where $Y_1,\cdots,Y_K$, $W_1,\cdots,W_K$ and $V_1,\cdots,V_K$ are Lagrange multipliers, and $\mu > 0$ is a penalty parameter. The inexact ALM method [90], also called alternating direction method (ADM) [90], is outlined in Algorithm 4. Note that the sub-problems of the algorithm are convex and they all have closed-form solutions. Step 1 is solved via the

singular value thresholding operator [95], while Steps 3 and 4 are solved via Lemma 3.2 of [93].

5.2.4 Discussions

On the Optimization Algorithm

Since ADM is a variation of the exact ALM method whose convergence properties have been generally proven, Algorithm 5.1 should converge well in practice, although proving the convergence properties of ADM in theory is still an open issue [96]. Supposing the number of superpixels is N, then the computation complexity of Algorithm 1 is $O(N^3)$, which is practical (note that the complexity is actually the same as computing the SVD of an $N \times N$ matrix). In our experiments, since the number of superpixels N is small ($N \approx 100$), the computational cost of Algorithm 5.1 is actually low. On an Intel Xeon $X5450$ workstation with 3.0 GHz CPU and 16GB memory, for example, it takes about 20 seconds to finish the computation for an image.

On the Extension to Multiple Visual Cues

Since our current MLAP method requires the features to be represented by vectors, it can only model image regions, which however only form one aspect of the visual cues. Generally, as pointed out by Malik et al. [74, 81], the other cues such as contour and spatial information should also be taken into account. Fortunately, it is actually feasible for our method to handle multiple cues. Namely, the affinity matrix can be learnt by jointly utilizing the information supplied by other cues. For example, when two superpixels P_i and P_j are separated by a strong contour, the edge between them may be removed (i.e.,

Algorithm 4 Solving Problem (5.2) by ADM

Inputs: Data matrices $\{X_i\}$, parameters λ and α.

while not converged **do**

1. Fix the others and update $J_1, \cdots, J_K$ by

$$J_i = \arg\min_{J_i} \frac{1}{\mu}||J_i||_* + \frac{1}{2}||J_i - (Z_i + \frac{W_i}{\mu})||_F^2.$$

2. Fix the others and update $S_1, \cdots, S_K$ by

$$S_i = (\mathtt{I} + X_i^T X_i)^{-1}(X_i^T(X_i - E_i) + Z_i + \frac{X_i^T Y_i + V_i - W_i}{\mu}).$$

3. Fix the others and update Z by

$$Z = \arg\min_{Z} \frac{\alpha}{2\mu}\|Z\|_{2,1} + \frac{1}{2}\sum_{i=1}^{K}\|Z - M\|_F^2,$$

where M is a $K \times N^2$ matrix formed as follows:

$$M = \begin{bmatrix} (F_1)_{11} & (F_1)_{12} & \cdots & (F_1)_{nn} \\ (F_2)_{11} & (F_2)_{12} & \cdots & (F_2)_{nn} \\ \vdots & \vdots & \ddots & \vdots \\ (F_K)_{11} & (F_K)_{12} & \cdots & (F_K)_{nn} \end{bmatrix},$$

where $F_i = (J_i + S_i - (W_i + V_i)\mu)/2, i = 1, \cdots, K$.

4. Fix the others and update $E_1, \cdots, E_K$ by

$$E_i = \arg\min_{E_i} \frac{\lambda}{\mu}||E_i||_{2,1} + ||E_i - (X_i - X_i S_i + \frac{Y_i}{\mu})||_F^2.$$

5. Update the multipliers

$$\begin{aligned} Y_i &= Y_i + \mu(X_i - X_i S_i - E_i), \\ W_i &= W_i + \mu(Z_i - J_i), \\ V_i &= V_i + \mu(Z_i - S_i). \end{aligned}$$

6. Update the parameter μ by $\mu = \min(\rho\mu, 10^{10})$
($\rho = 1.1$ in all experiments).

7. Check the convergence condition: $X_i - X_i S_i - E_i \to 0$, $Z_i - J_i \to 0$ and $Z_i - S_i \to 0$, $i = 1, \cdots, K$.

end while

Output: Z.

set $\mathbb{S}_{ij} = 0$) or give high penalty to Z_{ij}'s. In a similar way, our method may also model the spatial information. We leave these as our future work.

5.3 Experiments

5.3.1 Experiment Setting

Datasets

We use two publicly available databases, the MSRC [63] dataset and the Berkeley [74] segmentation dataset, in our experiments. The MSRC dataset consists of 591 images from 23 categories. Here we use the cleaned up ground-truth object instance labeling [97], which is cleaner and more precise than the original data. The Berkeley segmentation dataset is comprised of 500 natural images, which cover a variety of nature scene categories, such as portraits, animals, landscape, beaches and so on. It also provides ground-truth segmentation results of all the images obtained by several human subjects. On average, five segmentation maps are available per image.

Superpixel and Features

As aforementioned, we need to partition each image into superpixels. There are several methods that can be used to obtain a superpixel initialization, such as those from Mori et al. [89], Felzenszwalb et al. [61] and Ren et al. [98]. Here we use the method in [89] and the number of superpixels for each image is set to be around 100 in our experiments.

After the superpixel initialization, three different types of features, color histogram (CH), local binary pattern (LBP) and bag-of-visual-words (BOW), are used to describe the appearance of each superpixel. Color histogram (CH) represents the number of pixels that have colors in each of a fixed list of color ranges. It can be built for any kind of color space such as RGB or HSV. In the experiments we use the RGB histogram. The local binary pattern (LBP) operator describes each pixel by the relative gray-levels of its neighbor pixels. In our experiments, the LBP codes are computed using 8 sampling points on a circle of radius 1. Bag-of-visual-words (BOW) is also applied to describe the appearance of the superpixels. In the experiments we compute the SIFT [87] features for each pixel and then perform the vector quantization to construct the visual vocabulary by K-means clustering approach. The number of clustering centers is set to be 50 for MSRC and 100 for Berkeley.

Baselines and Evaluation

To demonstrate the advantages of fusing multiple types of features, we consider the individual performance of applying LRR to a certain single feature, resulting in three benchmark baselines: LRR (CH), LRR (LBP) and LRR (BOW). Moreover, we also report the performance of our MLAP solution when integrating two types of features, which results in three competing methods: MLAP (CH+LBP), MLAP (CH+BOW) and MLAP (LBP+BOW). To show the superiority of our formulation in (5.2), we also consider some "naive" methods for fusing multiple types of features, including using the multi-view technique to combine multiple affinity matrices (denoted as "Multi-view"), an approach of stacking together multiple types of features to form a unified long vector (denoted as "Vector-stack"), a widely used approach of utilizing NCut to combine multiple affinity matrices, and a benchmark approach that performs the Mean-shift [75]

Table 5.1: Evaluation results on the MSRC dataset and the Berkeley 500 segmentation dataset. The details of all the algorithms are presented in Section 5.3.1. The results are obtained over the best tuned parameters for each dataset (the parameters are uniform for an entire dataset). For comparison, we also include the results reported in [3], but note that, for the Berkeley dataset, [3] used Berkeley 300 instead.

	MSRC			Berkeley		
	VOI	PRI	CR	VOI	PRI	CR
MLAP(CH+LBP+BOW)	**1.1656**	**0.8306**	**0.7556**	**1.5311**	**0.8538**	**0.6411**
MLAP(CH+LBP)	1.1931	0.8020	0.7121	1.6573	0.8401	0.6227
MLAP(CH+BOW)	1.2505	0.7967	0.7032	1.7262	0.8320	0.6109
MLAP(LBP+BOW)	1.4245	0.7560	0.6541	1.7626	0.8305	0.5947
LRR(CH)	1.3002	0.7912	0.6932	1.7475	0.8295	0.5905
LRR(LBP)	1.4449	0.7490	0.6415	1.7875	0.8261	0.5734
LRR(BOW)	1.4880	0.7343	0.6275	1.8585	0.8045	0.5670
Multi-View	1.2511	0.8116	0.7194	1.6664	0.8441	0.6272
Vector-stack	1.4107	0.7728	0.6668	1.8993	0.7815	0.5516
NCut	1.2516	0.8052	0.7075	1.7235	0.8283	0.6054
Mean-shift	1.7472	0.7307	0.5983	2.0872	0.7196	0.5272
Ma et al. [3]*	1.49	0.76	–	1.76	0.80	–

operator on multiple features (denoted as "Mean-shift"). For presentation convenience, we refer to our method of integrating all three features as "MLAP(CH+LBP+BOW)".

To quantitatively evaluate the performance of various methods, three metrics for comparing pairs of image segmentations are used: the variation of information (VOI) [99], the probabilistic rand index (PRI) [100] and the segmentation covering rate (CR) [101].

5.3.2 Experiment Results

Main Results

We evaluate and compare all the algorithms under a unified setting, as discussed in Section 5.3.1. In summary, the results shown in Table 5.1 well verify the advantage of

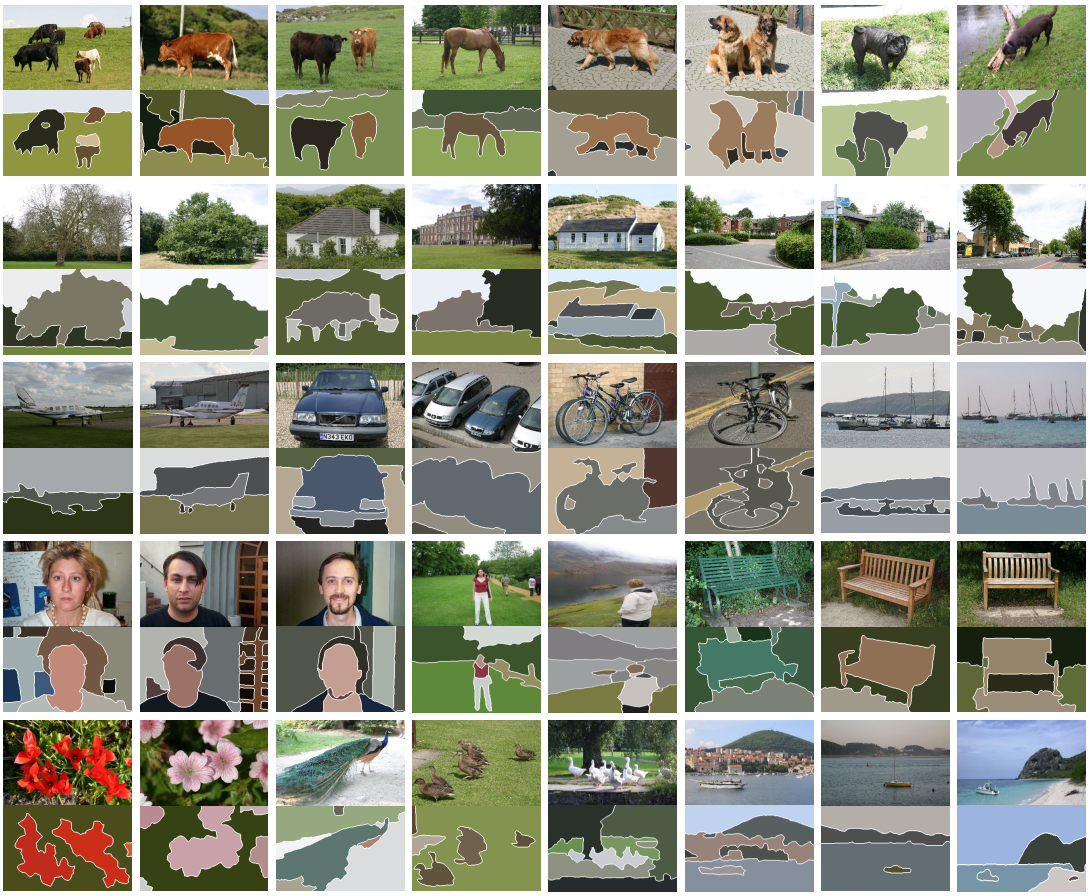


Figure 5.3: Some examples of the segmentation results on the MSRC database, produced by our MLAP method.

fusing multiple types of features and the superiority of our MLAP method. Namely, while all three features are combined together for segmentation, our method distinctly outperforms the other methods (which also use multiple types of features), including Multi-view, Vector-stack, NCut and Mean-shift. These results illustrate the effectiveness of our formulation (5.2), which generally learns a unified affinity matrix from multiple feature spaces. Compared with the approach of applying LRR to a certain single feature, again, the results in Table 5.1 clearly show the advantage of fusing multiple types of features. Figure 5.3 and Figure 5.4 show some examples of image segmentation results. It can be seen that the segmentation results produced by MLAP are quite promising.

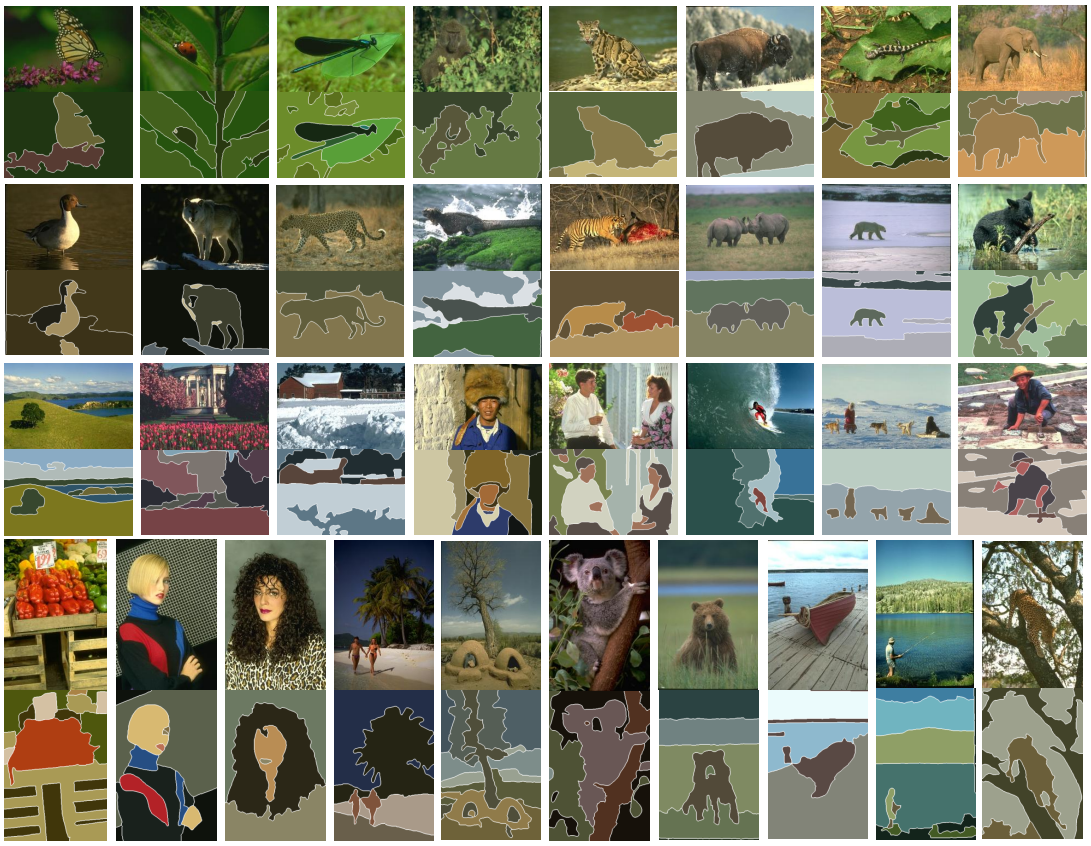


Figure 5.4: Some examples of the segmentation results on the Berkeley dataset, produced by our MLAP method.

Comparison to State-of-the-art Methods

To evaluate the competitiveness of the proposed solution, we also compare the results with other state-of-the-art methods, mainly including Rao et al. [3] and Arbelaez et al. [74]. Our method distinctly outperforms the results reported in [3], which achieves a PRI (higher is better) of 0.8 and a VOI (lower is better) of 1.76 on the Berkeley dataset. Whereas, as shown in Table 5.1, our MLAP method can obtain a PRI of 0.8538 and a VOI of 1.5311. On the Berkeley dataset, our results are better than the results reported from Arbelaez et al. [74] under the optimal dataset scale (VOI=1.69, PRI=0.83 and CR=0.59), and are close to their results under the optimal image scale (VOI=1.48,

PRI=0.86 and CR=0.65). On the MSRC database, our results are better than their results obtained under the optimal dataset scale (CR=0.66), and are also slightly better than their optimal image scale results (CR=0.75). These results illustrate that our solution is competitive for image segmentation. It is also worth noting that our current method may be further boosted by integrating other visual cues, e.g., contour and spatial information, as discussed in Section 5.2.4.

5.4 Conclusion

In this chapter, we presented a novel image segmentation framework called multi-task low-rank affinity pursuit (MLAP). In contrast with existing single-feature based methods, MLAP integrates the information of multiple types of features into a unified inference procedure, which can be efficiently performed by solving a convex optimization problem. The proposed method seamlessly integrates multiple types of features to collaboratively produce the affinity matrix within a single inference step, and thus produces more accurate and reliable results.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

In this dissertation, we explored sparse modeling for various tasks in computer vision and machine learning to address their specific challenges, which are summarized as follows:

1) **Graph Learning:** We proposed the concept of ℓ^1-graph, encoding the overall behavior of the data set in sparse representations. The ℓ^1-graph is robust to data noises and naturally sparse, and offers adaptive neighborhood for individual datum. It is also empirically observed that the ℓ^1-graph conveys greater discriminating power compared with classical graphs constructed by k-nearest-neighbor or ϵ-ball method. All these characteristics make it a better choice for many popular graph-oriented machine learning tasks.

2) **Misalignment-Robust Face Recognition:** We developed the SSPC, supervised sparse patch coding, framework towards a robust solution to the challenging face

recognition task with considerable spatial misalignments and possible image occlusions. In this framework, each image is represented as a set of local patches, and the classification of a probe image is achieved with the collective sparse reconstructions of the patches of the probe image from the patches of all the gallery images with the consideration of both spatial misalignments and the extra sparse enforcement on subject confidences. SSPC naturally integrates the patch-based representation, supervised learning and sparse coding, and thus is superior to most conventional algorithms in term of algorithmic robustness.

3) **Label to Region:** We proposed a novel sparse coding technique for addressing an interesting task of label-to-region assignment, which only requires image-level label annotations. With the popularity of the photo sharing websites, the community-contributed images with rich tag information are becoming much easier to obtain, it is predicated that the keyword query based semantic image search can greatly benefit from applying our proposed technique for label-to-region assignment on these tagged images.

4) **Image Segmentation:** This work presented a novel image segmentation framework called multi-task low-rank affinity pursuit (MLAP). In contrast with existing single-feature based methods, MLAP integrates the information of multiple types of features into a unified inference procedure, which can be efficiently performed by solving a convex optimization problem. The proposed method seamlessly integrates multiple types of features to collaboratively produce the affinity matrix within a single inference step, and thus produces more accurate and reliable results.

6.2 Future Works

During the research, we found that sparse analysis is a very powerful tool for statistical tasks. However, there still exist some limitations which impede its wide applications. For example, the computation cost for sparsity analysis is very high, *e.g.*, about 337 seconds for the 2414 samples in the YALE-B database on a PC with 3Ghz CPU and 2GB memory in ℓ^1-graph construction. And the computational cost for robust PCA is even higher. Thus, how to improve the computational efficiency is very important for large scale applications. So in the future, we are planning to further study the sparse analysis from two aspects:

1) **Acceleration for Sparse Analysis on Scalable Dataset**: The optimization procedure is the main cost of solving sparse analysis problem. Most of the previous methods considered the optimization problem as a whole. If the dataset is scalable, the computation cost will be higher and higher. In our opinion, if we can split the main optimization problem into many small subproblems, the cost will be exponentially cut down. This is a very interesting direction for furthur investigation.

2) **Sparsity Analysis for Video Content Analysis**: Video is an organic combination of images. It contains much information not only in the single frames but also in the connections of these frames. In our presented works, we applied the sparsity analysis for images. Due to the high computational cost, we didnot extend it to video analysis. After the acceleration for sparse analysis on scalable dataset, this will be the new direction for us.

List of Publications

1) Bin Cheng, Jianchao Yang, Shuicheng Yan, Yun Fu, Thomas S. Huang: Learning with $\ell 1$-graph for image analysis. IEEE Transactions on Image Processing, Vol. 19, No. 4, pp. 858-866, 2010.

2) Bin Cheng, Bingbing Ni, Shuicheng Yan, Qi Tian: Learning to photograph. ACM Multimedia 2010: 291-300.

3) Bin Cheng, Guangcan Liu, Jingdong Wang, ZhongYang Huang, Shuicheng Yan: Multi-task low-rank affinity pursuit for image segmentation. International Conference on Computer Vision 2011: 2439-2446.

4) Xiaobai Liu, Bin Cheng, Shuicheng Yan, Jinhui Tang, Tat-Seng Chua, Hai Jin: Label to region by bi-layer sparsity priors. ACM Multimedia 2009: 115-124.

5) Congyan Lang, Bin Cheng, Songhe Feng, Xiaotong Yuan: Supervised sparse patch coding towards misalignment-robust face recognition. Journal of Visual Communication and Image Representation 2012.

Bibliography

[1] Hull, J.: A database for handwritten text recognition research. IEEE Transactions on Pattern Analysis and Machine Intelligence **16** (1994) 550–554

[2] : (http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html)

[3] Rao, S., Mobahi, H., Yang, A.Y., Sastry, S., Ma, Y.: Natural image segmentation with adaptive texture and boundary encoding. In: ACCV. (2009) 135–146

[4] Donoho, D.: For most large underdetermined systems of linear equations the minimal ℓ^1-norm solution is also the sparsest solution. Communications on Pure and Applied Mathematics **59** (2004) 797–829

[5] Wright, J., Ganesh, A., Yang, A., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **31** (2009) 210–227

[6] Chen, S., D.Donoho, Saunders, M.: Atomic decomposition by basis pursuit. Society for Industrial and Applied Mathematics Review **43** (2001) 129–159

[7] Wagner, A., Wright, J., Ganesh, A., Zhou, Z., Ma, Y.: Towards a practical face recognition system: Robust registration and illumination by sparse representation.

In: IEEE Conference on Computer Vision and Pattern Recognition. (2009) 597–604

[8] Yang, A., Jafari, R., Sastry, S., Bajcsy, R.: Distributed recognition of human actions using wearable motion sensor networks. Ambient Intelligence and Smart Environments **1** (2009) 103–115

[9] Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Supervised dictionary learning. In: Advances in Neural Information Processing Systems. (2008) 1033–1040

[10] Bradley, D.M., Bagnell, J.A.: Differential sparse coding. In: Advances in Neural Information Processing Systems. (2008) 113–120

[11] Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition. (2009) 1794–1801

[12] Cevher, V., Sankaranarayanan, A.C., Duarte, M.F., Reddy, D., Baraniuk, R.G., Chellappa, R.: Compressive sensing for background subtraction. In: European Conference on Computer Vision. (2008) 155–168

[13] Hang, X., F.Wu: sparse representation for classification of tumor using gene expression data. Journal of Biomedicine and Biotechnology (2009)

[14] Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 888–905

[15] Tenenbaum, J., Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science **290** (2000) 2319–2323

[16] Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290** (2000) 2323–2326

[17] Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation **15** (2002) 1373–1396

[18] Joliffe, I.: Principal component analysis. Springer-Verlag (1986) 1580–1584

[19] Belhumeur, P., Hespanda, J., Kiregeman, D.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 711–720

[20] He, X., Niyogi, P.: Locality preserving projections. In: Advances in Neural Information Processing Systems. Volume 16. (2003) 585–591

[21] Yan, S., Xu, D., Zhang, B., Yang, Q., Zhang, H., Lin, S.: Graph embedding and extensions: A general framework for dimensionality reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence **29** (2007) 40–51

[22] Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: International Conference on Machine Learning. (2003) 912–919

[23] Belkin, M., Matveeva, I., Niyogi, P.: Regularization and semi-supervised learning on large graphs. In: International Conference on Learning Theory. Volume 3120. (2004) 624–638

[24] Meinshausen, N., B$\ddot{u}$hlmann, P.: High-dimensional graphs and variable selection with the lasso. Annals of Statistics **34** (2006) 1436–1462

[25] Olshausen, B., Field, D.: Sparse coding with an overcomplete basis set: A strategy employed by v1? Vision Research **37** (1998) 3311–3325

[26] : (http://sparselab.stanford.edu)

[27] Lee, K., Ho, J., Kriegman, D.: Acquiring linear subspaces for face recognition under variable lighting. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 684–698

[28] He, X., Cai, D., Yan, S., Zhang, H.: Neighborhood preserving embedding. In: IEEE International Conference on Computer Vision. Volume 2. (2005) 1208–1213

[29] Cai, D., He, X., Han, J.: Semi-supervised discriminant analysis. In: IEEE International Conference on Computer Vision. (2007) 1–7

[30] : (http://kdd.ics.uci.edu/databases/covertype/covertype.data.html/)

[31] Zheng, X., Cai, D., He, X., Ma, W., Lin, X.: Locality preserving clustering for image database. In: ACM International Conference on Multimedia. (2004) 885–891

[32] Li, X., Lin, S., Yan, S., Xu, D.: Discriminant locally linear embedding with high order tensor data. IEEE Transaction on Systems, Man, and Cybernetics, Part B: Cybernetics **38** (2008) 342–352

[33] Pang, Y., Tao, D., Yuan, Y., Li, X.: Binary two-dimensional pca. IEEE Transaction on Systems, Man, and Cybernetics, Part B: Cybernetics **38** (2008) 1176–1180

[34] Pang, Y., Yuan, Y., Li, X.: Gabor-based region covariance matrices for face recognition. IEEE Transaction on Circuits System and Video Technology **18** (2008) 989–993

[35] Shan, S., Chang, Y., Gao, W., Cao, B., Yang, P.: Curse of mis-alignment in face recognition: Problem and a novel mis-alignment learning solution. In: IEEE In-

ternational Conference on Automatic Face and Gesture Recognition. (2004) 314–320

[36] Yang, J., Yan, S., Huang, T.: Ubiquitously supervised subspace learning. IEEE Transaction on Image Processing **18** (2009) 241–249

[37] Xu, D., Yan, S., Luo, J.: Face recognition using spatially constrained earth movers distance. IEEE Transactions on Image Processing **17** (2008) 2256–2260

[38] Wang, H., Yan, S., Huang, T., Liu, J., Tang, X.: Misalignment-robust face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. (2008) 1–6

[39] Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (2001) 681–685

[40] Wang, P., Green, M., Ji, Q., Wayman, J.: Automatic eye detection and its validation. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 3. (2005) 164–171

[41] : (http://cvc.yale.edu/projects/yalefaces/yalefaces.html)

[42] Tenenbaum, J., Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 684–698

[43] Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (2003) 264–271

[44] Crandall, D.J., Huttenlocher, D.P.: Weakly supervised learning of part-based spatial models for visual object recognition. In: European Conference on Computer Vision. (2006) 16–29

[45] Winn, J., Jojic, N.: Locus: Learning object classes with unsupervised segmentation. In: IEEE International Conference on Computer Vision. Volume 1. (2005) 756–763

[46] Cao, L., Li, F.: Spatially coherent latent topic model for concurrent object segmentation and classification. In: IEEE 11th International Conference on Computer Vision. (2007) 1–8

[47] Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: ECCV workshop on statistical learning in computer vision. (2004) 17–32

[48] Chen, Y.: Unsupervised learning of probabilistic object models (poms) for object classification, segmentation and recognition. In: the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2008)

[49] Szummer, M., Picard, R.: Indoor-outdoor image classification. In: IEEE International Workshop on Content-Based Access to Image and Video Databases. (1998) 42–51

[50] Vailaya, A., Jain, A., Zhang, H.: On image classification: City vs. landscape. IEEE Workshop on Content-Based Access of Image and Video Libraries (1998) 3–8

[51] Haering, N., Myles, Z., Lobo, N.: Locating dedicuous trees. In: Proc. IEEE Workshop on Contentbased Access of Image and Video Libraries. (1997) 18–25

[52] Forsyth, D., Fleck, M.: Body plans. In: IEEE Conference on Computer Vision and Pattern Recognition. (1997) 678–683

[53] Li, Y., Shapiro, L.: Consistent line clusters for building recognition in cbir. Volume 3. (2002) 952–956

[54] Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: SIGIR Forum. (2003) 119–126

[55] Lavrenko, V., Manmatha, R., Jeon, J.: A model for learning the semantics of pictures. In: Neural Information Processing Systems. (2004) 553–560

[56] Feng, S., Manmatha, R., Lavrenko, V.: Multiple bernoulli relevance models for image and video annotation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (2004) 1002–1009

[57] Liu, J., Wang, B., Li, M., Li, Z., Ma, W., Lu, H., Ma, S.: Dual cross-media relevance model for image annotation. In: ACM International Conference on Multimedia. (2007) 605–614

[58] Kang, F., Jin, R., Sukthankar, R.: Correlated label propagation with application to multi-label learning. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2006) 1719–1726

[59] Zhang, M., Zhou, Z.: Ml-knn: A lazy learning approach to multi-label learning. Pattern Recognition **40** (2007) 2038–2048

[60] Jin, R., Chai, J.Y., Si, L.: Effective automatic image annotation via a coherent language model and active learning. In: Proceedings of the 12th annual ACM International Conference on Multimedia. (2004) 892–899

[61] Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. International Journal of Computer Vision **59** (2004) 167–181

[62] Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110

[63] Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for mulit-class object recognition and segmentation. In: European Conference on Computer Vision. (2006) 1–15

[64] Yuan, J., Li, J., Zhang, B.: Exploiting spatial context constraints for automatic image region annotation. In: ACM International Conference on Multimedia. (2007)

[65] Chua, T., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: A real-world web image database from national university of singapore. In: ACM International Conference on Image and Video Retrieval. (2009)

[66] Fan, R., Chen, P., Lin, C.: Working set selection using the second order information for training svm. In: Journal of Machine Learning Research. Volume 6. (2005) 1889–1918

[67] Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multilabel scene classification. Pattern Recognition **37** (2004) 1757–1771

[68] Elisseef, A., Weston, J.: A kernel method for multi-labelled classification. In: Advances in Neural Information Processing Systems. Volume 14. (2001) 681–687

[69] Comite, F., Gilleron, R., Tommasi., M.: Learning multi-label altenating decision tree from texts and data. In: Machine Learning and Data Mining in Pattern Recognition. (2003) 251–274

[70] Wertheimer, M.: Laws of organization in perceptual forms. A Sourcebook of Gestalt Psycychology (1938)

[71] Liu, G., Lin, Z., Tang, X., Yu, Y.: Unsupervised object segmentation with a hybrid graph model (HGM). TPAMI (2010)

[72] Pantofaru, C., Schmid, C., Hebert, M.: Object recognition by integrating multiple image segmentations. In: ECCV. (2008)

[73] Arbelaez, P.: Boundary extraction in natural images using ultrametric contour maps. In: CVPR Workshop. (2006)

[74] Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. TPAMI, to appear (2010)

[75] Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. TPAMI (2002)

[76] Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: CVPR. (2005)

[77] Geman, S., Geman, D.: Readings in computer vision: issues, problems, principles, and paradigms. (1987)

[78] Schoenemann, T., Kahl, F., Cremers, D.: Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation. In: ICCV. (2009)

[79] Tu, Z., Zhu, S.C.: Image segmentation by data-driven markov chain monte carlo. TPAMI (2002)

[80] Wang, J., Jia, Y., Hua, X.S., Zhang, C., Quan, L.: Normalized tree partitioning for image segmentation. In: CVPR. (2008)

[81] Malik, J., Belongie, S., Shi, J., Leung, T.: Textons, contours and regions: Cue integration in image segmentation. In: ICCV. (1999)

[82] Freixenet, J., Muñoz, X., Raba, D., Martí, J., Cufí, X.: Yet another survey on image segmentation: Region and boundary information integration. In: ECCV. (2002)

[83] Delaunoy, A., Fundana, K., Prados, E., Heyden, A.: Convex multi-region segmentation on manifolds. In: ICCV. (2009)

[84] Ma, Y., Derksen, H., Hong, W., Wright, J.: Segmentation of multivariate mixed data via lossy data coding and compression. TPAMI (2007)

[85] Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR. (2008)

[86] M., P., T., O.: Texture analysis in industrial applications. Image Technology (1996)

[87] Lowe, D.: Object recognition from local scale-invariant features. In: ICCV. (1999)

[88] Zhou, D., Burges, C.: Spectral clustering and transductive learning with multiple views. In: ICML. (2007)

[89] Mori, G., Ren, X., Efros, A., Malik, J.: Recovering human body configurations: combining segmentation and recognition. In: CVPR. (2004)

[90] Lin, Z., Chen, M., Wu, L., Ma, Y.: The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report, UILU-ENG-09-2215 (2009)

[91] Ma, Y., Yang, A., Derksen, H., Fossum, R.: Estimation of subspace arrangements with applications in modeling and segmenting mixed data. SIAM Review (2008)

[92] Cheng, B., Yang, J., Yan, S., Fu, Y., Huang, T.: Learning with ℓ^1-graph for image analysis. TIP (2010)

[93] Liu, G., Lin, Z., Yu, Y.: Robust subspace segmentation by low-rank representation. In: ICML. (2010)

[94] Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., Ma, Y.: Robust recovery of subspace structures by low-rank representation. Preprint (2010)

[95] Cai, J.F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization (2010)

[96] Zhang, Y.: Recent advances in alternating direction methods: Practice and theory. Tutorial (2010)

[97] Malisiewicz, T., Efros, A.: Improving spatial support for objects via multiple segmentations. In: BMVC. (2007)

[98] Ren, X., Fowlkes, C., Malik, C.: Scale-invariant contour completion using condition random fields. In: ICCV. (2005)

[99] Meila, M.: Comparing clustering: An axiomatic view. In: ICML. (2005)

[100] Rand, W.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association (1971)

[101] Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contour to regions:an empirical evaluation. In: CVPR. (2009)