Structured Sparse Learning and Its Applications to Biomedical and Biological Data

by

Lei Yuan

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2013 by the
Graduate Supervisory Committee:

Jieping Ye, Chair
Yalin Wang
Guoliang Xue
Sudhir Kumar

ARIZONA STATE UNIVERSITY

May 2013

ABSTRACT

Sparsity has become an important modeling tool in areas such as genetics, signal and audio processing, medical image processing, etc. Via the penalization of ℓ_1 norm based regularization, the structured sparse learning algorithms can produce highly accurate models while imposing various predefined structures on the data, such as feature groups or graphs. In this thesis, I first propose to solve a sparse learning model with a general group structure, where the predefined groups may overlap with each other. Then, I present three real world applications which can benefit from the group structured sparse learning technique. In the first application, I study the Alzheimer's Disease diagnosis problem using multi-modality neuroimaging data. In this dataset, not every subject has all data sources available, exhibiting an unique and challenging block-wise missing pattern. In the second application, I study the automatic annotation and retrieval of fruit-fly gene expression pattern images. Combined with the spatial information, sparse learning techniques can be used to construct effective representation of the expression images. In the third application, I present a new computational approach to annotate developmental stage for Drosophila embryos in the gene expression images. In addition, it provides a stage score that enables one to more finely annotate each embryo so that they are divided into early and late periods of development within standard stage demarcations. Stage scores help us to illuminate global gene activities and changes much better, and more refined stage annotations improve our ability to better interpret results when expression pattern matches are discovered between genes.

DEDICATION

I lovingly dedicate this thesis to my wife, Yao Wang, for her love, care and understanding every step of the way. I also dedicate this thesis to my parents, my mother-in-law and father-in-law, for their never ending support.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Dr. Jieping Ye, for his guidance, encouragement, and support during my dissertation research. He is an outstanding mentor, an easygoing friend, and the most dedicated researcher I have ever known. The experiences with him are my lifelong assets. It has also been a very rewarding and pleasant experience to work in the Center for Evolutionary Medicine and Informatics, and I would like to thank Dr. Sudhir Kumar for his guidance, support, and feedback. I would like to thank my dissertation committee members, Dr. Yalin Wang and Dr. Guoliang Xue, for their valuable interactions and feedback.

I have been very fortunate to work on the FlyExpress project, where I experience how fundamental research in computer science can help make scientific discoveries possible in the field of developmental biology. I would like to thank Dr. Timothy Karr, Dr. Charlotte Konikoff, and Dr. Stuart Newfeld for providing me with biological knowledge, Michael McCutchan for providing seamless access to the biological data, Omy Keyes, Carol Williams, and Jason Wulf for providing administrative, editorial, and IT support.

Members of our Machine Learning Lab inspired me a lot through discussions, seminars, and project collaborations, and I would like to thank the following people for their valuable interactions: Dr. Shuiwang Ji, Dr. Jianhui Chen, Dr. Liang Sun, Ji Liu, Dr. Jun Liu, Alexander Woodard, Jiayu Zhou, Shuo Xiang, Qian Sun, Rita Chattopadhyay, Sen Yang.

TABLE OF CONTENTS

Page

LIST OF TABLES vii

LIST OF FIGURES ix

CHAPTER

1 Background and Introduction 1

1.1 Structured Sparse Learning and Overlapping Group Lasso 1

1.2 Multi-Source Learning with Incomplete Data in Alzheimer's Disease Study 2

1.3 Learning Sparse Representations for Fruit-Fly Gene Expression Pattern Image Annotation and Retrieval 6

1.4 Automated Annotation of Developmental Stages of *Drosophila* Embryos in Images Containing Spatial Patterns of Expression 9

2 Overlapping Group Lasso 11

2.1 The Associated Proximal Operator and Its Efficient Computation 12

Key Properties of the Proximal Operator 13

Reformulation as an Equivalent Smooth Convex Optimization Problem . . . 16

Computing the Duality Gap 17

2.2 ℓ_q Norm Overlapping Group Lasso 18

Properties of the ℓ_q Proximal Operator 19

Extending the Dual Method to the ℓ_q Case 20

2.3 Overlapping Group Lasso via the Capped Norm 21

2.4 Experiments 22

Synthetic Data 23

Efficiency of Calculating the Proximal Operator 23

Sparse Pattern Recovery 24

Gene Expression Data 27

CHAPTER Page

Comparison with SLasso, Prox-Grad and ADMM 28

Comparison with Picard-Nesterov 29

Computation of the Proximal Operator 30

Convergence with Inexact Proximal Operator 31

Additional Discussions on Overlapping Group Lasso 32

3 Multi-Source Learning with Incomplete Data 34

3.1 Proposed Method I: Incomplete Multi-Source Feature Learning 35

3.2 Proposed Method II: Model Score Completion 37

3.3 Comparison of iMSF, ScoreComp and imputation methods 39

4 Learning Sparse Representations for Fruit-Fly Gene Expression Pattern Image Annotation and Retrieval 45

4.1 The Bag-of-Words Approach 45

Feature Detection 46

Feature Description 46

Codebook Generation 46

4.2 The Spatial Bag-of-Words Approach 47

4.3 The Sparse Spatial Representation 48

4.4 Data Description 50

4.5 Evaluation of Developmental Term Annotation Performance 51

4.6 Evaluation of Retrieval Performance 55

5 Automated Annotation of Developmental Stages of *Drosophila* Embryos in Images Containing Spatial Patterns of Expression 58

5.1 Results and Discussion 58

Improving Similar Expression Pattern Retrieval 60

Genomewide-Expression-Maps with refined stage information 61

CHAPTER Page

5.2 Materials and Methods 62
Training Set Acquisition 66
Model Pool Construction 67
Feature Construction 68
Preliminary on Linear Classifiers 68
Exploiting the underlying sparse structure 70
Voting for Stage Annotation and beyond 73
Stage Annotation by Majority Voting 73
Embryogenesis as a Continuous Process 74
Sub-stage Annotation and Decimal Based Embryo Ordering 75
With or Without Stage Range Information 76
More on Model Ensemble 77
6 Conclusion and Outlook 81
6.1 Summary of Contributions 81
6.2 Future Directions 83
REFERENCES 84

LIST OF TABLES

Table Page

2.1 Cross-validation performance of sparse pattern recovery of the convex overlapping group Lasso formulation and the non-convex overlapping group Lasso formulation based on the capped norm on synthetic data with different problem sizes. 27

2.2 Scalability study of the FoGLasso algorithm under different numbers (p) of genes involved. The reported results are the total computational time (seconds) for all nine regularization parameter values. 30

2.3 Comparison of FoGLasso, Picard-Nesterov and Picard-Nesterov with the proposed pre-processing technique using different numbers (p) of genes and various precision levels. For each particular method, the first row denotes the number of outer iterations required for convergence, while the second row represents the total number of inner iterations. 30

3.1 Classification performance comparison of the proposed iMSF, ScoreComp and missing value estimation methods (Zero, EM, KNN, SVD and SVT) in terms of accuracy, sensitivity and specificity when the training percentage varies from 50% to 75%. 41

4.1 Comparison of different annotation methods in terms of AUC. 54

4.2 Comparison of different annotation methods in terms of macro F1. 54

5.1 Number of Annotated BDGP Images for each developmental stage. This collection of images are manually annotated with precise stage labels. The orientation of all images in this study is standardized, and the size is scaled to 128 by 320 pixels. 59

Table Page

5.2 Average classification accuracy on the validation set for different algorithms and different training ratio, when stage range information is not available. 7 algorithms are evaluated, including SVM with linear kernel (SVM) and 6 sparse learning methods: Lasso (lasso), group Lasso (glasso), sparse group Lasso (sglasso) with least square loss (least) and logistic loss (log). 72

5.3 Average rate of disagreement among all 7 different algorithms for different training ratios. 73

5.4 Average classification accuracy on the validation set for different algorithms and different training ratios, when stage range information is available. 7 algorithms are evaluated, including SVM with linear kernel (SVM) and 6 sparse learning methods: Lasso (lasso), group Lasso (glasso), sparse group Lasso (sglasso) with least square loss (least) and logistic loss (log). 76

5.5 Average classification accuracy within each stage range for different training ratios. SVM with linear kernel is used as the classifier and stage range information is available. 77

5.6 Performance evaluation of model ensemble using different subsets of the learning algorithms. Three evaluation criteria are used, namely, the sub-stage accuracy ($\text{Acc}_{0.5}$), the stage accuracy ($\text{Acc}_{\text{Stage}}$) and the plus-minus-half accuracy ($\text{Acc}_{\pm 0.5}$). Both with and without range information cases are considered. . . . 78

5.7 Performance evaluation of model ensemble using different subsets of the trained models. In the "Random Subset" category, a number of models are randomly selected from the trained models to form the model ensemble. In the "Ranked by Validation Accuracy" category, the trained models are first ranked by their accuracy on the corresponding validation set such that the top models can be selected. Only the case without stage range information is considered. 80

LIST OF FIGURES

Figure Page

1.1 Illustration of integrating multiple heterogeneous data sources for disease status prediction tasks. 4

1.2 Sample images from the six stage ranges from BDGP. It has the largest collection of images for early as well as late stages. The images in BDGP are grouped into six stage ranges: 1-3, 4-6, 7-8, 9-10, 11-12 and 13-17. 10

2.1 Time comparison for computing the proximal operators. The group number is fixed in the left figure and the problem size is fixed in the middle figure. The right figure illustrates the effectiveness of the pre-processing. 24

2.2 Performance of sparse pattern recovery of the convex overlapping group Lasso formulation (2.1) (top row) and the non-convex overlapping group Lasso formulation with the capped norm (2.31) (bottom row) on synthetic data with different problem sizes. 26

2.3 Comparison of SLasso [1], ADMM [2], Prox-Grad [3] and the proposed FoGLasso algorithm in terms of computational time (in seconds and in the logarithmic scale) when different numbers of genes (variables) are involved. Different precision levels are used for comparison. 28

2.4 Performance of the computation of the proximal operator in FoGLasso. The left plot shows the objective function value during the FoGLasso iteration. The middle plot shows the percentage of the identified zero groups by applying Lemma 3. The right plot shows the number of inner iterations for achieving the duality gap less than 10^{-10} when one solves the proximal operator via the dual reformulation (see Section 3.2). 31

2.5 Illustration of the objective function values of the first 50 iterations with different stopping criteria used for computing the proximal operator. 32

Figure Page

3.1 Illustration of the "block-wise" pattern of missing data for the ADNI data set. In this figure, I show AD and normal control subjects only. For simplicity, I focus on those subjects with complete MRI measures. 34

3.2 Illustration of the proposed multi-task feature learning framework for incomplete multi-source data fusion. In the proposed framework, first I partition the samples into multiple blocks (four blocks in this case), one for each combination of data sources available: (1) PET, MRI; (2) PET, MRI, CSF; (3) MRI, CSF; (4) MRI. Then I build four models, one for each block of data, resulting in four prediction tasks. I use a joint feature learning framework that learns all models simultaneously. Specifically, all models involving a specific source are constrained to select a common set of features for that particular source. 35

3.3 Illustration of the proposed model score completion scheme. First I train a base model on each individual data source using the available samples, and the base model is applied to produce prediction scores for this data source; thus each data source is represented by a single column of (incomplete) scores. A missing value estimation method is applied to obtain a complete set of model scores, which are treated as newly derived features to train our final classifier. . 38

4.1 Illustration of image patch extraction and the three levels of bag-of-words partitioning with weighting factors for the spatial pyramid approach. After feature description using the overlapping circular patches, three levels of bag-of-words partitioning are shown. The top level of partitioning is just a global bag-of-words representation. 49

Figure Page

4.2 Different histogram representation obtained for a given image. The histogram on the left is obtained by assigning each local patch to a single visual word, while the one on the right is obtained by applying the sparse learning formula to select a set of visual words for each patch. 51

4.3 Retrieval results for query image ID insitu16633 with the lateral view in stage range 9-10. 56

4.4 Retrieval results for query image ID insitu23837 with the lateral view in stage range 13-16. 57

5.1 Overview of my stage annotation system. By learning from a training dataset with manually labeled stage information, I build a pool of 1050 classification models. I then apply this pool to the unlabeled images in our FlyExpress database, providing a histogram of voting values for each image. The histogram is then used to annotate the image with a specific stage, as well as a more refined ”sub-stage” and numerical based ”stage score”. 59

5.2 Examples of refining image retrieval results using stage annotation. Two example query images are used, with the left one (A) from the Kr gene and right one (B) from Gasp. The top matches from the FlyExpress lateral BDGP images are listed, with corresponding pattern as well as similarity values. The annotated stage from my system is presented on the left side of each expression image. . . 61

5.3 Stages 7 - 12 GEMs generated by using only the stage range information (A, left column), the predicted stage information (B, middle column) and the sub-stage information (C, right column). The total number of images involved for creating each individual GEM is also reported. 63

5.4 Stage 4 - 6 GEMs generated by using only the stage range information (left column), the predicted stage information (middle column) and the sub-stage information (right column). The total number of images involved for creating each individual GEM is also reported. 64

5.5 Stage 13 - 16 GEMs generated by using only the stage range information (left column), the predicted stage information (middle column) and the sub-stage information (right column). The total number of images involved for creating each individual GEM is also reported. 64

5.6 Genomewide-Expression-Maps (GEMs) obtained from automatically annotated lateral BDGP images (stages 7-12) overlaid with previously published overview images of stage development. 65

5.7 Lateral BDGP images from gene twi, sorted by decimal-stage annotations. The annotated stages as well as decimal-stage scores are reported on the top of each expression image. 65

5.8 Lateral BDGP images from gene tkv, sorted by decimal-stage annotations. The annotated stages as well as decimal-stage scores are reported on the top of each expression image. 66

5.9 Lateral BDGP images from gene gt, sorted by decimal-stage annotations. The annotated stages as well as decimal-stage scores are reported on the top of each expression image. 67

5.10 Illustration of the feature extraction process. The standardized image is first processed by a series of log-Gabor filters, resulting in 24 Gabor images. These Gabor images are then down sampled and concatenated into a single feature vector, which is the final representation of the original image. As indicated by the red cross in the figure, one sub-block of the original image will correspond to 24 features in the feature vector, one for each Gabor image. 69

5.11 Summary of total voting for each stage in the labeled data set. For each stage in the training dataset, I summarize the predicted stages from all the models on all the images to obtain a voting distribution for that stage. The voting distribution will always peak at the corresponding stage. 74

Chapter 1

Background and Introduction

1.1 Structured Sparse Learning and Overlapping Group Lasso

Problems with high dimensionality have become common over the recent years. The high dimensionality poses significant challenges in building interpretable models with high prediction accuracy. Regularization has been commonly employed to obtain more stable and interpretable models. A well-known example is the penalization of the ℓ_1 norm of the estimator, known as Lasso [4]. The ℓ_1 norm regularization has achieved great success in many applications. Although Lasso can simultaneously select a small number of relevant variables when building a highly accurate model, it lacks the ability to incorporate any structural information among the input variables. Recently, as extensions to Lasso, various structured sparse learning models have been proposed to take advantage of the prior structural knowledge of the input features. The prior knowledge can take various forms, such as non-overlapping groups over variables [5], a chain-ordering [6, 7], a tree structure [8] or a general graph [9, 10].

For group feature selection, group Lasso based on the combination of the ℓ_1 norm and the ℓ_2 norm has been proposed, and quite a few efficient algorithms [11–13] have been proposed for efficient optimization. However, the non-overlapping group structure in group Lasso limits its applicability in practice. For example, in microarray gene expression data analysis, genes may form overlapping groups as each gene may participate in multiple pathways [14]. Several recent work [14–18] studies the overlapping group Lasso, where groups of features are given, potentially with overlaps between the groups. By allowing arbitrary group overlap in this formulation, many existing sparse learning formulations can be considered as its special cases, including Lasso, group Lasso, tree structured Lasso, as well as the ones used in this thesis. I develop an efficient algorithm for the overlapping group Lasso penalized problem via the accelerated gradient

descent method. The accelerated gradient descent method has recently received increasing attention in machine learning due to the fast convergence rate even for non-smooth convex problems. One of the key operations is the computation of the proximal operator associated with the penalty. I reveal several key properties of the proximal operator associated with the overlapping group Lasso penalty, and proposed a dual reformulation that can be solved efficiently.

In addition, I propose two extensions to the proposed algorithm. Firstly, we generalize our method and theoretical results to solve the general overlapping group Lasso with the ℓ_q norm with $q > 1$ (when $q = 1$, there is no grouping effect). I then tackle a non-convex overlapping group Lasso formulation based on the capped norm regularization. I propose to decompose the non-convex capped norm penalty to the difference of two convex functions, and solve the equivalent problem using DC programming. The sub-problem of each DC step is equivalent to the original overlapping group Lasso problem.

I have performed empirical evaluations using both synthetic data and the breast cancer gene expression data set, which consists of 8,141 genes organized into (overlapping) gene sets. Experimental results demonstrate the efficiency of the proposed algorithm in comparison with existing state-of-the-art algorithms. Results also demonstrate the effectiveness of the non-convex overlapping group Lasso formulation.

1.2 Multi-Source Learning with Incomplete Data in Alzheimer's Disease Study

Alzheimer's disease (AD) is a highly prevalent neurodegenerative disease, and is widely recognized as a major, escalating epidemic and a world-wide challenge to global health care systems [19]. AD is the the most common type of dementia, accounting for 60-80% of age-related dementia cases. The direct cost of care for AD patients by family members and healthcare professionals is more than $100 billion per year; this figure is expected to rise dramatically as the population ages over the next several decades [20]. In

AD patients, neurons and their connections are progressively destroyed, leading to loss of cognitive function and ultimately death. The underlying pathology is thought to precede the onset of cognitive symptoms by many years [21, 22]. Efforts are underway to find early diagnostic markers to evaluate AD risk pre-symptomatically in a rapid and rigorous way. Such findings will help establish early interventions to prevent or at least postpone the onset of AD, or reduce the risk of developing the disease.

Neuroimaging is a powerful tool to measure disease progression and therapeutic efficacy in AD and mild cognitive impairment (MCI). Neuroimaging research offers great potential to discover features that can identify individuals early in the course of dementing illness; several candidate neuroimaging biomarkers have been examined in recent cross-sectional and longitudinal neuroimaging studies [23, 24]. Reduced fluorodeoxyglucose (FDG) PET measurements of the cerebral metabolic rate for glucose in brain regions preferentially affected by AD, structural MRI measures of brain shrinkage, and cerebrospinal fluid (CSF) measurements are among the best established biomarkers of AD progression and pathology [20]. Realizing the importance of combining neuroimaging and genetics, NIH in 2003 funded the Alzheimer's Disease Neuroimaging Initiative (ADNI [25, 26], PI: Michael W. Weiner). The initiative is facilitating the scientific evaluation of neuroimaging data including magnetic resonance imaging (MRI), positron emission tomography (PET), other biomarkers, and clinical and neuropsychological assessments for predicting the onset and progression of MCI and AD. By identifying more sensitive and specific markers of very early AD progression, these efforts should make it easier to diagnose AD earlier as well as develop, assess, and monitor new treatments.

Clinical and research studies commonly acquire complementary brain images, neuropsychological and genetic data for each participant for a more accurate and rigorous assessment of the disease status and likelihood of progression. Advances in image analysis

make it possible to use one image modality to support the analysis of a complementary image modality [27–30]. However, only a few systems, e.g., [31–39], applied data mining and machine learning techniques such as the multivariate linear model and partial least squares to characterize the linkage between the patterns of information from the same individual's brain images and other biological measures. Instead, most researchers perform statistical analysis by analyzing different images separately. In general, these "unimodal" analysis could be improved by considering other sources of relevant information from multiple imaging modalities, e.g., PET and MRI, and non-imaging data sets from genomics and proteomics. It is a common belief that by integrating multiple heterogeneous sources (as illustrated in Figure 1.1), one may not only provide more accurate information on AD progression and pathology, but also better predict cognitive decline before the onset of illness, or at least in the earliest stages of disease.

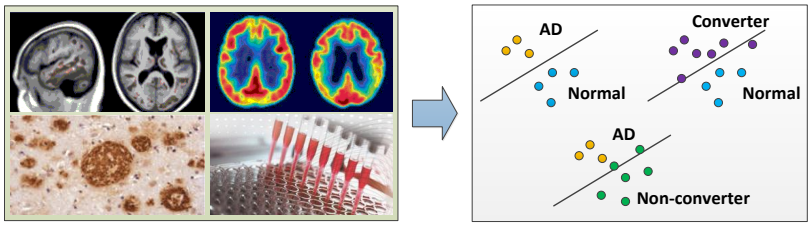


Figure 1.1: Illustration of integrating multiple heterogeneous data sources for disease status prediction tasks.

One common problem that hampers the use of multi-modality imaging approach is the problem of *missing data*. Missing data present a special challenge when integrating large-scale biomedical data. Incomplete data is ubiquitous in real-world biomedical applications. In ADNI, over half of the subjects lack CSF measurements; an independent half of the subjects do not have FDG-PET; many lack proteomics measurements. Missing data may be due to the high cost of certain measures (e.g., PET scans), poor data quality, dropout of the patients from the study, etc. Some measures, such as CSF biomarkers, require more invasive procedures (such as lumbar puncture) which not all study

participants are willing to consent to. Some subjects in a longitudinal study may miss at least one of the regular assessments, or their data quality may be insufficient for accurate analysis at some time points. The simplest approach removes all samples with missing values, but this throws away a vast amount of useful information and dramatically reduces the number of samples in the analysis. As a result, a subject with incomplete data cannot be studied for classification and prognosis. Moreover, with this approach, the resource and time devoted to those subjects with incomplete data are totally wasted. A number of previous works acknowledged the challenge of missing data and discussed general strategies [40–42]. An alternative and popular approach is to estimate missing entries based on the observed values. Many algorithms have been proposed for this [43–46]. While these methods work well when missing data are rare, they are less effective when a significant amount of data is missing, e.g., when all PET features from half of the subjects are missing. Recently, trace norm minimization has been proposed for missing data estimation [47, 48]. This can be effective even when a large amount of data is missing. However, it does assume that the missing locations are random; it is less effective when a complete block of the data is missing, e.g., the complete block of all PET features from half of the subjects. Therefore, computational methods are needed to integrate heterogeneous data with a block-wise missing pattern ("block-wise missing" means a large chunk of data is missing for one or more data sources - an example is shown in Figure 3.1). Without such a method, it is quite challenging to build a highly accurate classifier to process any real multi-modality imaging data sets.

To achieve multi-modality integration while taking into account the block-wise missing nature of the data, I propose two novel learning frameworks. My first system is based on a novel multi-task sparse learning framework, named incomplete multi-source feature learning (iMSF). Based on the availability of different feature sources, I divide the data set into several learning tasks, from each of which a unique classifier is learned. I

then impose a structural sparse learning regularization onto these tasks, such that a common set of features is selected among these tasks. Therefore, I exploit the multi-task nature of the problem and the feature set is learned jointly among different tasks. In my second system, I tackle the difficulty of block-wise data completion by proposing a model score completion scheme (ScoreComp). Each data source is first treated independently, where a base classifier is learned such that the data source is converted into a single column consisting of model prediction scores. Thus, the missing data blocks become single missing values in the new representation of the data. Data imputation techniques are then applied and a new classifier is learned on the completed score matrix such that multiple sources are integrated together.

As an illustrative application, I study clinical group (diagnostic) classification problems in the ADNI baseline data set. Comprehensive experiments demonstrate the promising and stable performance of the proposed systems. 780 subjects in the ADNI baseline data set have their diagnosis (AD, MCI or Normal) available and have at least one type of features available (meaning an image or related clinical measure), including MRI, FDG-PET, CSF and proteomics. For the MCI subjects, I further make use of their 4-year follow up diagnosis to divide them into 2 sub-groups. I label those who had converted to AD by the time of a later visit as "converter" and those who stayed stable as "non-converter". I set out to use these data to tackle clinical group classification problems (AD vs. Normal; AD vs. Non-converter and Converter vs. Normal).

1.3 Learning Sparse Representations for Fruit-Fly Gene Expression Pattern Image Annotation and Retrieval

Embryos undergo a temporally ordered differentiation process, starting as basic undifferentiated eggs. Through the process of differentiation, gene expressions take on increasingly complex patterns. Transcriptional regulation of the fruit-fly *Drosophila melanogaster* is one of the best understood examples of the regulatory networks that

govern gene expression patterning. An understanding of the regulatory networks responsible for gene patterning in *Drosophila* embryos has been aided by digital images produced via *in situ* hybridization [49–51]. These images document the spatiotemporal dynamics of differentiation found in *Drosophila* embryos. A comparative analysis of these images is beneficial for the understanding of functions and interactions in gene networks [52–62]. To facilitate these discoveries, tools have been developed to searching for images based on keywords that describe embryonic structures [63], and searching for images based on gene expression patterns [61, 62]. Images for these tools have been obtained from databases of *Drosophila* embryonic images, e.g. the Berkeley *Drosophila* Genome Project (BDGP), and they are annotated with a controlled vocabulary (CV) [49, 50]. The CV terms describe the developmental and anatomical properties of gene expression during embryogenesis [49]. Currently, groups of BDGP images are manually annotated with CV terms. This is done collectively so that not all images in a group necessarily correspond with each CV annotation. The manual nature of these tasks puts an inordinate burden on biologists as the collection of *Drosophila* gene expression patterns are growing rapidly [49]. It is therefore imperative to investigate efficient and effective computational methods to automate this task [64–66].

Image annotation and image retrieval problems have been studied extensively in computer vision and machine learning. However, natural images are the most common subjects of study for image annotation and image retrieval problems; and commonly-used annotation and retrieval techniques may not be effective for our task. For example, unlike most natural images, BDGP images have all been aligned and scaled semi-automatically. The binary feature vector (BFV) representation has been developed to correlate pattern similarities between images [61], however the BFV representation is not robust to distortions; there were also some studies which tried to use robust descriptors to represent the BDGP images [67–70], however they have not exploited spatial information. It is

desirable to represent images in a way that takes advantage of the spatial properties of image features, while at the same time being robust to image distortions. In our annotation problem, we are interested in collectively annotating groups of images, with each group annotated with multiple CV terms. Previous studies have revealed that ignoring group memberships can be detrimental to annotation performance [67], and formulating the task as learning the function between local input patterns and CV terms leads to significant performance improvement [69].

In this thesis I propose a novel approach for the automated annotation and retrieval of *Drosophila melanogaster* images. I present an image representation model that takes advantage of the spatial information provided by the BDGP images while at the same time being more robust against distortions. I also take advantage of a state-of-the-art learning model in order to boost the performance of our tasks. My feature representation framework is inspired by the spatial bag-of-words (BoW) approach for image representation. The BoW approach involves first extracting features from local patches on images. These patches are then quantized to a visual word that has been determined by a pre-computed codebook. My approach involves extracting these local patches from each image in a group, while maintaining a record of the locations where features are extracted. Thus, my bag-of-words method is essentially a spatial-bag-of-words method. As previous experiments have discovered [64], using only one codebook word to describe a local patch does not capture the slight differences between a word and the actual feature. Therefore, I have adopted a sparse learning framework in order to take advantage of multiple codebook words that show varying levels of similarity to a single feature, leading to a "visual sentence" representation of the image patch.

I have tested our methods on BDGP images from the FlyExpress database (www.flyexpress.net). Annotation results from our study show that the spatial-bag-of-words approach consistently outperforms the non-spatial, bag-of-words

approach as well as the binary feature vector approach. Results also show that incorporating the sparse learning framework into our representation model further improves performance. While for the image retrieval task, experiments show that utilizing the sparse representation alone is sufficient.

1.4 Automated Annotation of Developmental Stages of *Drosophila* Embryos in Images Containing Spatial Patterns of Expression

Overall, research efforts into the spatial and temporal characteristics of gene expression patterns of *Drosophila* have been at the leading-edge of scientific investigations into the fundamental principles of animal development [63, 71–74]. The comparative analysis of gene expression patterns is most biologically meaningful when images from a similar time point are compared [75]. Based on morphological landmarks, the continuous process of Drosophila embryogenesis is traditionally divided into a series of stages (1, 2, . . ., 17) [61]. However, the standard practice of manually inspecting images containing spatial patterns is a rate limiting step, especially when it has to be done for large number of images produced by high-throughput techniques. In fact, images generated in some high-throughput experiments are currently given stage range assignments (e.g., 4-6) rather than individual stages (Figure 1.2). Since the original developmental stage delineations are based on major morphological events in the fruit fly development, it is, in principle, possible to distinguish embryos in images at the level of individual stages (e.g., [64, 66, 76]). However, previous methods only predict stage ranges and no methods currently exist to provide specific stage annotations for Drosophila embryos. Furthermore, no approach currently exists to annotate developmental stage for an embryo on a continuous numerical basis, which would be more biologically realistic because development is a continuous process that is reflected in the output of the high-throughput experiments. Visually, it is possible to scan a set of embryonic expressions and arrange them into a progression of gene expression, which informs us

about the change and direction of spatial expression over time. This indicates need for a system that has the ability to assign more finely graded stage information that enables one to conduct biological discovery using images with higher resolution of stage similarity. In this thesis, I report one such computational system and show how it enhances visualization and scientific discovery.

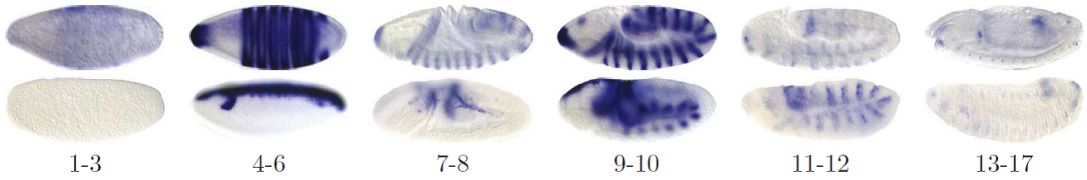


Figure 1.2: Sample images from the six stage ranges from BDGP. It has the largest collection of images for early as well as late stages. The images in BDGP are grouped into six stage ranges: 1-3, 4-6, 7-8, 9-10, 11-12 and 13-17.

Chapter 2

Overlapping Group Lasso

In this chapter, I consider the following overlapping group Lasso penalized problem:

$$\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x}) = l(\mathbf{x}) + \phi_{\lambda_2}^{\lambda_1}(\mathbf{x}) \tag{2.1}$$

where $l(\cdot)$ is a smooth convex loss function, e.g., the least squares loss,

$$\phi_{\lambda_2}^{\lambda_1}(\mathbf{x}) = \lambda_1\|\mathbf{x}\|_1 + \lambda_2 \sum_{i=1}^{g} w_i \|\mathbf{x}_{G_i}\| \tag{2.2}$$

is the overlapping group Lasso penalty, $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are regularization parameters, $w_i > 0, i = 1,2,\ldots,g$, $G_i \subseteq \{1,2,\ldots,p\}$ contains the indices corresponding to the i-th group of features, and $\|\cdot\|$ denotes the Euclidean norm. Note that the first term in (2.2) can be absorbed into the second term, which however will introduce p additional groups. The g groups of features are pre-specified, and they may overlap. The penalty in (2.2) is a special case of the more general Composite Absolute Penalty (CAP) family [18]. When the groups are disjoint with $\lambda_1 = 0$ and $\lambda_2 > 0$, the model in (2.1) reduces to the group Lasso [5]. If $\lambda_1 > 0$ and $\lambda_2 = 0$, then the model in (2.1) reduces to the standard Lasso [4].

In this thesis, I propose to make use of the accelerated gradient descent (AGD) [77–79] for solving (2.1), due to its fast convergence rate. The algorithm is called "FoGLasso", which stands for **F**ast **o**verlapping **G**roup **Lasso**. One of the key steps in the proposed FoGLasso algorithm is the computation of the proximal operator associated with the penalty in (2.2); and I present an efficient algorithm for the computation in the next section.

In FoGLasso, I first construct a model for approximating $f(\cdot)$ at the point $\mathbf{x}$ as:

$$f_{L,\mathbf{x}}(\mathbf{y}) = [l(\mathbf{x}) + \langle l'(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle] + \phi_{\lambda_1}^{\lambda_2}(\mathbf{y}) + \frac{L}{2}\|\mathbf{y}-\mathbf{x}\|^2, \tag{2.3}$$

where $L > 0$. The model $f_{L,\mathbf{x}}(\mathbf{y})$ consists of the first-order Taylor expansion of the smooth function $l(\cdot)$ at the point $\mathbf{x}$, the non-smooth penalty $\phi_{\lambda_1}^{\lambda_2}(\mathbf{x})$, and a regularization term

$\frac{L}{2}\|\mathbf{y}-\mathbf{x}\|^2$. Next, a sequence of approximate solutions $\{\mathbf{x}_i\}$ is computed as follows: $\mathbf{x}_{i+1} = \arg\min_{\mathbf{y}} f_{L_i,\mathbf{s}_i}(\mathbf{y})$, where the search point $\mathbf{s}_i$ is an affine combination of $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ as $\mathbf{s}_i = \mathbf{x}_i + \beta_i(\mathbf{x}_i - \mathbf{x}_{i-1})$, for a properly chosen coefficient β_i, L_i is determined by the line search according to the Armijo-Goldstein rule so that L_i should be appropriate for $\mathbf{s}_i$, i.e., $f(\mathbf{x}_{i+1}) \leq f_{L_i,\mathbf{s}_i}(\mathbf{x}_{i+1})$.

A key building block in FoGLasso is the minimization of (2.3), whose solution is known as the proximal operator [80]. The pseudo-code of FoGLasso is summarized in Algorithm 1, where the proximal operator $\pi(\cdot)$ is defined in (2.4). In practice, I terminate Algorithm 1 if the change of the function values corresponding to adjacent iterations is within a small value, say 10^{-5}.

Algorithm 1 The FoGLasso Algorithm

Input: $L_0 > 0, \mathbf{x}_0, k$

Output: $\mathbf{x}_{k+1}$

1: Initialize $\mathbf{x}_1 = \mathbf{x}_0$, $\alpha_{-1} = 0$, $\alpha_0 = 1$, and $L = L_0$.
2: **for** $i = 1$ to k **do**
3: Set $\beta_i = \frac{\alpha_{i-2}-1}{\alpha_{i-1}}$, $\mathbf{s}_i = \mathbf{x}_i + \beta_i(\mathbf{x}_i - \mathbf{x}_{i-1})$
4: Find the smallest $L = 2^j L_{i-1}, j = 0, 1, \ldots$ such that $f(\mathbf{x}_{i+1}) \leq f_{L,\mathbf{s}_i}(\mathbf{x}_{i+1})$ holds, where $\mathbf{x}_{i+1} = \pi^{\lambda_1/L}_{\lambda_2/L}(\mathbf{s}_i - \frac{1}{L}l'(\mathbf{s}_i))$
5: Set $L_i = L$ and $\alpha_{i+1} = \frac{1+\sqrt{1+4\alpha_i^2}}{2}$
6: **end for**

2.1 The Associated Proximal Operator and Its Efficient Computation

The proximal operator associated with the overlapping group Lasso penalty is defined as follows:

$$\pi^{\lambda_1}_{\lambda_2}(\mathbf{v}) = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ g^{\lambda_1}_{\lambda_2}(\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|^2 + \phi^{\lambda_1}_{\lambda_2}(\mathbf{x}) \right\}, \tag{2.4}$$

which is a special case of (2.1) by setting $l(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|^2$. It can be verified that the approximate solution $\mathbf{x}_{i+1} = \arg\min_{\mathbf{y}} f_{L_i,\mathbf{s}_i}(\mathbf{y})$ is given by $\mathbf{x}_{i+1} = \pi^{\lambda_1/L_i}_{\lambda_2/L_i}(\mathbf{s}_i - \frac{1}{L_i}l'(\mathbf{s}_i))$. Recently, it has been shown in [81] that, the efficient computation of the proximal operator is key to many sparse learning algorithms. Next, I focus on the efficient computation of

$\pi_{\lambda_2}^{\lambda_1}(\mathbf{v})$ in (2.4) for a given $\mathbf{v}$. The rest of this section is organized as follows. Firstly, I discuss some key properties of the proximal operator, based on which I propose a pre-processing technique that will significantly reduce the size of the problem. I then proposed to solve it via the dual formulation, and the duality gap is also derived.

Key Properties of the Proximal Operator

I first reveal several basic properties of the proximal operator $\pi_{\lambda_2}^{\lambda_1}(\mathbf{v})$.

Lemma 1. *Suppose that $\lambda_1, \lambda_2 \geq 0$, and $w_i > 0$, for $i = 1, 2, \ldots, g$. Let $\mathbf{x}^* = \pi_{\lambda_2}^{\lambda_1}(\mathbf{v})$. The following holds: 1) if $v_i > 0$, then $0 \leq x_i^* \leq v_i$; 2) if $v_i < 0$, then $v_i \leq x_i^* \leq 0$; 3) if $v_i = 0$, then $x_i^* = 0$; 4)* $\mathrm{SGN}(\mathbf{v}) \subseteq \mathrm{SGN}(\mathbf{x}^*)$*; and 5)* $\pi_{\lambda_2}^{\lambda_1}(\mathbf{v}) = \mathrm{sgn}(\mathbf{v}) \odot \pi_{\lambda_2}^{\lambda_1}(|\mathbf{v}|)$.

Proof. When $\lambda_1, \lambda_2 \geq 0$, and $w_i \geq 0$, for $i = 1, 2, \ldots, g$, the objective function $g_{\lambda_2}^{\lambda_1}(\cdot)$ is strictly convex, thus $\mathbf{x}^*$ is the unique minimizer. I first show if $v_i > 0$, then $0 \leq x_i^* \leq v_i$. If $x_i^* > v_i$, then I can construct a $\hat{\mathbf{x}}$ as follows: $\hat{x}_j = x_j^*, j \neq i$ and $\hat{x}_i = v_i$. Similarly, if $x_i^* < 0$, then I can construct a $\hat{\mathbf{x}}$ as follows: $\hat{x}_j = x_j^*, j \neq i$ and $\hat{x}_i = 0$. It is easy to verify that $\hat{\mathbf{x}}$ achieves a lower objective function value than $\mathbf{x}^*$ in both cases. I can prove the second and the third properties using similar arguments. Finally, I can prove the fourth and the fifth properties using the definition of $\mathrm{SGN}(\cdot)$ and the first three properties. □

Next, I show that $\pi_{\lambda_2}^{\lambda_1}(\cdot)$ can be directly derived from $\pi_{\lambda_2}^{0}(\cdot)$ by soft-thresholding. Thus, I only need to focus on the case when $\lambda_1 = 0$. This simplifies the optimization in (2.4). It is an extension of the result for Fused Lasso in [82].

Theorem 1. *Let* $\mathbf{u} = \mathrm{sgn}(\mathbf{v}) \odot \max(|\mathbf{v}| - \lambda_1, 0)$*, and*

$$\pi_{\lambda_2}^{0}(\mathbf{u}) = \arg\min_{\mathbf{x} \in \mathbb{R}^p} \left\{ h_{\lambda_2}(\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|^2 + \lambda_2 \sum_{i=1}^{g} w_i \|\mathbf{x}_{G_i}\| \right\}. \tag{2.5}$$

Then, the following holds: $\pi_{\lambda_2}^{\lambda_1}(\mathbf{v}) = \pi_{\lambda_2}^{0}(\mathbf{u})$.

Proof. Denote the unique minimizer of $h_{\lambda_2}(\cdot)$ as $\mathbf{x}^*$. The sufficient and necessary condition for the optimality of $\mathbf{x}^*$ is:

$$\mathbf{0} \in \partial h_{\lambda_2}(\mathbf{x}^*) = \mathbf{x}^* - \mathbf{u} + \partial \phi^0_{\lambda_2}(\mathbf{x}^*), \tag{2.6}$$

where $\partial h_{\lambda_2}(\mathbf{x})$ and $\partial \phi^0_{\lambda_2}(\mathbf{x})$ are the sub-differential sets of $h_{\lambda_2}(\cdot)$ and $\phi^0_{\lambda_2}(\cdot)$ at $\mathbf{x}$, respectively.

Next, I need to show $\mathbf{0} \in \partial g^{\lambda_1}_{\lambda_2}(\mathbf{x}^*)$. The sub-differential of $g^{\lambda_1}_{\lambda_2}(\cdot)$ at $\mathbf{x}^*$ is given by

$$\partial g^{\lambda_1}_{\lambda_2}(\mathbf{x}^*) = \mathbf{x}^* - \mathbf{v} + \partial \phi^{\lambda_1}_{\lambda_2}(\mathbf{x}^*) = \mathbf{x}^* - \mathbf{v} + \lambda_1 \mathrm{SGN}(\mathbf{x}^*) + \partial \phi^0_{\lambda_2}(\mathbf{x}^*). \tag{2.7}$$

It follows from the definition of $\mathbf{u}$ that $\mathbf{u} \in \mathbf{v} - \lambda_1 \mathrm{SGN}(\mathbf{u})$. Using the fourth property in Lemma 1, I have $\mathrm{SGN}(\mathbf{u}) \subseteq \mathrm{SGN}(\mathbf{x}^*)$. Thus,

$$\mathbf{u} \in \mathbf{v} - \lambda_1 \mathrm{SGN}(\mathbf{x}^*). \tag{2.8}$$

It follows from (2.6)-(2.8) that $\mathbf{0} \in \partial g^{\lambda_1}_{\lambda_2}(\mathbf{x}^*)$. □

It follows from Theorem 1 that, I only need to focus on the optimization of (2.5) in the following discussion. The difficulty in the optimization of (2.5) lies in the large number of groups that may overlap. In practice, many groups will be zero, thus achieving a sparse solution (a sparse solution is desirable in many applications).

However, the zero groups are not known in advance. The key question I aim to address is how to identify as many zero groups as possible to reduce the complexity of the optimization. Next, I present a sufficient condition for a group to be zero.

Lemma 2. *Denote the minimizer of $h_{\lambda_2}(\cdot)$ in* (2.5) *by $\mathbf{x}^*$. If the i-th group satisfies $\|\mathbf{u}_{G_i}\| \leq \lambda_2 w_i$, then $\mathbf{x}^*_{G_i} = \mathbf{0}$, i.e., the i-th group is zero.*

Proof. Decompose $h_{\lambda_2}(\mathbf{x})$ into two parts as follows:

$$h_{\lambda_2}(\mathbf{x}) = \left(\frac{1}{2}\|\mathbf{x}_{G_i} - \mathbf{u}_{G_i}\|^2 + \lambda_2 w_i \|\mathbf{x}_{G_i}\| \right) + \left(\frac{1}{2}\|\mathbf{x}_{\overline{G}_i} - \mathbf{u}_{\overline{G}_i}\|^2 + \lambda_2 \sum_{j \neq i} w_j \|\mathbf{x}_{G_j}\| \right), \tag{2.9}$$

where $\overline{G}_i = \{1,2,\ldots,p\} - G_i$ is the complementary set of G_i. Consider the minimization of $h_{\lambda_2}(\mathbf{x})$ in terms of $\mathbf{x}_{G_i}$ when $\mathbf{x}_{\overline{G}_i} = \mathbf{x}^*_{\overline{G}_i}$ is fixed. It can be verified that if $\|\mathbf{u}_{G_i}\| \leq \lambda_2 w_i$, then $\mathbf{x}^*_{G_i} = \mathbf{0}$ minimizes both terms in (2.9) simultaneously. Thus I have $\mathbf{x}^*_{G_i} = \mathbf{0}$. □

Lemma 2 may not identify many true zero groups due to the strong condition imposed. The lemma below weakens the condition in Lemma 2. Intuitively, for a group G_i, I first identify all existing zero groups that overlap with G_i, and then compute the overlapping index subset S_i of G_i as:

$$S_i = \bigcup_{j \neq i, \mathbf{x}^*_{G_j} = \mathbf{0}} (G_j \cap G_i). \tag{2.10}$$

I can show that $\mathbf{x}^*_{G_i} = \mathbf{0}$ if $\|\mathbf{u}_{G_i - S_i}\| \leq \lambda_2 w_i$ is satisfied. Note that this condition is much weaker than the condition in Lemma 2, which requires that $\|\mathbf{u}_{G_i}\| \leq \lambda_2 w_i$.

Lemma 3. *Denote the minimizer of $h_{\lambda_2}(\cdot)$ by $\mathbf{x}^*$. Let S_i, a subset of G_i, be defined in* (2.10). *If $\|\mathbf{u}_{G_i - S_i}\| \leq \lambda_2 w_i$ holds, then $\mathbf{x}^*_{G_i} = \mathbf{0}$.*

Proof. Suppose that I have identified a collection of zero groups. By removing these groups, the original problem (2.5) can then be reduced to:

$$\min_{\mathbf{x}(I_1) \in \mathbb{R}^{|I_1|}} \frac{1}{2} \|\mathbf{x}(I_1) - \mathbf{u}(I_1)\|^2 + \lambda_2 \sum_{i \in \mathscr{G}_1} w_i \|\mathbf{x}_{G_i - S_i}\|$$

where I_1 is the reduced index set, i.e., $I_1 = \{1,2,\ldots,p\} - \bigcup_{i:\mathbf{x}^*_{G_i} = \mathbf{0}} G_i$, and $\mathscr{G}_1 = \{i : \mathbf{x}^*_{G_i} \neq \mathbf{0}\}$ is the index set of the remaining non-zero groups. Note that $\forall i \in \mathscr{G}_1$, $G_i - S_i \in I_1$. By applying Lemma 2 again, I show that if $\|\mathbf{u}_{G_i - S_i}\| \leq \lambda_2 w_i$ holds, then $\mathbf{x}^*_{G_i - S_i} = \mathbf{0}$. Thus, $\mathbf{x}^*_{G_i} = \mathbf{0}$. □

Lemma 3 naturally leads to an iterative procedure for identifying the zero groups: For each group G_i, if $\|\mathbf{u}_{G_i}\| \leq \lambda_2 w_i$, then I set $\mathbf{u}_{G_i} = \mathbf{0}$; I cycle through all groups repeatedly until $\mathbf{u}$ does not change. Let $p' = |\{u_i : u_i \neq 0\}|$ be the number of nonzero

elements in $\mathbf{u}$, $g' = |\{\mathbf{u}_{G_i} : \mathbf{u}_{G_i} \neq \mathbf{0}\}|$ be the number of the nonzero groups, and $\mathbf{x}^*$ denote the minimizer of $h_{\lambda_2}(\cdot)$. It follows from Lemma 3 and Lemma 1 that, if $u_i = 0$, then $x_i^* = 0$. Therefore, by applying the above iterative procedure, I can find the minimizer of (2.5) by solving a reduced problem that has $p' \leq p$ variables and $g' \leq g$ groups. With some abuse of notation, I still use (2.5) to denote the resulting reduced problem. In addition, from Lemma 1, I only focus on $\mathbf{u} > 0$ in the following discussion, and the analysis can be easily generalized to the general $\mathbf{u}$.

Reformulation as an Equivalent Smooth Convex Optimization Problem

It follows from the first two properties of Lemma 1 that, I can rewrite (2.5) as:

$$\pi^0_{\lambda_2}(\mathbf{u}) = \arg \min_{\substack{\mathbf{x}\in\mathbb{R}^p \\ \mathbf{0}\preceq\mathbf{x}\preceq\mathbf{u}}} h_{\lambda_2}(\mathbf{x}), \tag{2.11}$$

where $\preceq$ denotes the element-wise inequality, and

$$h_{\lambda_2}(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|^2 + \lambda_2 \sum_{i=1}^{g} w_i \|\mathbf{x}_{G_i}\|,$$

and the minimizer of $h_{\lambda_2}(\cdot)$ is constrained to be non-negative due to $\mathbf{u} > 0$ (refer to the discussion at the end of Section 2.1).

Making use of the dual norm of the Euclidean norm $\|\cdot\|$, I can rewrite $h_{\lambda_2}(\mathbf{x})$ as:

$$h_{\lambda_2}(\mathbf{x}) = \max_{Y\in\Omega} \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|^2 + \sum_{i=1}^{g} \langle \mathbf{x}, Y^i \rangle, \tag{2.12}$$

where Ω is defined as follows:

$$\Omega = \{Y \in \mathbb{R}^{p\times g} : Y^i_{\overline{G}_i} = \mathbf{0}, \|Y^i\| \leq \lambda_2 w_i, i = 1,2,\ldots,g\},$$

$\overline{G}_i$ is the complementary set of G_i, Y is a sparse matrix satisfying $Y_{ij} = 0$ if the i-th feature does not belong to the j-th group, i.e., $i \notin G_j$, and Y^i denotes the i-th column of Y. As a result, I can reformulate (2.11) as the following min-max problem:

$$\min_{\substack{\mathbf{x}\in\mathbb{R}^p \\ \mathbf{0}\preceq\mathbf{x}\preceq\mathbf{u}}} \max_{Y\in\Omega} \left\{ \psi(\mathbf{x}, Y) = \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|^2 + \langle \mathbf{x}, Y\mathbf{e} \rangle \right\}, \tag{2.13}$$

where $\mathbf{e} \in \mathbb{R}^g$ is a vector of ones. It is easy to verify that $\psi(\mathbf{x}, Y)$ is convex in $\mathbf{x}$ and concave in Y, and the constraint sets are closed convex for both $\mathbf{x}$ and Y. Thus, (2.13) has a saddle point, and the min-max can be exchanged.

It is easy to verify that for a given Y, the optimal $\mathbf{x}$ minimizing $\psi(\mathbf{x}, Y)$ in (2.13) is given by

$$\mathbf{x} = \max(\mathbf{u} - Y\mathbf{e}, \mathbf{0}). \tag{2.14}$$

Plugging (2.14) into (2.13), I obtain the following minimization problem with regard to Y:

$$\min_{Y \in \mathbb{R}^{p \times g}: Y \in \Omega} \{\omega(Y) = -\psi(\max(\mathbf{u} - Y\mathbf{e}, \mathbf{0}), Y)\}. \tag{2.15}$$

My methodology for minimizing $h_{\lambda_2}(\cdot)$ defined in (2.5) is to first solve (2.15), and then construct the solution to $h_{\lambda_2}(\cdot)$ via (2.14). Using standard optimization techniques, I can show that the function $\omega(\cdot)$ is continuously differentiable with Lipschitz continuous gradient. Therefore, I convert the non-smooth problem (2.11) to the smooth problem (2.15), making the smooth convex optimization tools applicable. In this thesis, I employ the accelerated gradient descent to solve (2.15), due to its fast convergence property. Note that, the Euclidean projection onto the set Ω can be computed in closed form. I would like to emphasize here that, the problem (2.15) may have a much smaller size than (2.4).

Computing the Duality Gap

I show how to estimate the duality gap of the min-max problem (2.13), which can be used to check the quality of the solution and determine the convergence of the algorithm.

For any given approximate solution $\tilde{Y} \in \Omega$ for $\omega(Y)$, I can construct the approximate solution $\tilde{\mathbf{x}} = \max(\mathbf{u} - \tilde{Y}\mathbf{e}, \mathbf{0})$ for $h_{\lambda_2}(\mathbf{x})$. The duality gap for the min-max problem (2.13) at the point $(\tilde{\mathbf{x}}, \tilde{Y})$ can be computed as:

$$\text{gap}(\tilde{Y}) = \max_{Y \in \Omega} \psi(\tilde{\mathbf{x}}, Y) - \min_{\substack{\mathbf{x} \in \mathbb{R}^p \\ \mathbf{0} \preceq \mathbf{x} \preceq \mathbf{u}}} \psi(\mathbf{x}, \tilde{Y}). \tag{2.16}$$

The main result of this section is summarized in the following theorem:

Theorem 2. *Let* $\text{gap}(\tilde{Y})$ *be the duality gap defined in* (2.16)*. Then, the following holds:*

$$\text{gap}(\tilde{Y}) = \lambda_2 \sum_{i=1}^{g} (w_i \|\tilde{\mathbf{x}}_{G_i}\| - \langle \tilde{\mathbf{x}}_{G_i}, \tilde{Y}^i_{G_i} \rangle). \tag{2.17}$$

In addition, I have

$$\omega(\tilde{Y}) - \omega(Y^*) \leq \text{gap}(\tilde{Y}), \tag{2.18}$$

$$h(\tilde{\mathbf{x}}) - h(\mathbf{x}^*) \leq \text{gap}(\tilde{Y}). \tag{2.19}$$

Proof. Denote $(\mathbf{x}^*, Y^*)$ as the optimal solution to the problem (2.13). From (2.12)-(2.15), I have

$$-\omega(\tilde{Y}) = \psi(\tilde{\mathbf{x}}, \tilde{Y}) = \min_{\substack{\mathbf{x} \in \mathbb{R}^p \\ \mathbf{0} \preceq \mathbf{x} \preceq \mathbf{u}}} \psi(\mathbf{x}, \tilde{Y}) \leq \psi(\mathbf{x}^*, \tilde{Y}), \tag{2.20}$$

$$\psi(\mathbf{x}^*, \tilde{Y}) \leq \max_{Y \in \Omega} \psi(\mathbf{x}^*, Y) = \psi(\mathbf{x}^*, Y^*) = -\omega(Y^*), \tag{2.21}$$

$$h_{\lambda_2}(\mathbf{x}^*) = \psi(\mathbf{x}^*, Y^*) = \min_{\substack{\mathbf{x} \in \mathbb{R}^p \\ \mathbf{0} \preceq \mathbf{x} \preceq \mathbf{u}}} \psi(\mathbf{x}, Y^*) \leq \psi(\tilde{\mathbf{x}}, Y^*), \tag{2.22}$$

$$\psi(\tilde{\mathbf{x}}, Y^*) \leq \max_{Y \in \Omega} \psi(\tilde{\mathbf{x}}, Y) = h_{\lambda_2}(\tilde{\mathbf{x}}). \tag{2.23}$$

Incorporating (2.11), (2.20)-(2.23), I prove (2.17)-(2.19). □

In my experiments, I terminate the algorithm when the estimated duality gap is less than 10^{-10}.

2.2 ℓ_q Norm Overlapping Group Lasso

In this section, I extend my previous results to solving the overlapping group lasso formulation (2.1) based on the ℓ_q norm with $q > 1$. Specifically, I extend the group lasso penalty (2.2) to:

$$\phi^{\lambda_1}_{q,\lambda_2}(\mathbf{x}) = \lambda_1 \|\mathbf{x}\|_1 + \lambda_2 \sum_{i=1}^{g} w_i \|\mathbf{x}_{G_i}\|_q. \tag{2.24}$$

To extend to the ℓ_q norm case, the only change to Algorithm 1 is to generalize the proximal operator:

$$\pi^{\lambda_1}_{q,\lambda_2}(\mathbf{v}) = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ g^{\lambda_1}_{q,\lambda_2}(\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|^2 + \phi^{\lambda_1}_{q,\lambda_2}(\mathbf{x}) \right\}. \tag{2.25}$$

In the rest of the section, I extend the properties of the proximal operator as well as the dual method from the ℓ_2 norm case to the general ℓ_q norm case.

Properties of the ℓ_q Proximal Operator

First of all, it is easy to verify that Lemma 1 and Theorem 1 hold for all $q > 1$ given that the ℓ_q norm is convex.

Denote the dual norm of the ℓ_q norm as $\ell_{\overline{q}}$ with $1/q + 1/\overline{q} = 1$, and $h_{q,\lambda_2}(\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|^2 + \phi^0_{q,\lambda_2}(\mathbf{x})$. I then extend the pre-processing techniques in the following two lemmas.

Lemma 4. *Denote the minimizer of $h_{q,\lambda_2}(\cdot)$ by $\mathbf{x}^*$. If the i-th group satisfies $\|\mathbf{u}_{G_i}\|_{\overline{q}} \leq \lambda_2 w_i$, then $\mathbf{x}^*_{G_i} = \mathbf{0}$, i.e., the i-th group is zero.*

Proof. I decompose $h_{q,\lambda_2}(\mathbf{x})$ into two parts as follows:

$$\begin{aligned} h_{q,\lambda_2}(\mathbf{x}) = &\left(\frac{1}{2}\|\mathbf{x}_{G_i} - \mathbf{u}_{G_i}\|^2 + \lambda_2 w_i \|\mathbf{x}_{G_i}\|_q \right) \\ &+ \left(\frac{1}{2}\|\mathbf{x}_{\overline{G}_i} - \mathbf{u}_{\overline{G}_i}\|^2 + \lambda_2 \sum_{j\neq i} w_j \|\mathbf{x}_{G_j}\|_q \right), \end{aligned} \tag{2.26}$$

where $\overline{G}_i = \{1,2,\ldots,p\} - G_i$ is the complementary set of G_i. I consider the minimization of $h_{q,\lambda_2}(\mathbf{x})$ in terms of $\mathbf{x}_{G_i}$ when $\mathbf{x}_{\overline{G}_i} = \mathbf{x}^*_{\overline{G}_i}$ is fixed.

Clearly, $\mathbf{x}^*_{G_i} = 0$ minimizes the second term in (2.26). Therefore, I just need to show that $\mathbf{x}^*_{G_i} = 0$ minimizes $y(\mathbf{x}_{G_i})$ defined as following:

$$y(\mathbf{x}_{G_i}) = \frac{1}{2}\|\mathbf{x}_{G_i} - \mathbf{u}_{G_i}\|^2 + \lambda_2 w_i \|\mathbf{x}_{G_i}\|_q$$

Given any direction $d \in \mathbb{R}^{|G_i|}$, I take the directional derivative of y at point $\mathbf{0}$:

$$\begin{aligned} Dy(\mathbf{0})[d] &= \lim_{\alpha \downarrow 0} \frac{y(\alpha d) - y(\mathbf{0})}{\alpha} \\ &= \lim_{\alpha \downarrow 0} \frac{\frac{1}{2}\|\alpha d - \mathbf{u}_{G_i}\|^2 + \lambda_2 w_i \|\alpha d\|_q - \frac{1}{2}\|\mathbf{u}_{G_i}\|^2}{\alpha} \\ &= -\langle d, \mathbf{u}_{G_i} \rangle + \lambda_2 w_i \|d\|_q \\ &\geq -\|d\|_q \|\mathbf{u}_{G_i}\|_{\overline{q}} + \lambda_2 w_i \|d\|_q \\ &\geq 0, \quad \forall d \end{aligned}$$

where the last inequality follows since $\|\mathbf{u}_{G_i}\|_{\overline{q}} \leq \lambda_2 w_i$. Thus $\mathbf{x}^*_{G_i} = 0$. □

Similarly to Lemma 3, I have:

Lemma 5. *Denote the minimizer of $h_{q,\lambda_2}(\cdot)$ by $\mathbf{x}^*$. Let S_i, a subset of G_i, be defined in* (2.10). *If $\|\mathbf{u}_{G_i - S_i}\|_{\overline{q}} \leq \lambda_2 w_i$ holds, then $\mathbf{x}^*_{G_i} = \mathbf{0}$.*

The proof is similar to Lemma 3 based on the result in Lemma 4.

Extending the Dual Method to the ℓ_q Case

When reformulating (2.25) as an equivalent smooth problem, I only need to make two changes:

- The feasible region of the dual variable Y is generalized as:

$$\begin{aligned} \Omega_q = \{Y \in \mathbb{R}^{p \times g} : & Y^i_{\overline{G}_i} = \mathbf{0}, \|Y^i\|_{\overline{q}} \leq \lambda_2 w_i, \\ & i = 1, 2, \ldots, g\} \end{aligned}$$

- During the optimization process, I need to compute the Euclidean projection onto the ℓ_q ball, which can be calculated efficiently [83].

The duality gap is now calculated as:

$$\text{gap}_p(\tilde{Y}) = \sum_{i=1}^{g} (\lambda_2 w_i \|\tilde{\mathbf{x}}_{G_i}\|_q - \langle \tilde{\mathbf{x}}_{G_i}, \tilde{Y}^i_{G_i} \rangle).$$

It is easy to verify that this value can still be used to check the convergence of my proposed dual method in the ℓ_q norm case.

2.3 Overlapping Group Lasso via the Capped Norm

In this section, I consider the following problem:

$$\min_{\mathbf{x}\in\mathbb{R}^p} l(\mathbf{x}) + \lambda_1 \|\mathbf{x}\|_0 + \lambda_2 \sum_{i=1}^{g} w_i I(\|\mathbf{x}_{G_i}\| \neq 0), \tag{2.27}$$

where $I(\cdot)$ is the indicator function. Note that this is an NP-hard problem where convex relaxation such as (2.1) is normally applied. However, due to the looseness of convex relaxation, the ℓ_1 norm type regularization will introduce bias to the parameter estimation [84, 85]. Several recent works use the non-convex capped norm that is closer to the ℓ_0 norm than the ℓ_1 norm as follows [84–86]:

$$\begin{aligned} \|\mathbf{x}\|_0 &\approx \sum_{j=1}^{p} \min\left(1, \frac{|\mathbf{x}_j|}{\theta_1}\right) \\ \sum_{i=1}^{g} w_i I(\|\mathbf{x}_{G_i}\| \neq 0) &\approx \sum_{i=1}^{g} w_i \min\left(1, \frac{\|\mathbf{x}_{G_i}\|}{\theta_2}\right), \end{aligned} \tag{2.28}$$

for some small $\theta_1, \theta_2 > 0$. Parameter estimation using the non-convex capped norm has been studied. It has been shown that under appropriate conditions, the local solution obtained by using the capped norm has better statistical property than the one based on the convex ℓ_1 norm penalty.

The approximation used in (2.28) is still non-convex. Following [84–86], I use the following two decompositions:

$$\begin{aligned} &\sum_{j=1}^{p} \min\left(1, \frac{|\mathbf{x}_j|}{\theta_1}\right) \\ =&\frac{1}{\theta_1}\left[\|\mathbf{x}\|_1 - \sum_{j=1}^{p} \max(|\mathbf{x_j}| - \theta_1, 0)\right] \end{aligned} \tag{2.29}$$

and

$$\begin{aligned} &\sum_{i=1}^{g} w_i \min\left(1, \frac{\|\mathbf{x}_{G_i}\|}{\theta_2}\right) \\ =&\frac{1}{\theta_2}\left[\sum_{i=1}^{g} w_i \|\mathbf{x}_{G_i}\| - \sum_{i=1}^{g} w_i \max(\|\mathbf{x}_{G_i}\| - \theta_2, 0)\right]. \end{aligned} \tag{2.30}$$

Combining (2.28), (2.29) and (2.30), I can approximate (2.27) as:

$$\min_{\mathbf{x}\in\mathbb{R}^p} l(\mathbf{x}) + \frac{\lambda_1}{\theta_1}\|\mathbf{x}\|_1 + \frac{\lambda_2}{\theta_2}\sum_{i=1}^{g} w_i\|\mathbf{x}_{G_i}\| - P(\mathbf{x}) - D(\mathbf{x}) \tag{2.31}$$

where

$$P(\mathbf{x}) = \frac{\lambda_1}{\theta_1}\sum_{j=1}^{p}\max(|\mathbf{x}_j| - \theta_1, 0)$$

and

$$\begin{aligned} D(\mathbf{x}) &= \frac{\lambda_2}{\theta_2}\sum_{i=1}^{g} w_i D_i(\mathbf{x}_{G_i}) \\ &= \frac{\lambda_2}{\theta_2}\sum_{i=1}^{g} w_i \max(\|\mathbf{x}_{G_i}\| - \theta_2, 0). \end{aligned}$$

Note that both P and D are convex functions, and therefore I have converted the problem into a "difference of two convex functions" (DC) programming.

It can be shown that

$$\frac{\partial}{\partial \mathbf{x}_j} P(\mathbf{x}) \ni \begin{cases} \frac{\lambda_1}{\theta_1}\mathrm{sgn}(\mathbf{x}_j) & |\mathbf{x}_j| > \theta_1 \\ 0 & |\mathbf{x}_j| \le \theta_1 \end{cases}$$

and

$$\frac{\partial}{\partial \mathbf{x}_{G_i}} D_i(\mathbf{x}_{G_i}) \ni \begin{cases} \frac{\mathbf{x}_{G_i}}{\|\mathbf{x}_{G_i}\|} & \|\mathbf{x}_{G_i}\| > \theta_2 \\ \mathbf{0} & \|\mathbf{x}_{G_i}\| \le \theta_2 \end{cases}.$$

I then propose to solve (2.31) using the DC programming, and the details are provided in Algorithm 2.

The sub-problem (2.32) can be solved using Algorithm 1. Therefore, by solving a sequence of overlapping group lasso problems, I can find a local solution for (2.31).

2.4 Experiments

In this section, I present extensive experiments to demonstrate the efficiency of my proposed methods. I use both synthetic data sets and a real world data set and the evaluation is done in various problem size and precision settings. The proposed

Algorithm 2 DC Programming for Overlapping Group Lasso with the Capped Norm

Input: $\theta_1, \theta_2 > 0, \mathbf{x}^0, k$
Output: $\mathbf{x}^{k+1}$

1: Initialize $\mathbf{x}^1 = \mathbf{x}^0$
2: **for** $i = 1$ to k **do**
3: Choose $U^k \in \partial P(\mathbf{x}^k)$ and $V^k \in \partial D(\mathbf{x}^k)$
4: Solve

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ l(\mathbf{x}) + \frac{\lambda_1}{\theta_1}\|\mathbf{x}\|_1 + \frac{\lambda_2}{\theta_2}\sum_{i=1}^{g} w_i \|\mathbf{x}_{G_i}\| - \langle U^k + V^k, \mathbf{x}\rangle \right\} \tag{2.32}$$

5: Set $k \leftarrow k+1$
6: **end for**

algorithms are mainly implemented in Matlab, with the proximal operator implemented in standard C for improved efficiency. The source codes can be found online [87].

Several state-of-the-art methods are also included for comparison purpose, including SLasso developed by Jenatton [1] (with key components implemented in C), the ADMM reformulation suggested by Boyd [2], the Prox-Grad method proposed by Chen [3] and the Picard-Nesterov algorithm [88].

Synthetic Data

Efficiency of Calculating the Proximal Operator

In the first set of simulation I consider only the key component of our algorithm, the proximal operator. The group indices are predefined such that $G_1 = \{1, 2, \ldots, 10\}$, $G_2 = \{6, 7, \ldots, 20\}, \ldots$, with each group overlapping half of the previous group. The target vector $\mathbf{v} \in \mathbb{R}^p$ in (2.4) is generated randomly such that $\mathbf{v}_i \sim N(0,1)$. I fix $\lambda_1 = 1$ and $\lambda_2 = 10$.

100 examples are generated for each set of fixed problem size p and group size g, and for each particular random example, I first run the dual method till the gap is less than 10^{-8}, then I run ADMM and Dykstra method until a smaller function value is attained. The results are summarized in Figure 2.1. As we can observe from the figure, the dual

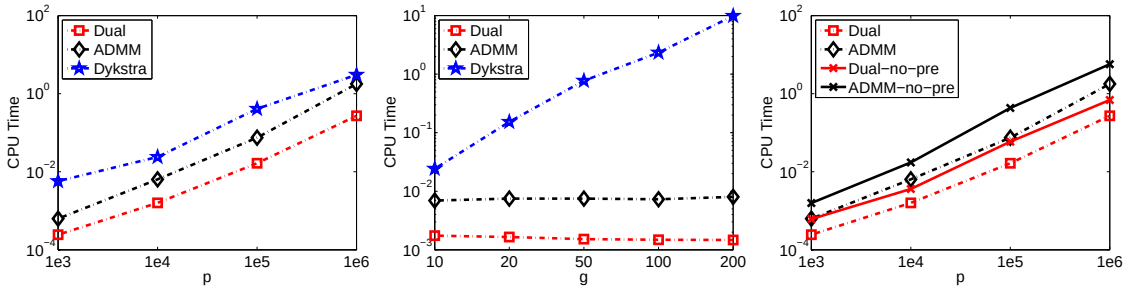


Figure 2.1: Time comparison for computing the proximal operators. The group number is fixed in the left figure and the problem size is fixed in the middle figure. The right figure illustrates the effectiveness of the pre-processing.

formulation yields the best performance, followed closely by ADMM and then the Dykstra method. I can also observe that my method scales very well to high dimensional problems, since even with $p = 10^6$, the proximal operator can be computed in a few seconds. It is also not surprising that the Dykstra method is much more sensitive to the number of groups, which equals to the number of projections in one Dykstra step.

To illustrate the effectiveness of my pre-processing technique, I repeat the previous experiment by removing the pre-processing step. The results are shown in the right plot of Figure 2.1. As we can observe from the figure, the proposed pre-processing technique effectively reduces the computational time. As is evident from Figure 2.1, the dual formulation proposed in Section 2.1 consistently outperforms other proximal splitting methods. In the following experiments, only the dual method with the pre-processing step will be used for computing the proximal operator, and my method will then be called as "FoGLasso".

Sparse Pattern Recovery

Although the focus of this paper is on the efficiency of the proposed algorithm, it is also interesting to see if the overlapping group Lasso formulation can recover the underlying sparse pattern. For a given problem size n, p and group size g, I first define the overlapping groups as in Section 2.4. I then generate the ground truth model $\mathbf{x}_0$ with each

entry sampled i.i.d. from a standard Gaussian distribution. Next, I randomly set half of the predefined groups and half of the remaining entries of $\mathbf{x}_0$ to be 0. I sample the entries of the data matrix $A \in \mathbb{R}^{n\times p}$ i.i.d. from a standard Gaussian distribution; and the response vector $\mathbf{b}$ is obtained from $\mathbf{b} = A\mathbf{x}_0 + \varepsilon$, where $\varepsilon \sim N(0, \sigma^2 I_{n\times n})$.

I solve (2.1) with the least squares loss $l(\mathbf{x}) = \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|^2$, and I set $w_i = \sqrt{|G_i|}$, and $\lambda_1 = \lambda_2 = \gamma \times \lambda_1^{\max}$, where $|G_i|$ denotes the size of the i-th group G_i, $\lambda_1^{\max} = \|A^{\mathrm{T}}\mathbf{b}\|_\infty$ [the zero point is a solution to (2.1) if $\lambda_1 \geq \lambda_1^{\max}$], and γ is chosen from the set $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$. Denote the obtained solution as $\mathbf{x}$. The following two criteria are used to evaluate the recovery performance:

- Entry Recovery Rate:

$$\Pr\{\|\mathbf{x}_0(i)\|_0 = \|\mathbf{x}(i)\|_0\},$$

  which is the entry-wise accuracy of sparse pattern recovery.

- Group Recovery Rate:

$$\Pr\{I(\|\mathbf{x}_0(G_i)\| = 0) = I(\|\mathbf{x}(G_i)\| = 0)\},$$

  where $I(\cdot)$ is the indicator function. This can be considered as the group-wise accuracy of sparse pattern recovery.

I set $n \in \{200, 300, 400\}$, $p = 600$, $\sigma = 10^{-3}$ and $g = 60$. For each γ value, 100 random instances are generated and the average performance for different problem sizes is reported in the top row of Figure 2.2.

Using similar problem settings, I also evaluate the recovery performance of the overlapping group Lasso formulation with the capped norm. I set $\theta_1 = \theta_2 = 0.01$, and instead of using a ratio, I set $\lambda_1 = \lambda_2 = \lambda \in \{0.1, 1, 1.5, 2, 3, 4, 5\}$. The results are summarized in the bottom row of Figure 2.2.

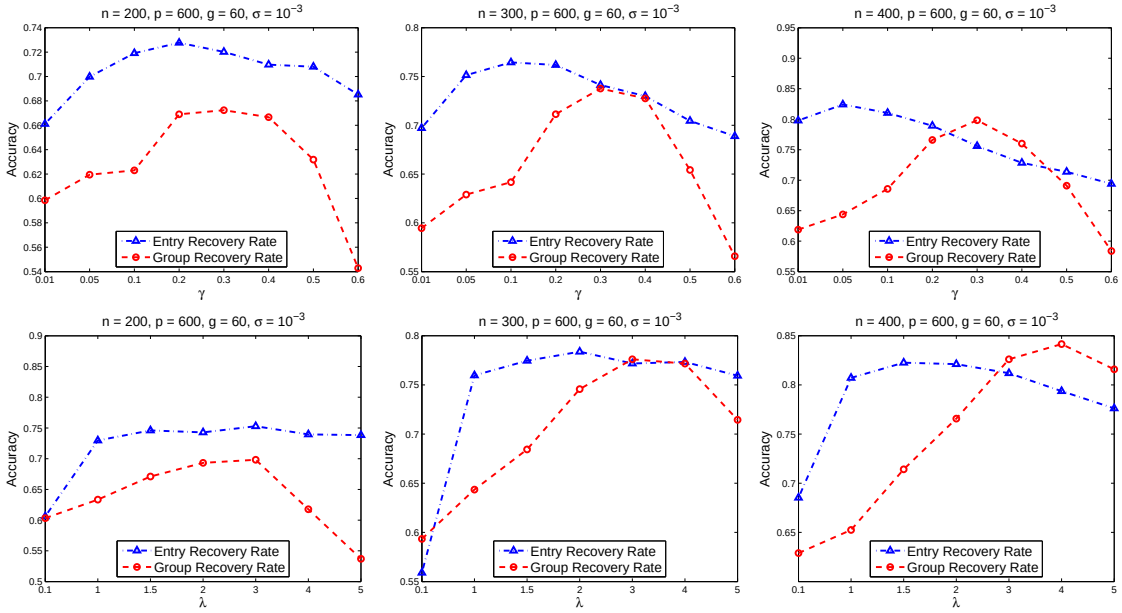


Figure 2.2: Performance of sparse pattern recovery of the convex overlapping group Lasso formulation (2.1) (top row) and the non-convex overlapping group Lasso formulation with the capped norm (2.31) (bottom row) on synthetic data with different problem sizes.

I can observe from Figure 2.2 that as I increase the sample size, the performance generally improves. The best performance is normally attained in the middle of the parameter space, where the solution is not too dense (mostly non-zeros) or too sparse (mostly zeros). My preliminary evaluation shows that using the non-convex formulation indeed improves the pattern recovery rate compared to the original formulation. For example, when the sample size is 400, the best group recovery rate for the original formulation is 0.81, while for the formulation with the capped norm, the rate is about 0.85.

Figure 2.2 illustrates the recovery performance across the parameter space. Here I also provide results with parameters selected via cross validation, which is usually done in practice. For each randomly generated example, I first use 4-fold cross validation to select the parameter with the smallest error. I then use this parameter to obtain the model $\mathbf{x}$, and compare it to the ground truth to obtain recovery performance. I repeat this process for 100 times and the results are summarized in Table 2.1. As I can see in Table 2.1, using the

Table 2.1: Cross-validation performance of sparse pattern recovery of the convex overlapping group Lasso formulation and the non-convex overlapping group Lasso formulation based on the capped norm on synthetic data with different problem sizes.

	Convex		Non-convex	
n	Entry Rate	Group Rate	Entry Rate	Group Rate
300	0.71	0.60	0.77	0.71
400	0.80	0.61	0.82	0.70

non-convex formulation also improves the pattern recovery rate when the parameters are selected using cross-validation. Further evaluation of this non-convex formulation in real world applications will be my future work.

Gene Expression Data

I have also conducted experiments to evaluate the efficiency of the proposed algorithm using the breast cancer gene expression data set [89], which consists of 8,141 genes in 295 breast cancer tumors (78 metastatic and 217 non-metastatic). For the sake of analyzing microarrays in terms of biologically meaningful gene sets, different approaches have been used to organize the genes into (overlapping) gene sets. In my experiments, I follow Jacob et al. [14] and employ the following two approaches for generating the overlapping gene sets (groups): pathways [90] and edges [91]. For pathways, the canonical pathways from the Molecular Signatures Database (MSigDB) [90] are used. It contains 639 groups of genes, of which 637 groups involve the genes in my study. The statistics of the 637 gene groups are summarized as follows: the average number of genes in each group is 23.7, the largest gene group has 213 genes, and 3,510 genes appear in these 637 groups with an average appearance frequency of about 4. For edges, the network built by Chuang [91] will be used, and I follow Jacob et al. [14] to extract 42,594 edges from the network, leading to 42,594 overlapping gene sets of size 2. All 8,141 genes appear in the 42,594 groups with an average appearance frequency of about 10. Here I set $\lambda_1 = \lambda_2 = \gamma \times \lambda_1^{\max}$,

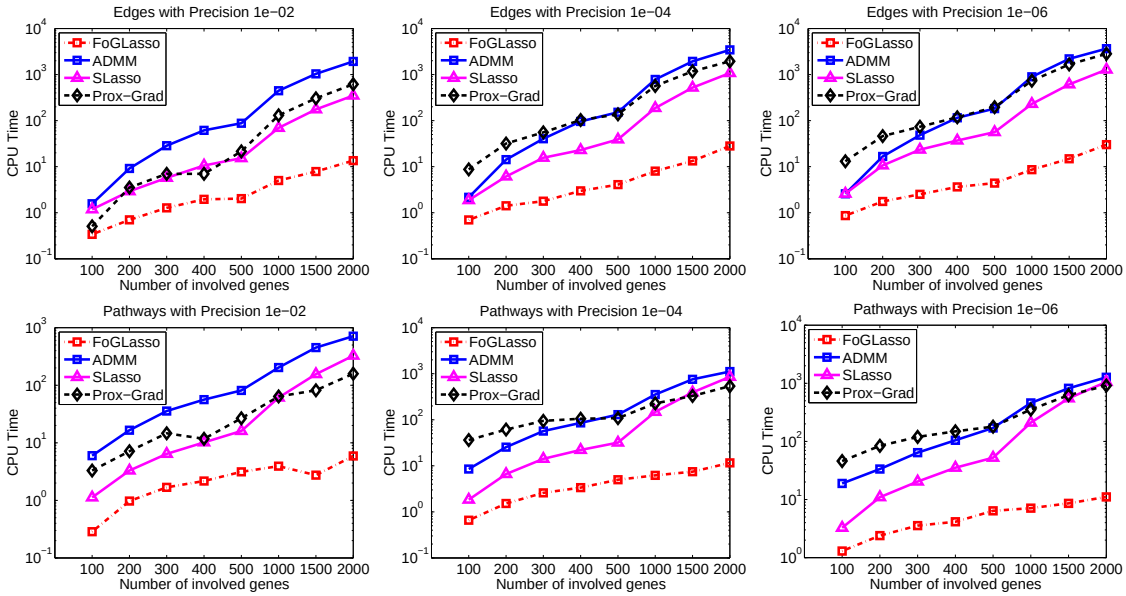


Figure 2.3: Comparison of SLasso [1], ADMM [2], Prox-Grad [3] and the proposed FoGLasso algorithm in terms of computational time (in seconds and in the logarithmic scale) when different numbers of genes (variables) are involved. Different precision levels are used for comparison.

where γ is chosen from the set

$$\{5\times10^{-1}, 2\times10^{-1}, 1\times10^{-1}, 5\times10^{-2}, 2\times10^{-2}, 1\times10^{-2}, 5\times10^{-3}, 2\times10^{-3}, 1\times10^{-3}\}.$$

Comparison with SLasso, Prox-Grad and ADMM

I first compare the proposed FoGLasso with the SLasso algorithm [1], ADMM [2] and Prox-Grad [3]. The comparisons are based on the computational time, since all these methods have efficient Matlab implementations with key components written in C. For a given γ, I first run SLasso till a certain precision level is reached, and then run the others until they achieve an objective function value smaller than or equal to that of SLasso. The precision level here is used as the convergence condition for SLasso, such that when the change of objective function value is smaller than a certain value, the algorithm terminates. Different precision levels of the solutions are evaluated such that a fair comparison can be made. I vary the number of genes involved, and report the total computational time (seconds) for all nine regularization parameters in Figure 2.3. I can

observe that: 1) for all precision levels, the proposed FoGLasso is much more efficient than SLasso, ADMM and Prox-Grad; 2) the advantage of FoGLasso over other three methods in efficiency grows with the increasing number of genes (variables). For example, with the grouping by pathways, FoGLasso is about 25 and 70 times faster than SLasso for 1000 and 2000 genes, respectively; and 3) the efficiency on edges is inferior to that on pathways, due to the larger number of overlapping groups. Additional scalability study of the proposed method using larger problem sizes can be found in Table 2.2.

Comparison with Picard-Nesterov

Since Picard-Nesterov was implemented purely in Matlab, a computational time comparison might not be fair. Therefore, only the number of iterations required for convergence is reported, as both methods adopt the first order method. Also note that, unlike the previous three methods (SLasso, ADMM and Prox-Grad), the pre-processing technique can be applied to Picard-Nesterov since it solves the same proximal operator in each iteration. To further validate the effectiveness of my pre-processing technique, I apply it to Picard-Nesterov as an independent method for comparison.

I use edges to generate the groups, and vary the problem size from 100 to 400, using the same set of regularization parameters. For each problem, I record both the number of outer iterations (the gradient steps) and the total number of inner iterations (the steps required for computing the proximal operators). The average number of iterations among all the regularization parameters are summarized in Table 2.3. As I can observe from the table, though Picard-Nesterov often takes less outer iterations to converge, it takes a lot more inner iterations to compute the proximal operator. It is easy to verify that the inner iterations in the Picard-Nesterov method and the proposed method have the same complexity of $O(pg)$. In terms of the pre-processing technique, we can see that in all cases the number of gradient steps remains exactly the same, while the number of inner

Table 2.2: Scalability study of the FoGLasso algorithm under different numbers (p) of genes involved. The reported results are the total computational time (seconds) for all nine regularization parameter values.

p	3000	4000	5000	6000	7000	8141
pathways	37.6	48.3	62.5	68.7	86.2	99.7
edges	58.8	84.8	102.7	140.8	173.3	247.8

Table 2.3: Comparison of FoGLasso, Picard-Nesterov and Picard-Nesterov with the proposed pre-processing technique using different numbers (p) of genes and various precision levels. For each particular method, the first row denotes the number of outer iterations required for convergence, while the second row represents the total number of inner iterations.

Precision Level	10^{-2}			10^{-4}			10^{-6}		
p	100	200	400	100	200	400	100	200	400
FoGLasso	81	189	353	192	371	1299	334	507	1796
	288	401	921	404	590	1912	547	727	2387
Picard-Nesterov	78	176	325	181	304	1028	318	504	1431
	8271	6.8e4	2.2e5	2.6e4	1.0e5	7.8e5	5.1e4	1.3e5	1.1e6
Picard-Nesterov-PreProc	78	176	325	181	304	1028	318	504	1431
	2683	3.8e4	1.1e5	8427	6.4e4	4.9e5	1.9e4	8.2e4	7.3e5

iterations is significantly reduced. This verifies that by using the proposed pre-processing technique, the proximal operator yields the same solution while solving a smaller problem.

Computation of the Proximal Operator

In this experiment, I run FoGLasso on the breast cancer data set using all 8,141 genes. I terminate FoGLasso if the change of the objective function value is less than 10^{-5}. I use the 42,594 edges to generate the overlapping groups. I set $\rho = 0.01$. The results are shown in Figure 2.4. The left plot shows that the objective function value decreases rapidly in the proposed FoGLasso. In the middle plot, I report the percentage of the identified zero groups by applying Lemma 3. My result shows that, 1) after 16 iterations, 50% of the zero groups are correctly identified; and 2) after 50 iterations, 80% of the zero groups are identified. Therefore, with Lemma 3, I can significantly reduce the problem size of the subsequent dual reformulation (see Section 2.1). In the right plot of Figure 2.4, I present the number of inner iterations for solving the proximal operator via the dual

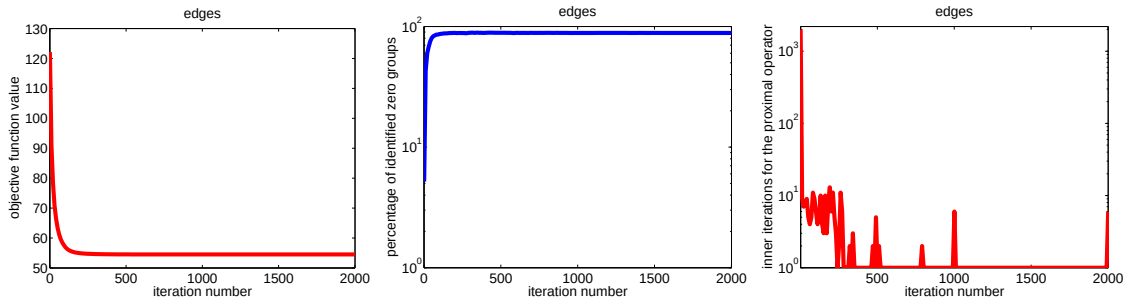


Figure 2.4: Performance of the computation of the proximal operator in FoGLasso. The left plot shows the objective function value during the FoGLasso iteration. The middle plot shows the percentage of the identified zero groups by applying Lemma 3. The right plot shows the number of inner iterations for achieving the duality gap less than 10^{-10} when one solves the proximal operator via the dual reformulation (see Section 3.2).

reformulation. We observe from the figure that the number of inner iterations decreases. This is because 1) the size of the reduced problem decreases when many zero groups are identified (see the middle plot); and 2) in solving the dual reformulation, I can apply the Y computed in the previous iteration as the "warm" start for computing the proximal operator in the next iteration.

Convergence with Inexact Proximal Operator

With an inexact proximal operator, the optimal convergence rate of the accelerated gradient descent might not be guaranteed [92, 93]. However, recent work [94] has shown that if the error introduced in the proximal operator decreases at a certain rate, the convergence rate of AGD remains the same as in the exact case. Specifically, if I denote the duality gap in the k step as $\varepsilon_k = \text{gap}(\tilde{Y}_k)$, AGD will converge at the optimal rate $O\left(\frac{1}{k^2}\right)$ if ε_k is of order $O\left(\frac{1}{k^{2+\delta}}\right)$ with $\delta > 0$.

One advantage of the proposed dual method is that the error of the proximal operator can be easily controlled by the duality gap. Here I continue to use the gene expression data set in Section 2.4 to evaluate how the error in the proximal operator affects the performance of the algorithm in practice. I use the following ways to terminate the calculation of (2.4):

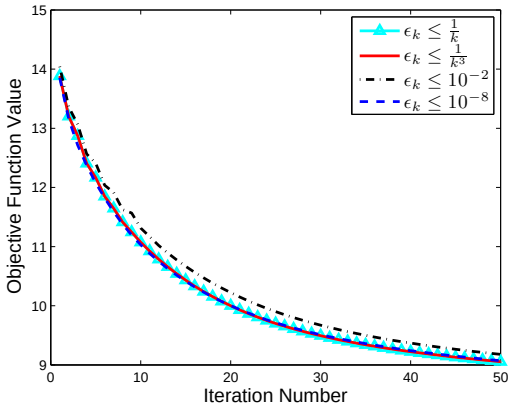


Figure 2.5: Illustration of the objective function values of the first 50 iterations with different stopping criteria used for computing the proximal operator.

- $\varepsilon_k \leq \frac{1}{k^n}$: terminate the calculation until the duality gap is below $\frac{1}{k^n}$.
- $\varepsilon_k \leq \varepsilon$: terminate the calculation until the duality gap is below a fixed value ε.

I use the objective function value against the number of iterations as the evaluation criterion, and the performance for different termination conditions is illustrated in Figure 2.5.

As we can see from Figure 2.5, the convergence of the objective value does not change dramatically with different termination conditions. For example, setting $\varepsilon \leq \frac{1}{k}$ and $\varepsilon \leq \frac{1}{k^3}$ performs equally well. This might due to the fact that in most cases, my proximal operator takes only one step to converge even with a small duality gap, e.g., 10^{-10} (as shown in the right plot, Figure 2.4).

Additional Discussions on Overlapping Group Lasso

Throughout this section, I have performed extensive experiments to illustrate the empirical performance of the proposed method. It is also interesting to analyze the relationship between the proposed method and the existing methods. In general, the methods for solving overlapping group lasso formulation can be divided into three groups:

- **AGD (FISTA) with proximal operator**, such as FoGLasso and Picard-Nesterov. These methods solve the non-smooth optimization by first-order methods that involve the computation of a proximal operator. When an analytical solution exists for the corresponding proximal operator, an optimal convergence rate of $O\left(\frac{1}{k^2}\right)$ can be achieved. For problems such as overlapping group lasso where no closed-form solution is known for the proximal operator, the optimal convergence rate can only be guaranteed when the error of solving the proximal operator can be controlled at each iteration. For both FoGLasso and Picard-Nesterov, the complexity of the inner iteration for computing the proximal operator is $O(pg)$. This group of methods often work quite well when the proximal operator can be solved efficiently, while one disadvantage is that for a new class of problems, one needs to design a dedicated solver for computing the new proximal operator, which can be challenging for certain cases.

- **AGD with Nesterov's smoothing technique**, such as Prox-Grad. For non-smooth problems, the smoothing technique can guarantee a convergence rate of $O\left(\frac{1}{k}\right)$, with per iteration cost being $O(p^2+pg)$ [3]. One advantage of Prox-Grad is that it can be easily applied to a wide range of structured sparse learning models, including overlapping group lasso and graph induced lasso. However, Prox-Grad involves a smoothing parameter μ, which can affect the speed of the algorithm and needs to be tuned properly.

- **ADMM**. The worst case convergence rate of ADMM is $O\left(\frac{1}{\sqrt{k}}\right)$, and the actual speed of the implementation may rely on the choice of the penalty parameter ρ. In each iteration, ADMM solves a $p \times p$ linear system, which can be solved in $O(p^2)$ when the Cholesky decomposition of the matrix for the system can be pre-computed. Therefore, the per-iteration cost of ADMM is $O(p^2+pg)$. ADMM is known to work well in certain problems such as trace norm minimization [95].

Chapter 3

Multi-Source Learning with Incomplete Data

Starting from this chapter, I present three real world applications where intrinsic sparsity and structure in the data sets can be exploited to aid problem solving. I first consider the feature learning using block-wise incomplete data. In many applications, multiple data sources may suffer from a considerable amount of missing data. For example, in the ADNI data acquisition phase, many subjects lack a subset of measures, resulting in a scenario shown in Figure 3.1, where large chunks of missing data are marked by the white areas. A simple and popular approach is to remove all the subjects with missing values, but this greatly reduces the number of samples and fails to fully use the information in the data set. In Figure 3.1, only 79 subjects (Subjects 61-139) out of a total of 245 subjects do not have missing values. Next, I present two methods for dealing with multi-source data with block-wise missing values.

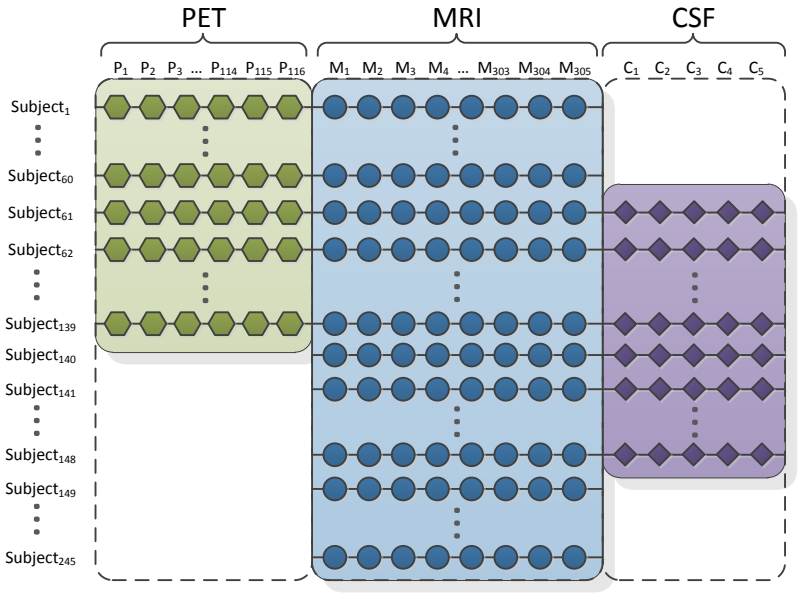


Figure 3.1: Illustration of the "block-wise" pattern of missing data for the ADNI data set. In this figure, I show AD and normal control subjects only. For simplicity, I focus on those subjects with complete MRI measures.

3.1 Proposed Method I: Incomplete Multi-Source Feature Learning

In my feature learning framework described below, I fully use the multiple heterogeneous data with a block-wise missing pattern by exploiting the underlying structure in the multi-source data. My proposed framework formulates the prediction problem as a multi-task learning problem [96–99] by first decomposing the prediction problem into a set of tasks, one for each combination of data sources available, and then building the models for all tasks simultaneously.

For example, considering a data set with three sources (CSF, MRI, PET) and assuming all samples have MRI measures, first I partition the samples into multiple blocks (4 in this case), one for each combination of data sources available: (1) PET, MRI; (2) PET, MRI, CSF; (3) MRI, CSF; and (4) MRI. Then I build four models, one for each block of data, resulting in four prediction tasks (Figure 3.2).

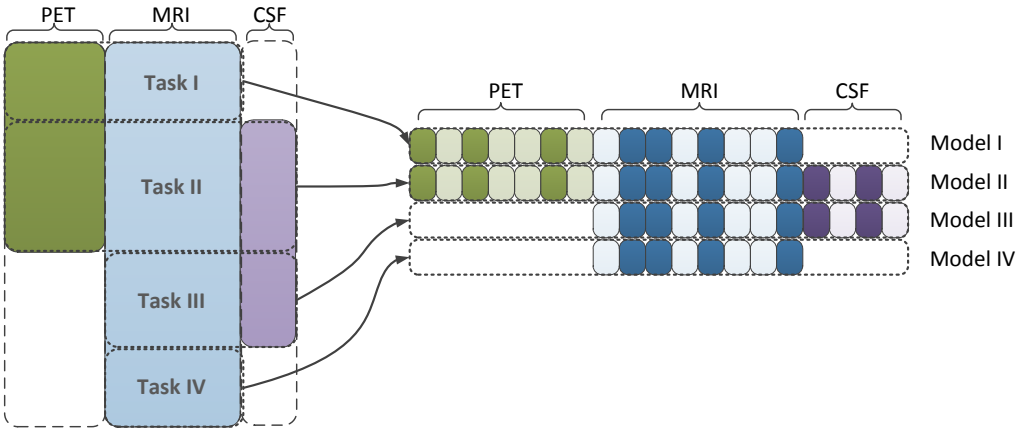


Figure 3.2: Illustration of the proposed multi-task feature learning framework for incomplete multi-source data fusion. In the proposed framework, first I partition the samples into multiple blocks (four blocks in this case), one for each combination of data sources available: (1) PET, MRI; (2) PET, MRI, CSF; (3) MRI, CSF; (4) MRI. Then I build four models, one for each block of data, resulting in four prediction tasks. I use a joint feature learning framework that learns all models simultaneously. Specifically, all models involving a specific source are constrained to select a common set of features for that particular source.

A simple approach to deal with the missing data is to build these four models separately, but that does not fully use the information in the multi-source data. Indeed, the sample size for each of these four tasks is even smaller, resulting in the *large dimension small sample size* problem. I address this by employing a joint feature learning formulation. I formulate our proposed framework as follows. Suppose the data set is divided into m tasks: $T^i = \{x_j^i, y_j^i\}$, $i = 1 \ldots m$, $j = 1 \ldots N_i$, where N_i is the number of subjects in the i-th task, and (x_j^i, y_j^i) is the j-th subject from the i-th task. For each task, I consider the following linear model:

$$f^i(x_j) = (\beta^i)^T x_j^i$$

where β^i is the weight vector, including the model parameters for the i-th task. Denote $\beta = \{\beta^1, \ldots, \beta^m\}$ as the collection of all model parameters. Assume that we have a total of S data sources, and the feature dimensionality of the s-th source is denoted as p_s. For notational convenience, I introduce an index function $I(s,k)$ as follows: $\beta_{I(s,k)}$ denotes all the model parameters corresponding to the k-th feature in the s-th data source. The proposed multi-task feature learning framework is:

$$\min_{\beta} \frac{1}{m} \sum_{i=1}^{m} \frac{1}{N_i} \sum_{j=1}^{N_i} L(x_j^i, y_j^i, \beta_i) + \lambda \sum_{s=1}^{S} \sum_{k=1}^{p_s} \left\| \beta_{I(s,k)} \right\|_2 \tag{3.1}$$

where $L(\cdot)$ is the loss function, and I adopt the logistic loss in my study. The second part of the formulation, which is essentially an $\ell_{2,1}$-norm regularization on the model parameters [100], leads to a solution with the desired sparsity, that is, all models involving a specific source are constrained to select a common set of features for this particular source. The proposed formulation is novel as it (1) formulates the incomplete multi-source fusion as a multi-task learning problem, and (2) extends existing multi-task feature learning formulations to accommodate missing feature values.

The regularization parameter λ in (3.1) controls the sparsity of the solution. Generally speaking, the larger λ is, the sparser the solution will be. However, in practice,

the same λ value will induce different sparsity for different data sets. To select a proper range of parameters, we follow a similar approach as discussed by Liu et al. [101] to obtain a value λ_{max} for each specific problem such that if $\lambda \geq \lambda_{max}$, the optimal solution to (3.1) is $\mathbf{0}$. Therefore, I just need to set a *ratio r* such that $\lambda = r\lambda_{max}$, and r is selected in the region $(0,1)$.

3.2 Proposed Method II: Model Score Completion

The iMSF framework proposed in Section 3.1 tackles the problem in a "row-wise" manner by dividing samples into different groups. Next I propose to tackle the problem in a "column-wise" manner. From Figure 3.1 we can observe another characteristic of this problem: if a certain data source is available for a particular sample (e.g., PET for subject 1), the complete set of features from this data source will be available. Intuitively, it is more challenging to estimate a complete block of missing values than a single value. This motivates me to design the model score completion scheme (ScoreComp). The overview of the ScoreComp is illustrated in Figure 3.3. Given an incomplete multi-source data set, first I train a base model on each individual data source using the available samples, and the base model is applied to produce prediction scores for the corresponding samples for this data source; thus each data source is represented by a single column of scores, and all data sources together are represented as a matrix of prediction scores with missing values. A missing value estimation method is then applied to obtain a complete set of model scores, which are treated as newly derived features to train our final classifier. That is, the prediction score from each data source is considered as a feature.

I formally describe our ScoreComp method as follows. Consider a labeled data set $\{\mathscr{D}_i, \mathscr{Y}_i\}$, $i = 1, \ldots, N$, with S incomplete data sources. Let the set $sc(i) \subset \{1, \ldots, S\}$ denote the available data sources for the i^{th} subject, such that $\mathscr{D}_i = \{X_i^s | s \in sc(i)\}$, where X_i^s is the feature vector of the i^{th} subject from the s^{th} source. The goal of the ScoreComp

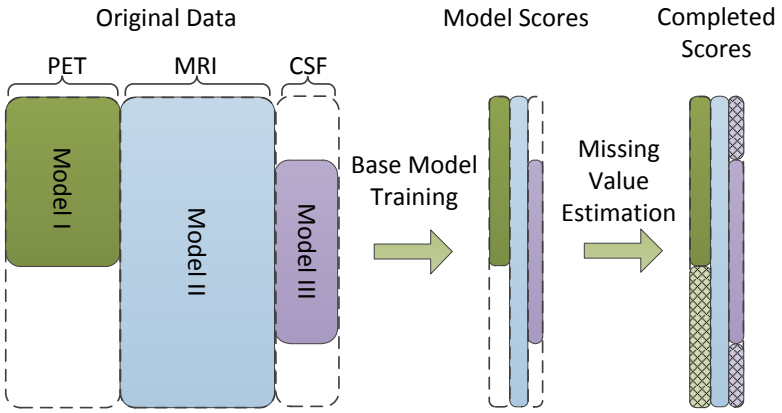


Figure 3.3: Illustration of the proposed model score completion scheme. First I train a base model on each individual data source using the available samples, and the base model is applied to produce prediction scores for this data source; thus each data source is represented by a single column of (incomplete) scores. A missing value estimation method is applied to obtain a complete set of model scores, which are treated as newly derived features to train our final classifier.

method is to derive a completed prediction score matrix $\tilde{A} \in \mathbb{R}^{N\times S}$ from the original data set. Details are given below:

Base Model Training Step. First I choose a classifier learning algorithm $\mathscr{L}$, based on which a prediction model is constructed for each data source:

$$\mathscr{M}_s = \mathscr{L}\left(\{(X_i^s, \mathscr{Y}_i) | s \in sc(i)\}\right), \quad s = 1, \ldots S.$$

Then, I use these models to construct an incomplete prediction score matrix $\hat{A} \in \mathbb{R}^{N\times S}$ given by:

$$\hat{A}_{i,s} = \begin{cases} \mathscr{M}_s(X_i^s) & \text{if } s \in sc(i) \\ \text{NaN} & \text{otherwise} \end{cases},$$

where $\mathscr{M}_s(X_i^s)$ is the prediction score of model $\mathscr{M}_s$ on feature vector X_i^s.

Missing Value Estimation Step. In this step, I choose a missing value estimation algorithm $\mathscr{E}$ such that $\tilde{A} = \mathscr{E}(\hat{A})$, where $\tilde{A}$ is the completed prediction score matrix. $\tilde{A}$ is then treated as the derived feature matrix for the original data set $\{\mathscr{D}_i, \mathscr{Y}_i\}$. The final model $\mathscr{M}$ is learned using $(\tilde{A}, \mathscr{Y})$ so that the data sources are integrated.

Prediction of Unlabeled Sample. Suppose we are given a set of unlabeled data $\{\mathscr{U}_j\}$, $j = 1,\ldots,M$, such that $\mathscr{U}_j = \{X_j^s | s \in sc(j)\}$. ScoreComp will first derive a completed feature matrix $\tilde{B} \in \mathbb{R}^{M\times S}$, which is then fed into $\mathscr{M}$ for prediction. To obtain $\tilde{B}$, first I use the models learned in the base model training step to construct an incomplete model score matrix $\hat{B} \in \mathbb{R}^{M\times S}$ given by:

$$\hat{B}_{j,s} = \begin{cases} \mathscr{M}_s(X_j^s) & \text{if } s \in sc(j) \\ \text{NaN} & \text{otherwise} \end{cases}.$$

Combining this with the previously obtained complete matrix $\tilde{A}$, I obtain:

$$C = \mathscr{E}\left(\begin{bmatrix}\tilde{A}\\ \hat{B}\end{bmatrix}\right), \quad C \in \mathbb{R}^{(N+M)\times S}.$$

Finally, by extracting the lower M rows of matrix C, I can obtain the derived feature matrix $\tilde{B}$ for the unlabeled data set.

Like the iMSF method, all available information is used in the integration process. Note that in ScoreComp, I still estimate missing values; but instead of estimating blocks of missing data, I only need to impute the prediction scores. Another advantage of this framework is its simplicity. No additional parameters are introduced, and one can choose any classification algorithms and/or missing value estimation method that suit the data set at hand. In this study, I chose the random forest classifier [102] as the base model learning algorithm $\mathscr{L}$, and my final model $\mathscr{M}$ was trained using ridge regression. I also tried various missing data estimation methods, and more details may be found in Section 3.3.

3.3 Comparison of iMSF, ScoreComp and imputation methods

I apply the proposed methods to the full multi-source data set including MRI, PET, proteomics and CSF for solving clinical group classification problems (AD vs. Normal; AD vs. Non-converter and Converter vs. Normal). 780 subjects were analyzed. Among them, each subject has at least one of the four data sources (MRI, FDG-PET, CSF and

proteomics features) available. I first randomly select a portion (from 50% to 75%) of samples as the training set to learn the model, and then apply the model to predict the labels on the remaining data, used as a non-overlapping test set. I repeated this process 30 times; the average performance is reported.

Table 3.1: Classification performance comparison of the proposed iMSF, ScoreComp and missing value estimation methods (Zero, EM, KNN, SVD and SVT) in terms of accuracy, sensitivity and specificity when the training percentage varies from 50% to 75%.

	AD vs. Normal			AD vs. Non-converter			Converter vs. Normal		
Training Size 50%	Accuracy	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity
iMSF-Best	86.37%	85.64%	**87.65%**	78.05%	**57.34%**	89.91%	**91.20%**	94.11%	**89.59%**
iMSF-Average	83.00%	81.56%	84.80%	77.51%	55.97%	88.64%	90.34%	93.31%	84.18%
ScoreComp-Zero	87.31%	87.61%	87.35%	81.25%	53.26%	**95.68%**	83.15%	**99.32%**	48.12%
ScoreComp-EM	89.74%	94.13%	85.32%	81.05%	53.35%	95.31%	88.99%	98.73%	68.30%
ScoreComp-KNN	**90.13%**	**94.87%**	85.37%	**81.46%**	56.02%	94.58%	87.60%	98.89%	63.49%
Zero	85.38%	87.74%	83.23%	77.25%	44.10%	94.14%	89.48%	97.93%	71.36%
EM	87.20%	88.76%	85.75%	78.11%	47.25%	93.80%	89.40%	97.97%	70.95%
KNN	85.79%	88.10%	83.66%	75.15%	36.78%	94.59%	86.40%	97.88%	61.41%
SVD	83.89%	84.94%	82.96%	75.73%	43.00%	92.37%	87.57%	97.09%	67.07%
SVT	55.75%	61.31%	50.70%	81.15%	56.82%	93.48%	71.69%	87.81%	36.34%
Training Size 66.7%	Accuracy	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity
iMSF-Best	88.77%	86.77%	**91.25%**	80.55%	**65.83%**	89.03%	**93.17%**	95.36%	**90.52%**
iMSF-Average	86.07%	83.23%	89.23%	80.12%	63.70%	88.16%	92.40%	94.72%	87.37%
ScoreComp-Zero	88.18%	86.48%	90.10%	**83.48%**	59.90%	**94.96%**	84.48%	**99.53%**	50.13%
ScoreComp-EM	**90.28%**	93.64%	86.61%	83.25%	60.85%	94.12%	90.82%	99.16%	72.07%
ScoreComp-KNN	90.19%	**94.48%**	85.51%	83.30%	62.25%	93.56%	89.56%	99.24%	67.66%
Zero	86.79%	88.80%	84.61%	80.87%	52.21%	94.56%	92.30%	98.84%	77.32%
EM	88.46%	88.40%	88.52%	80.90%	53.28%	93.96%	91.69%	98.76%	75.75%
KNN	87.20%	87.60%	86.75%	78.46%	44.24%	94.75%	89.07%	98.68%	66.84%
SVD	85.25%	85.19%	85.36%	78.52%	48.89%	92.51%	90.44%	98.13%	72.75%
SVT	57.14%	61.08%	53.39%	83.25%	60.71%	94.01%	75.30%	91.66%	38.03%
Training Size 75%	Accuracy	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity
iMSF-Best	88.79%	87.16%	**90.92%**	81.33%	**66.02%**	89.06%	**93.19%**	95.94%	**90.58%**
iMSF-Average	85.81%	82.77%	89.38%	80.78%	65.20%	88.70%	92.98%	95.03%	88.87%
ScoreComp-Zero	88.53%	87.67%	89.75%	**84.51%**	61.48%	**96.00%**	84.96%	**99.60%**	52.74%
ScoreComp-EM	89.91%	92.97%	86.77%	83.80%	61.24%	94.95%	89.93%	99.30%	70.21%
ScoreComp-KNN	**90.35%**	**93.89%**	86.74%	84.35%	63.79%	94.60%	88.89%	99.11%	66.91%
Zero	86.49%	88.06%	85.13%	81.76%	55.54%	94.74%	91.48%	97.68%	78.49%
EM	88.96%	88.54%	89.54%	81.25%	55.72%	93.83%	90.74%	98.44%	74.38%
KNN	87.32%	88.20%	86.66%	77.73%	44.97%	93.66%	88.89%	98.43%	68.12%
SVD	85.93%	85.78%	86.23%	78.55%	50.13%	92.29%	89.19%	97.48%	70.92%
SVT	56.58%	59.38%	53.73%	81.92%	56.93%	94.17%	75.56%	90.23%	42.59%

For comparison purposes, I included the following missing value estimation methods:

Zero: this is the most intuitive way to impute missing values - we assign zero to any element that is missing. When the data set is first normalized to have zero mean and unit standard deviation, this is equivalent to mean value imputation.

KNN: missing value imputation using the k-nearest neighbor method [43]. The KNN method replaces the missing value in the data matrix with the corresponding value from the nearest column. That is to say, KNN will first identify the most similar feature to the current one with a missing value, and then use this feature as a guess for the missing one.

EM: this method imputes missing values using the expectation-maximization (EM) algorithm [44]. An iteration of the EM algorithm includes two steps. In the E step, we estimate the mean and covariance matrix from the data matrix (with missing values filled with guesses from previous M step, or initialized as zeros); then in the M step, the missing value of each data column is filled in with their conditional expectation values based on the available values and the estimated mean and the covariance. We then re-estimate the mean and the variance based on the new estimates, therefore entering the next EM iteration.

SVD: this is a standard method for matrix completion based on a low rank approximation. The SVD based estimation works in a similar way to the EM method above. We first provide some initial guesses (such as 0) to the missing data values, and then we apply singular value decomposition (SVD) to obtain a low-rank approximation of the filled-in matrix. Next, we update the missing values using their corresponding values in the low-rank estimation. Finally, we apply SVD to the updated matrix again and the process is repeated until convergence.

SVT: Recently, trace norm minimization has been proposed for missing data estimation [47, 48]. This can be effective even when a large amount of data is missing. Therefore, it will be interesting to see how this algorithm (singular value thresholding or SVT) performs in our particular setting. We acquire the SVT program online (`http://svt.stanford.edu`) and follow their suggestions for parameter setting.

The classification results are summarized in Table 3.1. For my proposed iMSF method, five ratios (0.001, 0.01, 0.1, 0.2 and 0.4) were used for the regularization parameter λ. First I run our iMSF method to learn a set of features for each different task, and then train a random forest classifier [102] on each learning task using the selected features. The best and average performance obtained using these five ratios are reported in Table 3.1, as "iMSF-Best" and "iMSF-Average" respectively. For the proposed ScoreComp scheme, first I use random forest to obtain the prediction score for the incomplete data set, and apply three different missing value estimation methods (Zero, EM and KNN) to obtain the completed score matrix. Then, ridge regression is applied to integrate the data sources together. The performance using three different missing value estimation methods is reported in Table 3.1 as "ScoreComp-Zero", "ScoreComp-EM" and "ScoreComp-KNN" respectively.

From Table 3.1, we can observe that for the AD vs. Normal problem, my ScoreComp performs best, with about 90% accuracy. The top 2 methods in terms of performance are "ScoreComp-EM" and "ScoreComp-KNN" in all training ratios, and they tend to produce comparable results. The AD vs. Normal problem is considered to be less challenging, and this might explain why its simplicity may give ScoreComp method an edge over iMSF.

In the more challenging settings where MCI subjects are involved, I obtain low sensitivity in the AD vs. Non-converter case and low specificity in the Converter vs. Normal case. In these settings, iMSF performs much better – it provides more balanced

classification results on both the positive and negative classes. For example, in the Converter vs. Normal problem, using 75% of data for training, missing value estimation based methods only achieve about 70% specificity, but our proposed iMSF achieves an average of 89% among the five different parameters. Therefore, even though ScoreComp achieves a higher accuracy in the AD vs. Non-converter case, I still consider iMSF as the best method for these settings. This may be due to the fact that our iMSF algorithm took a more systematic approach in using multiple sources of information for classification.

Among different variations of the ScoreComp method illustrated in Table 3.1, "ScoreComp-EM" and "ScoreComp-KNN" produce more stable results than "ScoreComp-Zero", where the missing scores are substituted with zeros. For example, in the AD vs. Non-converter case, all three variations yield comparable performance, while in the Converter vs. Normal case, "ScoreComp-Zero" yields much lower specificity. We can conclude that in the missing value estimation step of our ScoreComp method, an effective algorithm can indeed enhance the classification performance by using the model scores from other samples. Interestingly, the five other different missing value estimation methods (Zero, EM, KNN, SVD and SVT) perform comparably to each other. Thus, estimating the block-wise missing values directly does not give much edge over simply substituting missing elements with zeros. This further justifies the effectiveness of my ScoreComp method.

Chapter 4

Learning Sparse Representations for Fruit-Fly Gene Expression Pattern Image Annotation and Retrieval

In this chapter, I adapt advanced data mining and computer vision techniques to address the key challenges in annotating and retrieving fruit fly gene expression pattern images. To boost the performance of image annotation and retrieval, I propose representations integrating spatial information and sparse features, overcoming the limitations of prior schemes.

4.1 The Bag-of-Words Approach

The bag-of-words method was originally used for text classification problems where each document is represented as a feature vector indicating the frequency of each word in the document. Such feature vector representation is used to classify documents into one or more categories. This text categorization approach has been adapted to image analysis [103]. Specifically, images are represented as a collection of "visual words", based on features extracted from the images [104].

In the BoW approach for image representation, invariant visual features are usually extracted from a subset of images [104] to produce a visual codebook using a clustering algorithm, though a recent study shows that the clustering process is not really essential [105]. Here the cluster centers are considered to be visual words. From this codebook, each feature from an image patch is quantized to the closest visual word in the codebook. A histogram is then created to represent the number of occurrences of each word located in an image. This histogram is a global representation because it only tracks the number of occurrences of each word in an image but not the location of those words, thereby the spatial layout of local image features is not captured. This is considered as one

of the major drawbacks of the BoW model [67]. Next, I discuss each step involved in the BoW model when applied to fruit fly images in details.

Feature Detection

Feature detection involves locating regions in an image to serve as representative boundaries for visual words. The images in this study have been properly scaled and aligned semi-automatically. I use a series of overlapping circles to represent areas where feature information is extracted to construct a single visual word. An example of these overlapping circles is shown in Figure 4.1. In my experiments, the radius of the patches are set to 16.

Feature Description

Based on the regions described above, a local feature is extracted from each of the overlapping circle. Because of its robustness against variations in image scale and rotation, I use the scale-invariant feature transform (SIFT) descriptor [106] for representing each local patch. Thus, each image consists of a collection of feature vectors.

Codebook Generation

The codebook is constructed by obtaining a collection of representative vectors from the extracted features. I use the common generation approach of selecting a subset of images and then using the k-means algorithm to cluster their SIFT feature vectors [107]. The number of cluster centers which represent the visual words can be set manually. For the image annotation and retrieval problem, I set this number to 2000. The SIFT feature vectors can then be quantized to the closest codebook centers in order to form a visual word representation for each image.

Once the codebook has been created, I assign codebook words to features extracted from image patches. Formally, assume the number of patches (feature vectors) for a given image is I and the size of the codebook is J. Define $e_{ij} = 1$ if the i^{th} feature

vector is assigned to the j^{th} codeword, and 0 otherwise. Then the given image can be represented as $H = [h_1, h_2, \ldots, h_J]$ where

$$h_j = \sum_{i=1}^{I} e_{ij}. \tag{4.1}$$

4.2 The Spatial Bag-of-Words Approach

A major limitation of the BoW approach is that the spatial information of local image features is not encoded, as the bag-of-words representation is an un-ordered collection of visual words. A previous study on a bag-of-words approach [67] for automated annotation of *Drosophila* embryo image groups showed encouraging results, and a recent study [69] showed that using spatial information together with visual information is better than using only visual information. I expect the performance can be further improved by taking advantage of the spatial information, i.e., the location where visual words are found within images. Intuitively, the additional spatial information of visual words within images may facilitate the classification of images when the discriminant features are restricted to a certain region, which is the case for our CV terms. This can be implemented by adopting a method similar to the spatial pyramid matching scheme [108].

My approach for image representation is based on an implementation of the spatial bag-of-words method. Like the BoW method, the spatial BoW method creates a histogram for each image, counting the number of times each word appears in an image. Additionally, the spatial BoW tracks the position where each visual word is located. Therefore, the spatial BoW method benefits from the robustness of the BoW method while also taking advantage of the spatial properties of images.

A spatial bag-of-words is much like a normal bag-of-words except that it is represented by a larger feature vector. While a histogram of an image is represented by a non-spatial bag-of-words, H, a spatial bag-of-words consists of multiple non-spatial bags, concatenated. Specifically, for each image with n spatial sections, a spatial bag M_n can be

represented as $M_n = [H_1, H_2, \ldots, H_n]$, where each H_i corresponds to a non-spatial bag-of-words for a particular spatial section. Thus I have n bags-of-words from n spatial sections on each image that are concatenated to form M_n. This way, different sections of a spatial vector represent different sections of an image. My automated annotation representation is created by partitioning feature patches into 3 by 6 sections on each image. This representation creates a multiple of 18 in added dimensionality to a non-spatial representation of the same visual words. For each image group in the study we also create a global bag-of-words representation to test the differences in annotation performance that are seen between the global and the spatial approaches. Figure 4.1 shows a global bag-of-words representation, a 2 by 2 spatial BoW representation, and a 4 by 4 spatial BoW representation below the circular feature representations of two separate images.

4.3 The Sparse Spatial Representation

The original BoW representation, as applied to image analysis, assigns each feature vector to the closest visual word in the dictionary. Denote the feature vector obtained for a given patch as $y \in R^d$ and the dictionary matrix as $D \in R^{d \times c}$, in which each column is a centroid (visual word). Then, the assignment of an image patch to a visual word can be written formally as the following optimization problem:

$$\begin{aligned} \min_{e} \quad & \frac{1}{2}\|De - y\|_2^2 \\ \text{s.t.} \quad & e_i \in \{0,1\}, \quad \sum_{i=1}^{c} e_i = 1 \end{aligned} \tag{4.2}$$

Clearly, the constraints enforce that only one element in the solution e will be set to one, which corresponds to the visual word most similar to the image patch y. In this case, relationships between a feature vector and other visual words are discarded. This would not be a problem if a feature vector is an exact match with the visual word that it is assigned to, as in the case of text classification. However for images, a feature vector may

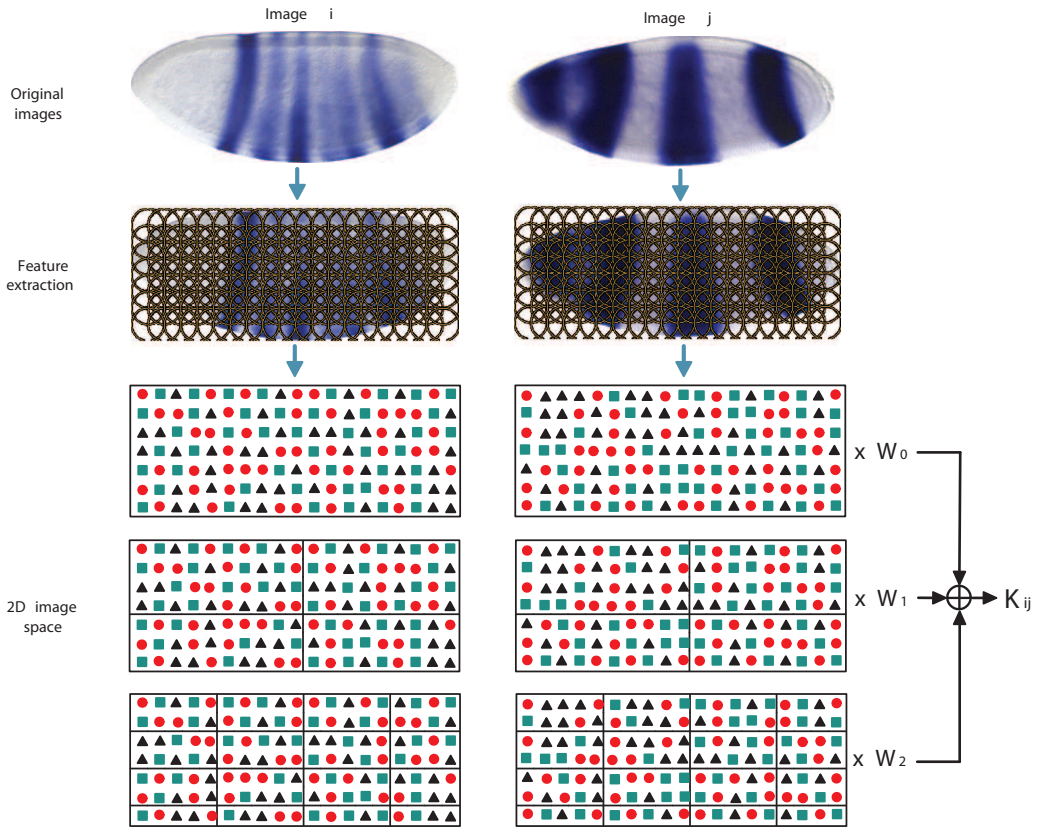


Figure 4.1: Illustration of image patch extraction and the three levels of bag-of-words partitioning with weighting factors for the spatial pyramid approach. After feature description using the overlapping circular patches, three levels of bag-of-words partitioning are shown. The top level of partitioning is just a global bag-of-words representation.

be close to multiple visual words. In such cases, the relationship with the closest word would be overestimated and the relationships with the other similar words would be lost, leading to degenerated representation accuracy.

The sparse approach for BoW representation addresses this problem by assigning feature vectors to multiple visual words simultaneously. I seek to represent the local patch using "visual sentence" with a *set* of "words" instead of a single one. Besides the selection of visual words to form this sentence, I also need to evaluate the "contributions". A commonly used approach is to formulate this problem as a sparse learning problem, which can be solved by state-of-the-art algorithms.

Mathematically, the generalization from visual word to visual sentence can be done by relaxing the constraint in (4.2). I construct the representation vector $x \in R^c$, such that for the i^{th} entry, $i = 1, \ldots, c$, $x_i = w_i$ when the i^{th} keyword is selected with contribution w_i, and 0 when the keyword is not selected.

In order to make x sparse (contains multiple 0 entries), an ℓ_1 regularization is imposed, resulting in the following optimization problem:

$$\begin{aligned} \min_{x} \quad & \|Dx - y\|_2 + \lambda |x|_1 \\ \text{s.t.} \quad & x_i \geq 0, \quad i = 1, \ldots, c \end{aligned}$$

In which $|\cdot|_1$ is the ℓ_1 norm and λ is a parameter that controls the sparsity. In our experiments, λ is fixed to be 0.01. This problem is closely related to *LASSO* [4], and can be solved by many existing software packages, such as SLEP [87].

The comparison between "visual word" and "visual sentence" for image representation is illustrated in Figure 4.2. As shown in the figure, the sparse learning provides more smooth representation.

Integrating the spatial and sparse approaches into the BoW representation model is therefore expected to produce a more accurate description of *Drosophila* images. I have created both sparse and non-sparse versions of both my global and spatial bag-of-words representations, and compare different combinations of approaches for image annotation and retrieval. Detailed performance evaluation can be found in the results section.

4.4 Data Description

The *Drosophila* gene expression pattern images used in my study are obtained from the FlyExpress database, which contains standardized images obtained from the Berkeley *Drosophila* Genome Project (BDGP). In BDGP, the *Drosophila* embryogenesis is divided into six stage ranges (1-3, 4-6, 7-8, 9-10, 11-12, 13-16). The first stage range is not included in this study because of the small number of CV terms used to describe its

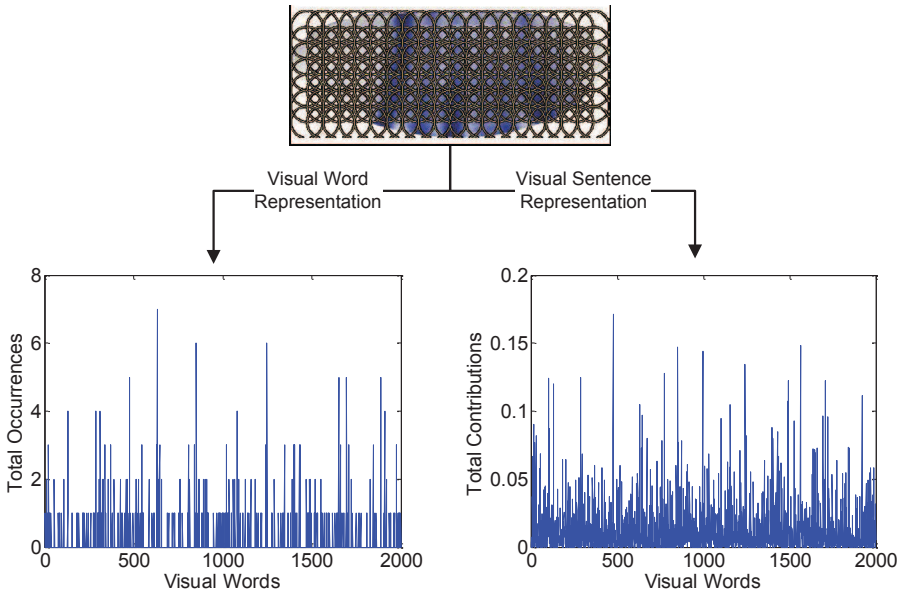


Figure 4.2: Different histogram representation obtained for a given image. The histogram on the left is obtained by assigning each local patch to a single visual word, while the one on the right is obtained by applying the sparse learning formula to select a set of visual words for each patch.

images. Images from the remaining stage ranges are annotated separately in their respective groups because the majority of terms are stage range specific. The second through sixth stage ranges consist of 1081, 877, 1072, 2113, and 2816 image groups, respectively. The last two stage ranges contain the largest number of lateral images as well as the highest counts of CV terms.

4.5 Evaluation of Developmental Term Annotation Performance

I employ the one-against-rest support vector machines (SVM) to annotate the gene expression pattern images, where the SVM builds a decision boundary between image groups that contain a particular term and the remaining image groups. I employ the LIBSVM package [109] and the linear kernel is used. The regularization parameter is set to 1 in all cases. My proposed method combines both the spatial and sparse approaches and is denoted by $\text{SVM}_{\text{Spatial+Sparse}}$. I compare my method with those that utilize only sparse, only spatial, or global bag-of-words approaches. These approaches are denoted by

SVM_{Sparse}, $SVM_{Spatial}$, and SVM_{Global}, respectively. The performance comparison of the four representations in terms of AUC and macro F1 scores is summarized in Tables 4.1 and 4.2, respectively.

Since most CV terms are stage-range specific, I annotate the image groups according to their stage ranges separately. For each stage range, I begin with the 10 terms that appear most frequently, and then we add additional terms in the order of their frequencies with a step size of 10. This results in different numbers of data sets in each stage range, depending on the total number of CV terms in that stage range. The extracted data sets are randomly partitioned into disjoint training and testing sets using the ratio 1:1 for each term. For each data set, I generate 30 random partitions and the average performance is reported. Because my method models each individual term separately, I can compare the results of our method against the results of the other method on a term-by-term basis. For example, I can compare annotation results of our method with the non-spatial method in stage range 13-16, term by term, where 40 CV terms are used. In this comparison, of the 40 terms being studied, 39 saw an average increased AUC performance and 31 saw average increased F1 Score (F1) performance. Due to space limitation, I will not show each individual term by term comparison. Instead, I show the results for each stage range where various numbers of CV terms are used.

Table 4.1 shows a comparison of AUC results for all four methods discussed. The best results for each case are highlighted in bold. The results show that both the spatial and the sparse methods consistently outperform the non-spatial method in terms of average AUC. The results also show that combining both sparse and spatial approaches outperforms any of the other three methods. The results indicate that the sparse approach offers improved performance over the spatial approach for the earlier stage ranges, and that the two approaches are comparable for the last stage range. The poorer performance of the spatial approach for the earlier stages may have been due to the less developed

embryonic structures found earlier in embryogenesis. Combining the spatial and sparse approaches resulted in the best results, particularly in the later stage ranges.

Table 4.2 shows a similar type of comparison as in Table 4.1. The only difference is that F1 score is used as a comparison measure instead of AUC. I observe a similar trend: both the spatial and sparse methods outperform the global approach; the sparse approach performs slightly better than the spatial approach in the earlier stages, and they achieve similar performance during the last stage. Again, we can observe that combining the sparse and spatial approaches generates better results than using sparse or spatial information alone.

I have observed that there were significant differences in performance increases between earlier stage ranges where Drosophila embryos were less developed and later stage ranges where embryos were more developed. We also observe that there are certain terms that benefit far greater from a spatial bag-of-words approach than other terms. For example, *mesectoderm anlage in statu nascendi*, *central brain anlage*, *crystal cell specific anlage*, *hypopharynx primordium P2*, *procrystal cell*, and *crystal cell* are all stage dependent terms that showed the most dramatic increases in annotation performance. These increases in performance are consistent across multiple stage range tests, where the number of terms being annotated varied. There are also a number of terms such as *pole cell*, *mesectoderm primordium*, *foregut primordium*, *germ cell*, *embryonic central brain neuron*, *embryonic central brain glia*, and *lateral cord glia* that showed good performance across multiple stage ranges, where various numbers of CV terms were annotated.

There are pioneering works on constructing feature representations for Drosophlia gene expression image annotation. Zhou et al. [110] applied multi-resolution 2D wavelet discrete transform followed by min-Redundancy max-Relevance feature selection. Puniyani et al. [60] proposed an automatic system named "SPEX^2" that performs pattern extraction using Markov random field and further extracts features using the SIFT

descriptor and singular value decomposition. Using the top 10 most frequent terms [60] in the BDGP data set, Zhou's system achieved an average F1 score of about **0.35**, while Puniyani's method achieved about **0.45**. For comparison purposes, I extract the individual F1-scores for the same terms. My Sparse + Spatial representation yields an average F1-score of **0.64**, which outperforms both methods.

Table 4.1: Comparison of different annotation methods in terms of AUC.

Stage range	# of terms	$SVM_{Spatial+Sparse}$	SVM_{Sparse}	$SVM_{Spatial}$	SVM_{Global}
4-6	10	**.8284 ± .0321**	.8250 ± .0319	.8064 ± .0321	.7984 ± .0320
	20	**.8310 ± .0286**	.8240 ± .0293	.8046 ± .0292	.7965 ± .0302
	30	**.7982 ± .0408**	.7892 ± .0399	.7777 ± .0400	.7635 ± .0405
7-8	10	**.7808 ± .0285**	.7685 ± .0297	.7567 ± .0301	.7472 ± .0293
	20	**.7734 ± .0431**	.7619 ± .0427	.7444 ± .0496	.7309 ± .0484
9-10	10	**.7917 ± .0260**	.7816 ± .0270	.7652 ± .0264	.7538 ± .0265
	20	**.7971 ± .0335**	.7829 ± .0344	.7706 ± .0344	.7476 ± .0349
11-12	10	**.8526 ± .0248**	.8478 ± .0249	.8316 ± .0243	.8257 ± .0240
	20	**.8574 ± .0206**	.8437 ± .0214	.8275 ± .0215	.8091 ± .0228
	30	**.8275 ± .0252**	.8085 ± .0254	.7940 ± .0274	.7673 ± .0268
	40	**.8193 ± .0290**	.7991 ± .0306	.7810 ± .0304	.7560 ± .0321
	50	**.8084 ± .0351**	.7894 ± .0363	.7648 ± .0370	.7426 ± .0382
13-16	10	**.8807 ± .0221**	.8659 ± .0223	.8632 ± .0218	.8398 ± .0225
	20	**.8504 ± .0172**	.8301 ± .0182	.8304 ± .0180	.8001 ± .0177
	30	**.8344 ± .0197**	.8089 ± .0198	.8066 ± .0190	.7713 ± .0198
	40	**.8175 ± .0196**	.7892 ± .0208	.7847 ± .0211	.7496 ± .0223
	50	**.8038 ± .0249**	.7748 ± .0208	.7672 ± .0261	.7340 ± .0271
	60	**.7947 ± .0282**	.7657 ± .0299	.7613 ± .0300	.7281 ± .0310

Table 4.2: Comparison of different annotation methods in terms of macro F1.

Stage range	# of terms	$SVM_{Spatial+Sparse}$	SVM_{Sparse}	$SVM_{Spatial}$	SVM_{Global}
4-6	10	**.5224 ± .0407**	.5094 ± .0393	.4926 ± .0414	.4767 ± .0386
	20	**.4454 ± .0461**	.4200 ± .0462	.4141 ± .0459	.3794 ± .0412
	30	**.3459 ± .0593**	.3230 ± .0516	.3153 ± .0565	.2942 ± .0479
7-8	10	**.5372 ± .0343**	.5282 ± .0312	.5131 ± .0329	.5055 ± .0329
	20	**.3653 ± .0517**	.3603 ± .0538	.3331 ± .0740	.3364 ± .0676
9-10	10	**.5561 ± .0282**	.5499 ± .0276	.5353 ± .0289	.5267 ± .0260
	20	**.3836 ± .0464**	.3764 ± .0442	.3527 ± .0370	.3429 ± .0342
11-12	10	**.6339 ± .0280**	.6261 ± .0269	.6109 ± .0271	.6060 ± .0257
	20	**.5226 ± .0379**	.4961 ± .0310	.4781 ± .0337	.4508 ± .0290
	30	**.4066 ± .0409**	.3761 ± .0310	.3488 ± .0400	.3373 ± .0300
	40	**.3351 ± .0480**	.3110 ± .0383	.2686 ± .0456	.2762 ± .0358
	50	**.2758 ± .0480**	.2626 ± .0404	.2343 ± .0434	.2293 ± .0370
13-16	10	**.6506 ± .0297**	.6310 ± .0272	.6273 ± .0261	.5993 ± .0253
	20	**.5240 ± .0280**	.4959 ± .0262	.4963 ± .0266	.4580 ± .0245
	30	**.4474 ± .0303**	.4115 ± .0262	.4089 ± .0275	.3692 ± .0243
	40	**.3876 ± .0340**	.3487 ± .0268	.3408 ± .0319	.3071 ± .0252
	50	**.3330 ± .0381**	.2981 ± .0281	.2764 ± .0347	.2607 ± .0263
	60	**.2886 ± .0434**	.2598 ± .0317	.2313 ± .0373	.2255 ± .0287

4.6 Evaluation of Retrieval Performance

Based on the proposed image representations, I obtain the pair-wise similarity for every two images in the database, which can be used for image retrieval. In my study, the representative images for different views and stage ranges from the well-known Interactive Fly website[1] are used as queries. Then, for a given method and a query image, I select 8 images with the highest similarity values to obtain a set of query results. Note that the query images are removed from the results since they are always the one with highest similarity. Sample query results from different views and stage ranges are presented in Figures 4.3 and 4.4.

First, I will compare different methods by visually inspecting the images retrieved for each query. The first conclusion we can draw from the figures is that the methods based on the bag-of-words (the first three columns) generally outperform the one that utilizes the binary representation only. For example, for the stripe patterns such as those in Figures 4.3, the BFV method retrieves less than 4 similar images in its top 8 matches.

I also give brief interpretations of the retrieved images by analyzing the functions of the corresponding genes in the biological process annotated in the gene ontology[2]. Figure 4.3 shows a pattern expressed by gene *slp1*, during stage range 9-10. As we can see, all of the three "visual sentence" based approaches retrieved 6 images with *slp1* expressed. The rest of the genes retrieved, such as *slp2* which is involved in periodic partitioning, and *en* which is associated with the head segmentation process, are all closely connected to the blastoderm segmentation controlled by *slp1*.

Figure 4.4 illustrates a pattern expressed by gene *Gasp*, during stage range 13-16, taken from the lateral view. The spatial and sparse representation retrieves 4 images with the same gene, compared to 2 images by spatial BoW and 1 image obtained by BFV. *Gasp*

[1] http://www.sdbonline.org/fly/aimain/1aahome.htm
[2] http://www.geneontology.org/

as well as *CG13676* is involved in the chitin metabolic process. Another gene, *Idgf2*, which is related to the chitin catabolic process, is also closely related. The *trh* gene, which affects the epithelial cell fate determination and open tracheal system, is also related because chitin regulates epithelial tube morphogenesis; in addition to its classical role, protecting mature epithelia.

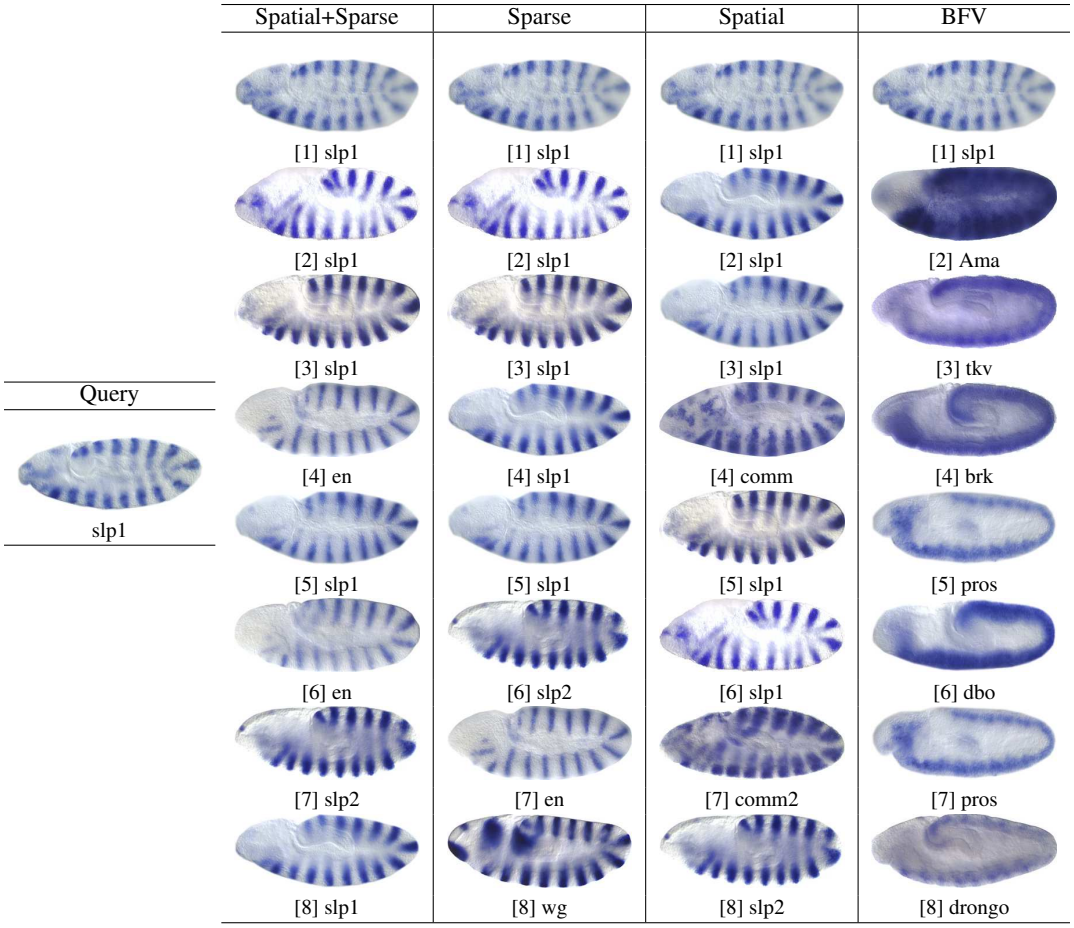


Figure 4.3: Retrieval results for query image ID insitu16633 with the lateral view in stage range 9-10.

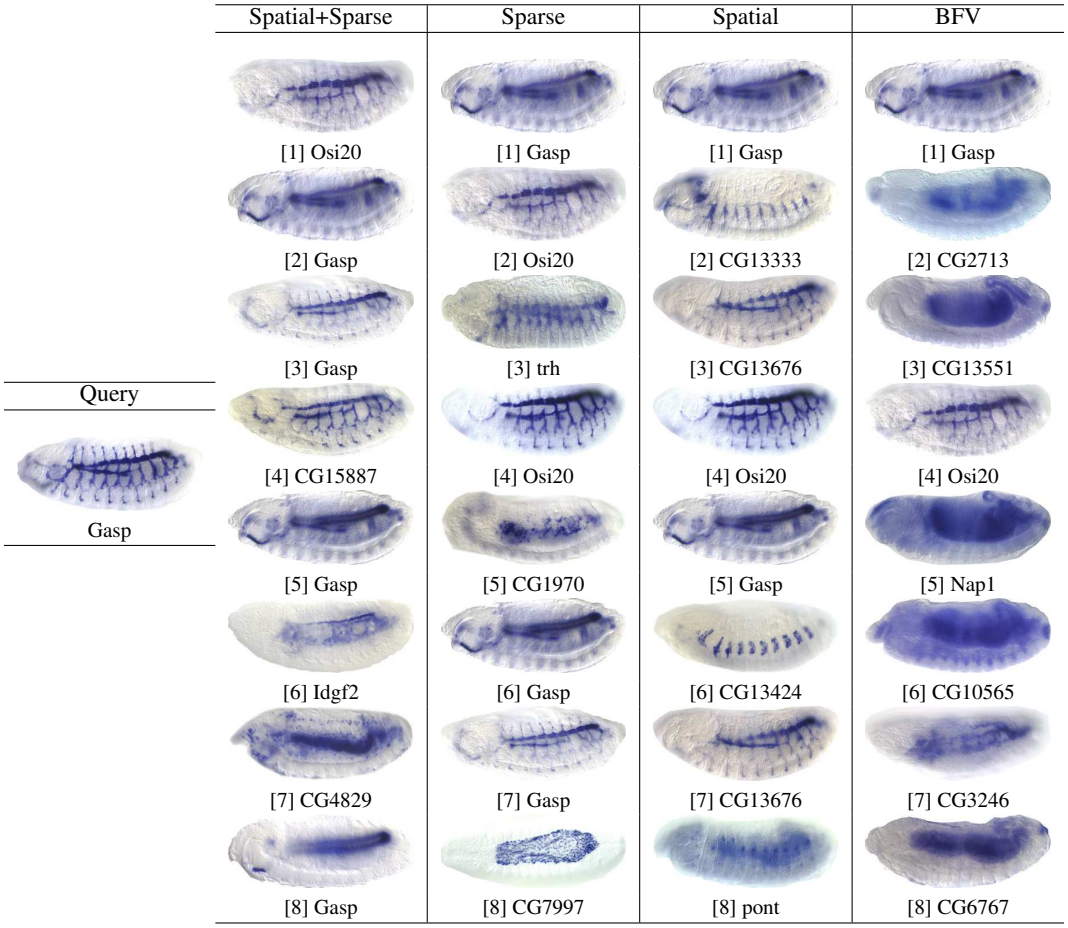


Figure 4.4: Retrieval results for query image ID insitu23837 with the lateral view in stage range 13-16.

Chapter 5

Automated Annotation of Developmental Stages of *Drosophila* Embryos in Images Containing Spatial Patterns of Expression

To develop an automated annotation system, I began by building a comprehensive training set, in which development experts identified images that were exemplar for each developmental stage defined in [75]. This constituted our initial training/testing set and contained 3,724 images such that there were over 200 images for each stage considered (Table 5.1). I applied machine learning [111] to develop a pool of 1,050 classification models to discriminate among stages (see details in Materials and Methods section). For any image, all 1,050 models are applied to generate a stage prediction, which produces the voting histogram (Figure 5.1). This histogram is used to generate estimates of embryo stage annotation at various levels of granularity. In the simplest case, I classify an embryo to be of stage S if a majority of models designated the image to be in stage S. For example, stage 10 gets the highest number of votes and thus assigned to the embryo in the image under consideration (Figure 5.1). This histogram also shows that the votes for stage 9 are higher than that for stage 11, which enables a finer stage designation (Early stage 10, 10E) for this embryo. I also generate a stage score (SS) using the frequencies in the voting histogram in order to incorporate non-symmetry of the distribution and relative size of the most frequent peaks (see Material and Methods). For the example in Figure 5.1, SS = 6.8. These stage scores can be used to order images based on embryonic developmental time or produce finer grade stage annotations (see later).

5.1 Results and Discussion

I estimated the cross-validation performance of the annotation system in correctly assigning a specific stage (S) for the training images first. This produced an accuracy of 79%, with the highest accuracy observed for stage 7 (89%) and the lowest accuracy for

Table 5.1: Number of Annotated BDGP Images for each developmental stage. This collection of images are manually annotated with precise stage labels. The orientation of all images in this study is standardized, and the size is scaled to 128 by 320 pixels.

Stage	1-3	4	5	6	7	8	9	10
# of images	250	251	274	224	236	260	248	248
Stage	11	12	13	14	15	16	17	Total
# of images	246	255	251	252	232	243	254	**3724**

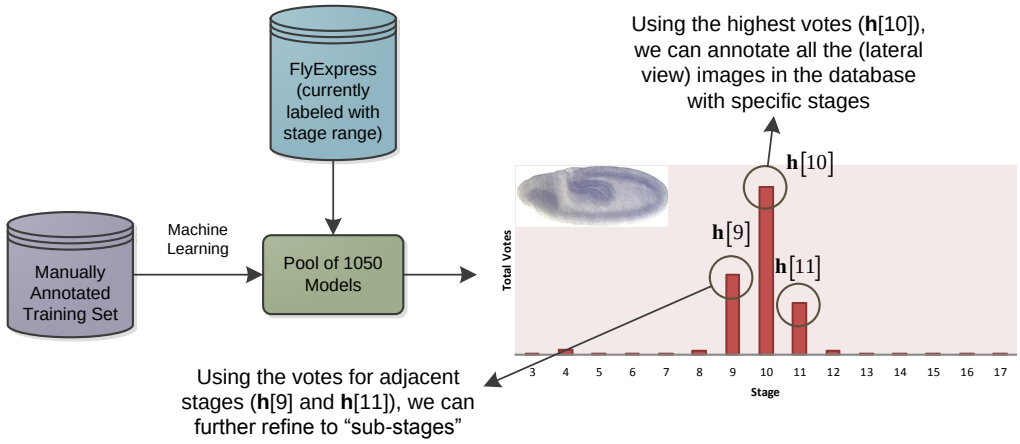


Figure 5.1: Overview of my stage annotation system. By learning from a training dataset with manually labeled stage information, I build a pool of 1050 classification models. I then apply this pool to the unlabeled images in our FlyExpress database, providing a histogram of voting values for each image. The histogram is then used to annotate the image with a specific stage, as well as a more refined "sub-stage" and numerical based "stage score".

stage 10 (44%). This may attribute to the fact that stages 9 and 10 correspond to the slow phase of germ band movement, where the differences between these stages are very small.

For evaluating the performance of my method on independent data (not used for training) at a large scale, I generated S for 36,802 images (lateral views) obtained from the FlyExpress database [112]. A stage assignment was deemed to be correct if S was within the stage range provided by the source BDGP [50, 71]. That is, if an image was annotated as stage 7 by my system (S = 7) and BDGP annotated it as stage range 7-8, then the annotation was considered to be correct. In this case, the accuracy of my annotations was 86.6%, with the highest accuracy seen for stage range 4-6 (96.9%) and the lowest for stage

range 9-10 (74.9%). Visual inspection of mistakes revealed that a handful of images were not lateral views, which means that my classification model was not appropriate for them.

I also performed another independent evaluation by randomly selecting 140 images from stages 4-17. I asked a domain expert to manually annotate these randomly selected images with specific stages (S) and more refined stages (e.g., Early stage 10 [10E], late stage 7 [7L]). Of these, manual annotations were not provided by experts for 23 images because they were too out-of-focus to annotate or not lateral (mislabeled in the database). For the remaining 117 images, computational and manual annotations matched 81% of the times, which is similar to the accuracy observed for the training set. At the level of sub stages, manual and computational annotations matched 75% of the time. Overall, I found that the computational prediction is within one sub-stage of the expert developmental biologists' annotation for 95% of the images tested. Therefore, the computational predictions can provide an excellent set of initial annotations.

Improving Similar Expression Pattern Retrieval

Within the FlyExpress database, we provide a tool for identifying similar gene expression patterns for a given query image [61]. Since the images in FlyExpress are assigned to a stage range, the search can only be done within a particular stage range. However, the comparison of gene expression is most biologically meaningful when the embryos are from similar developmental time points, which means that the use of specific stage would be useful to improving the interpretation of matches. I present two example cases where the use of specific and refined stages leads to better biological insights (Figure 5.2). In Figure 5.2-A, an expression profile of srp gene from stage range 4-6 is used to query for the best matching patterns. It produces results from many different genes within the same stage range. A view of the specific stage enables one to quickly realize that the query image was from stage 6 and that many of the resulting patterns are from earlier stages (e.g., 4 and 5). So, by incorporating specific stage information, the user would have

received results from stage 6 only, which would have been more relevant. A similar situation exists for the second case (Figure 5.2-B), where the expression of Gasp is used to query the database. Results in this case show spurious overlaps with many much earlier stage images (e.g., 13, 14), which have been included simply because of rather coarse stage annotations available. Therefore, we plan to provide users with an option in FlyExpress to view results that potentially represent the best matches that come from the closest predicted stage.

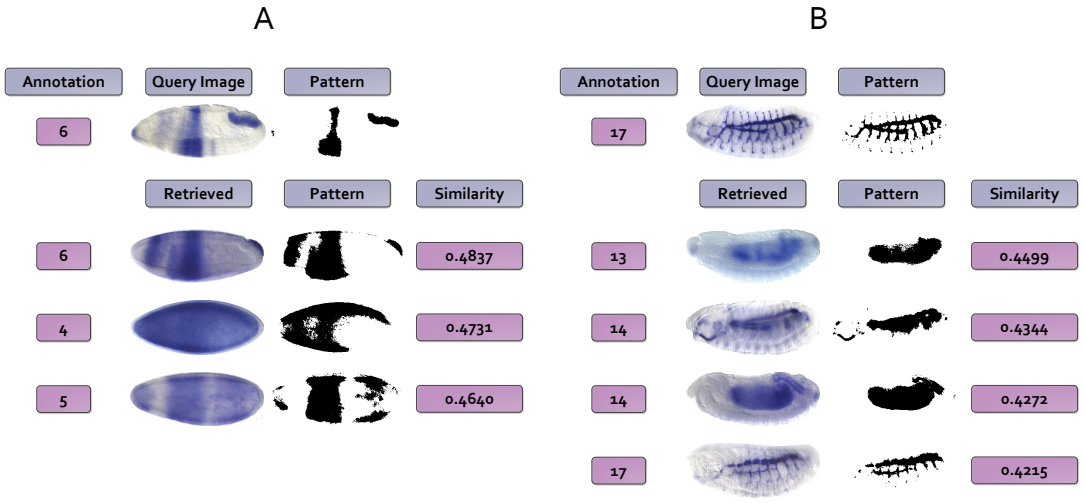


Figure 5.2: Examples of refining image retrieval results using stage annotation. Two example query images are used, with the left one (A) from the Kr gene and right one (B) from Gasp. The top matches from the FlyExpress lateral BDGP images are listed, with corresponding pattern as well as similarity values. The annotated stage from my system is presented on the left side of each expression image.

Genomewide-Expression-Maps with refined stage information

Using the predicted stage information for 36,802 images (lateral views) obtained from the FlyExpress database [112]. I created Genomewide-Expression-Maps (GEMs) that are generated by aggregating and normalizing all spatial gene expression patterns from the same stage [63, 112]. In Figure 5.3, I demonstrate how the use of increasingly refined stage information makes the global views of gene activities increasingly more informative and corresponds to key developmental components. The results are arranged from top to

bottom for images classified by BDGP in stages 7-8, 9-10, and 11-12 (see Figure 5.4 and Figure 5.5 for other stage ranges). In Figure 5.3-A, GEMS for stage range 7-8 lack the information that the germ band is initially more posterior in position and moves towards the anterior, which is easily revealed when images from stage range 7-8 are separated into stages 7 and 8. This trend is further illuminated when the stages are further refined into Early and Late parts (Figure 5.3-C). Overlaying the Hartenstein [113] images on top of these GEMs indicates this (Figure 5.6). Increasingly more refined trend is seen for stages 9-10 and 11-12 as shown in Figure 5.3 (top to bottom in the right column), such that one quickly gets a sense of the developmental progression illuminated by gene expression patterns. These results indicate that the automated stage annotations work well and that refined stages will enable scientists to identify better sets of co-expressed genes.

I also predicted stage score for each image and then build GEMs at an even higher resolution than those in Figure 5.3, which shows how global gene activities vary over developmental time. In addition to categorizing embryo images into finer sub-stages, my stage score can help to sort all embryo images based on their estimated developmental time. This will add great functionalities to our current FlyExpress database; and a preliminary version is already included in our iPhone app [114]. I use three genes (twi, tkv and gt) as examples, and their sorted images are summarized in Figure 5.7, Figure 5.8 and Figure 5.9, respectively. Note that my embryo ordering scheme can only be used to determine the relative order within the same sub-stage. For example, a stage 7.9 image is not necessarily closer to stage 8 than a stage 6.7 image is to stage 7 (refer to the "Material and Method" section for details).

5.2 Materials and Methods

In this section, I introduce the methodologies behind my automatic stage annotation system. In section "Training Set Acquisition" I discuss the training set I built as "ground truth" for my system. I then present the various machine learning methods

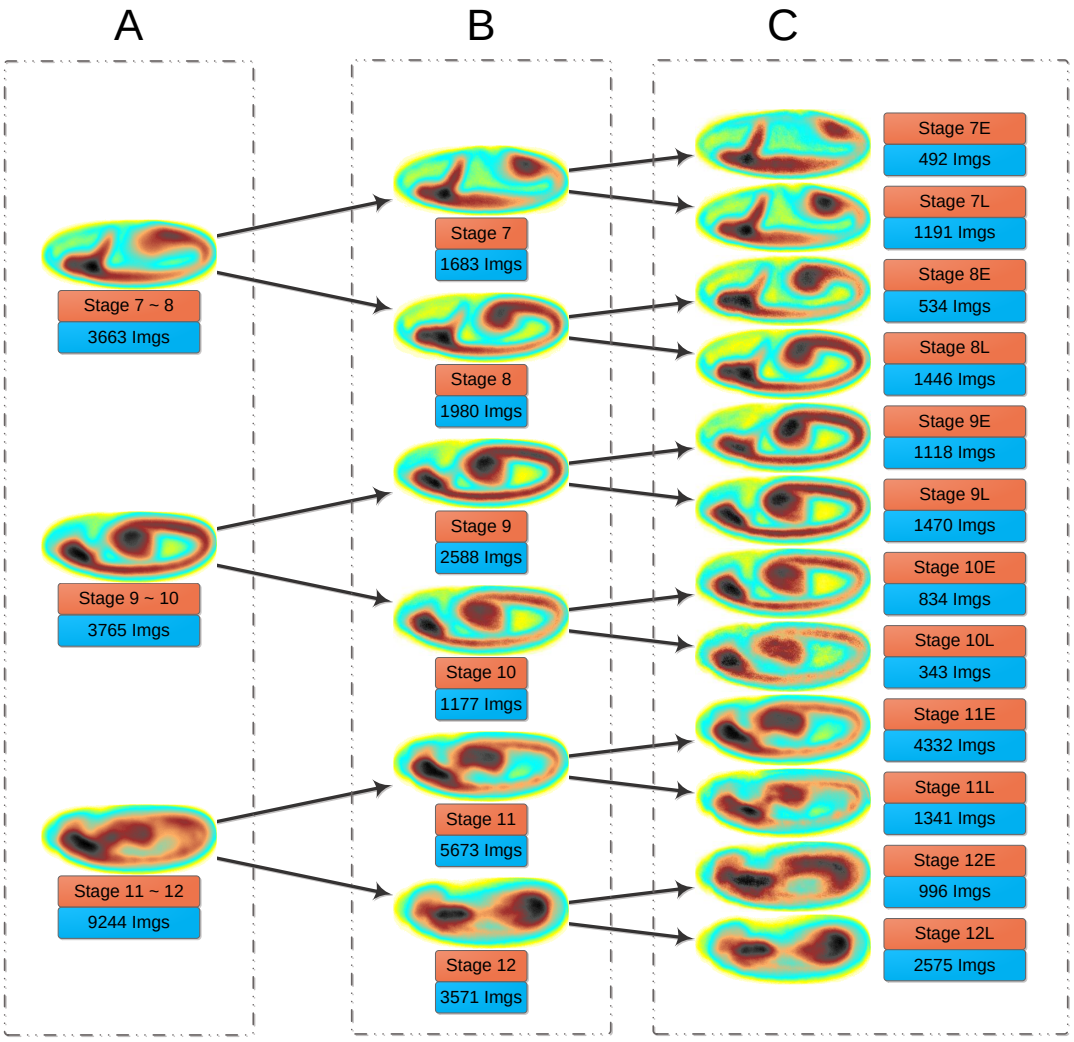


Figure 5.3: Stages 7 - 12 GEMs generated by using only the stage range information (A, left column), the predicted stage information (B, middle column) and the sub-stage information (C, right column). The total number of images involved for creating each individual GEM is also reported.

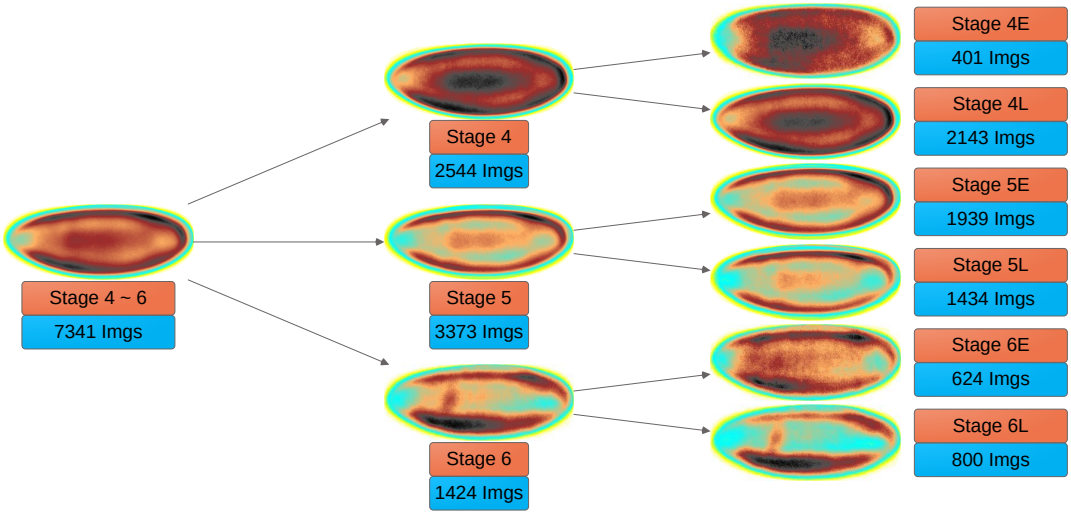


Figure 5.4: Stage 4 - 6 GEMs generated by using only the stage range information (left column), the predicted stage information (middle column) and the sub-stage information (right column). The total number of images involved for creating each individual GEM is also reported.

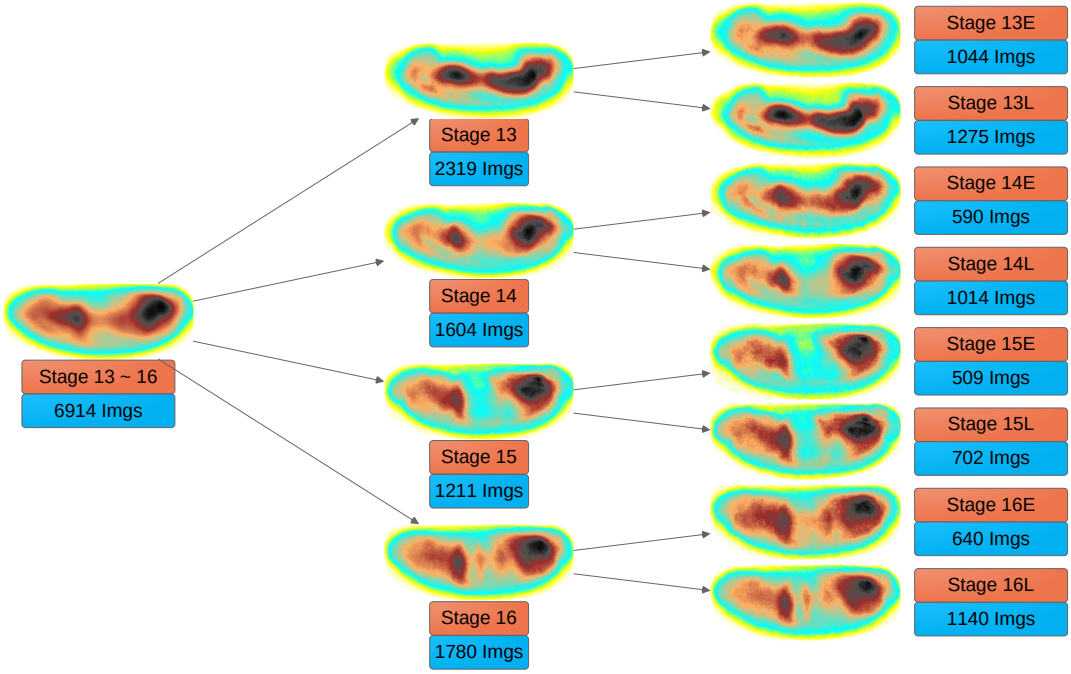


Figure 5.5: Stage 13 - 16 GEMs generated by using only the stage range information (left column), the predicted stage information (middle column) and the sub-stage information (right column). The total number of images involved for creating each individual GEM is also reported.

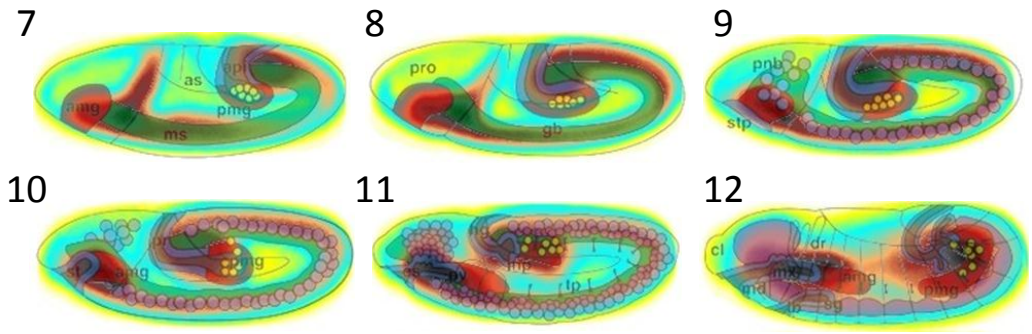


Figure 5.6: Genomewide-Expression-Maps (GEMs) obtained from automatically annotated lateral BDGP images (stages 7-12) overlaid with previously published overview images of stage development.

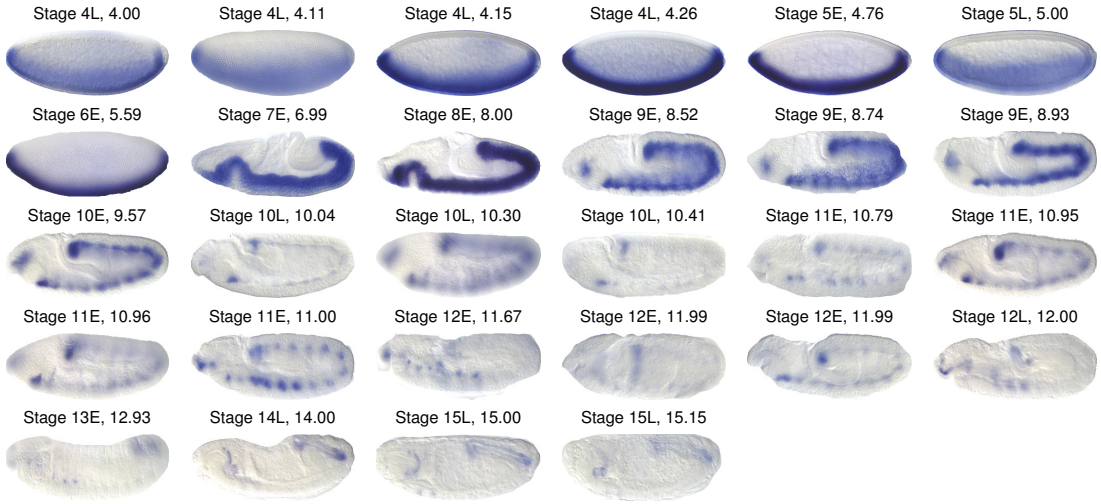


Figure 5.7: Lateral BDGP images from gene twi, sorted by decimal-stage annotations. The annotated stages as well as decimal-stage scores are reported on the top of each expression image.

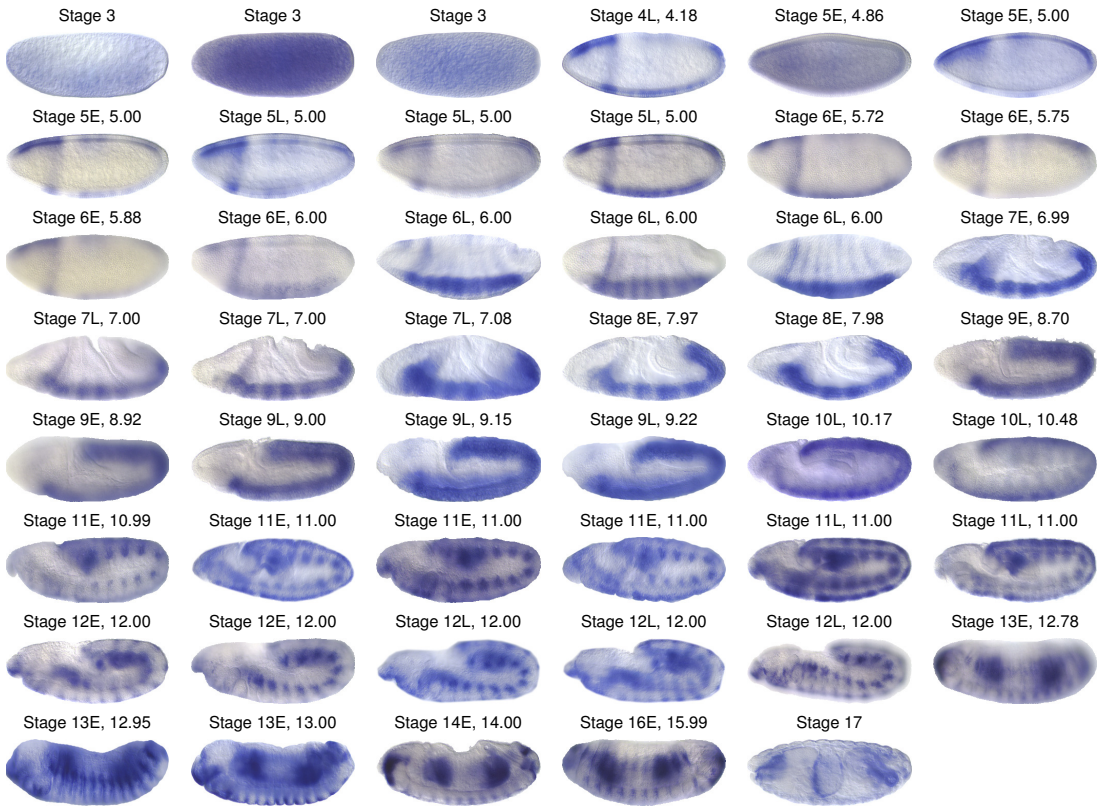


Figure 5.8: Lateral BDGP images from gene tkv, sorted by decimal-stage annotations. The annotated stages as well as decimal-stage scores are reported on the top of each expression image.

employed to create a big pool of models in “Model Pool Construction”. Finally, I introduce the annotation of previously unseen images in “Voting for Stage Annotation and beyond”.

Training Set Acquisition

To develop an automated annotation system, a key component is to build a comprehensive training set, in which each entity (in this case, images of gene expression in *Drosophila* embryo) is associated with the ”accurate” annotation (in this case, the corresponding stage). By learning from the training set, a system will extract critical information from the images that discriminates the developmental stages from each other, and uses the extracted knowledge to build classifiers for predicting the stage of previously unseen

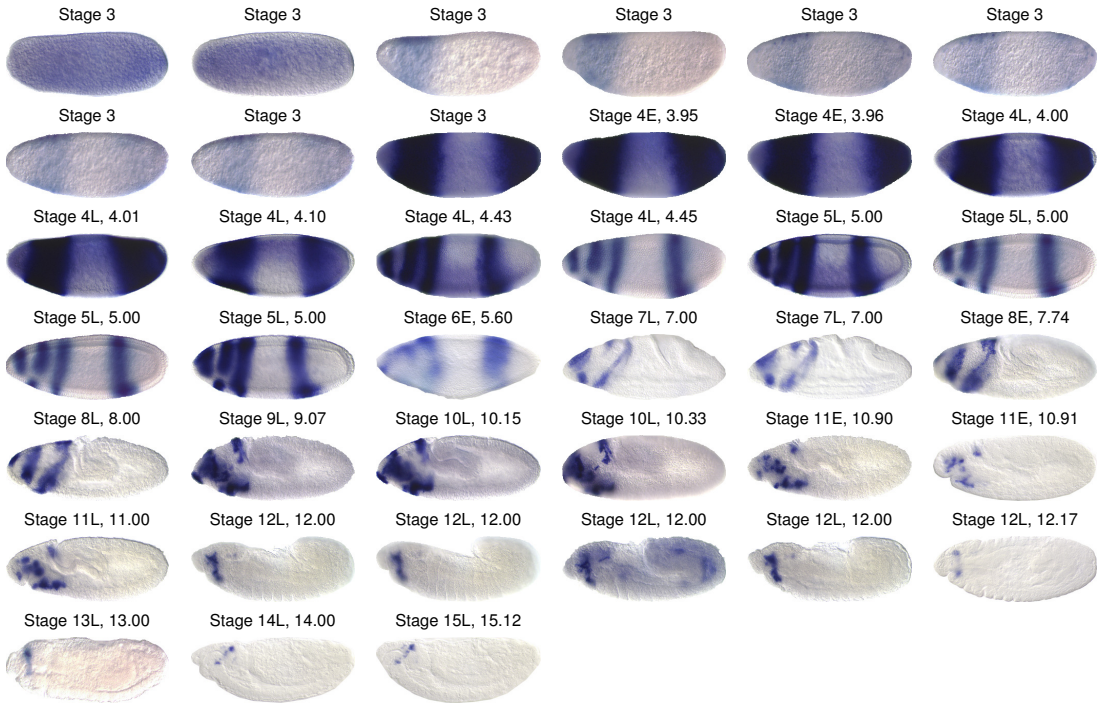


Figure 5.9: Lateral BDGP images from gene gt, sorted by decimal-stage annotations. The annotated stages as well as decimal-stage scores are reported on the top of each expression image.

images. I have manually annotated a collection of images with precise stage labels for a total of 3724 standardized BDGP images (in lateral view) in FlyExpress. The detailed numbers of labeled images are listed in Table 5.1. Embryogenesis in *Drosophila* starts with 13 rapid nuclear divisions after fertilization. Thus, the only morphological difference across the first stage range (stages 1-3) is the number of nuclei, a feature not visible with the microscopy employed by the BDGP consortium. Therefore they are considered as a single stage (stage 3) in this work. The orientation of all images in this study is standardized, and the size is scaled to 128 by 320 pixels.

Model Pool Construction

The key idea of a successful voting system is to build a pool of diverse classification models, each with reasonably good performance. In this section, I will first introduce the

feature extraction process, and then present different ways of building classification models by using the underlying structure of the feature.

Feature Construction

In order to make images from different stages easier for computational models to distinguish, appropriate feature extraction is critical. Log Gabor filters [115, 116] have been shown to offer the best simultaneous localization of spatial and frequency information with an arbitrary bandwidth. They are particularly suitable for my study, since the features distinguishing between different stages should focus on the general morphology of the embryo as well as subtle textures. In the frequency domain, the log Gabor function with respect to radius (r) and angle (θ) can be described by:

$$G(r,\theta) = \exp\left(-\frac{\left(\log\left(r/f_0\right)\right)^2}{2{\sigma_r}^2}\right)\exp\left(-\frac{(\theta-\theta_0)^2}{2{\sigma_\theta}^2}\right),$$

where f_0 is the filter's center frequency, θ_0 is the filter's orientation, σ_r and σ_θ are the corresponding standard deviations. By choosing different values of f_0 and θ_0, one can construct filters with different wavelet scales and orientations.

The procedure of my feature construction is illustrated in Figure 5.10. First, I converted the color image to gray scale. I then used log Gabor filters with 4 different wavelet scales and 6 different filter orientations to extract the texture information. Hence, 24 Gabor images were obtained from the filtering operation. Next, I divided each of the Gabor images into 640 sub-blocks of size 8 by 8; and the mean values were used to represent each of the sub-blocks. The 24 sub-sampled Gabor images were then converted to vectors which were concatenated together as the feature vector for the original image. Thus, the dimension of the final feature vector is $24 \times 640 = 15360$.

Preliminary on Linear Classifiers

The feature construction step maps the images into a feature space, with each dimension corresponding to a specific Gabor feature. I can then denote the training set as

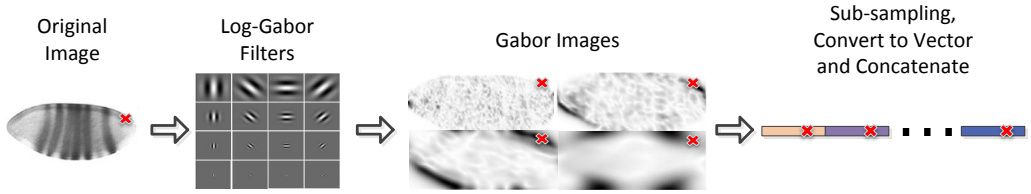


Figure 5.10: Illustration of the feature extraction process. The standardized image is first processed by a series of log-Gabor filters, resulting in 24 Gabor images. These Gabor images are then down sampled and concatenated into a single feature vector, which is the final representation of the original image. As indicated by the red cross in the figure, one sub-block of the original image will correspond to 24 features in the feature vector, one for each Gabor image.

$D = \{X, Y\}$, where $X = \{x_1, \ldots, x_n\}$ are the feature vectors of the annotated images, $Y = \{y_1, \ldots, y_n\}$ are the corresponding stages, and n is the number of training samples. In this study, I apply linear classifiers on this high-dimensional classification problem, and apply the one-versus-the-rest [111] method to convert the multi-class classification problem into a series of binary class problems. Therefore, only binary class classifiers will be discussed in the rest of this section. Specifically, a binary class linear classifier takes the linear combination of the feature vector $\mathbf{x}$ of a sample to make the prediction:

$$y = \operatorname{sgn}\left(\mathbf{w}^T \mathbf{x}\right), \tag{5.1}$$

where $y \in \{-1, 1\}$ is the decision, or the predicted "label" of $\mathbf{x} \in \mathbf{R}^d$, $\mathbf{w} \in \mathbf{R}^d$ is the weight vector of the classifier that needs to be learned from the training data, and $\operatorname{sgn}(\cdot)$ is the sign function.

Learning a linear classifier is to pursuit the optimal weight vector $\mathbf{w}$ on the training set, which can be formulated as the following optimization problem:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \ell(\mathbf{w}, X, Y) + \operatorname{Reg}(\mathbf{w}), \tag{5.2}$$

where $\ell(\mathbf{w}, X, Y)$ is the loss function measuring the discrepancy between the prediction and the ground truth for the training samples, and $\operatorname{Reg}(\mathbf{w})$ is a regularization term

designed to improve the generalization performance of the classifier. The regularization term can be used to impose specific structures on the weight vector; and it will be discussed in detail in the following sub-section. Three common loss functions are used in this study:

- Least square loss [111, 117] (e.g. ridge regression):

$$\ell(\mathbf{w}, X, Y) = \frac{1}{2}\sum_{i=1}^{n}\left(\mathbf{w}^T\mathbf{x}_i - y_i\right)^2$$

- Logistic loss [111, 117] (e.g. logistic regression):

$$\ell(\mathbf{w}, X, Y) = \sum_{i=1}^{n}\log\left(1+\exp\left(y_i\mathbf{w}^T\mathbf{x}_i\right)\right)$$

- Hinge loss [111, 118] (e.g. support vector machine, or SVM):

$$\ell(\mathbf{w}, X, Y) = \sum_{i=1}^{n}\max\left\{0, 1-y_i\mathbf{w}^T\mathbf{x}_i\right\}$$

Exploiting the underlying sparse structure

For high-dimensional, small sample size problems such as the one in my study, $\mathrm{Reg}(\mathbf{w})$ in (5.2) plays a critical role in alleviating over-fitting and improving generalization performance. A common choice (e.g. in ridge regression and SVM) of the regularization term is:

$$\mathrm{Reg}(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2$$

An alternative way of addressing the high-dimensional problem is feature selection. In the rest of this sub-section, I will discuss 3 variants of sparsity-inducing regularizations (ℓ_1 norm, $\ell_{2,1}$ norm and $\ell_{2,1}-\ell_1$ norm) that can impose different types of sparsity patterns on the solution of (5.2), and lead to simultaneous classification and feature selection.

From (5.1), one characteristic of a linear classifier is that if we set a certain entry of to be 0, it is equivalent to removing the corresponding feature. This motivates us to

introduce the ℓ_1 regularization [119]:

$$\text{Reg}(\mathbf{w}) = \lambda \sum_{j=1}^{d} \left|\mathbf{w}_j\right| = \lambda \|\mathbf{w}\|_1$$

The ℓ_1 regularization (also called Lasso) performs feature reduction and classification in a unified formulation. It has been applied successfully in various applications [120]. However, Lasso does not make full use of the underlying structure of our data. Specifically, as shown in my feature extraction process illustrated in Figure 5.10, each region of the image is associated with 24 features, one for each of the 24 different log-Gabor filters. Thus, the features can be naturally partitioned into distinct groups, one for each region of the image. It is then natural to apply group Lasso [100], which can be applied to select feature groups, i.e., image regions. Assume that we partition the index of the features into S disjoint groups $\{\mathscr{G}_1, \ldots, \mathscr{G}_S\}$, one for each region, such that $\mathscr{G}_1 \cup \mathscr{G}_2 \cup \cdots \cup \mathscr{G}_S = \{1, 2, \ldots, d\}$. I can then obtain the $\ell_{2,1}$ norm (also called group Lasso) regularization as follows:

$$\text{Reg}(\mathbf{w}) = \lambda \sum_{i=1}^{S} \left\|\mathbf{w}_{\mathscr{G}_i}\right\|_2,$$

where $\mathbf{w}_{\mathscr{G}_i}$ is the weight vector restricted to the i-th group of features and λ is the parameter that controls the group sparsity.

When we use the $\ell_{2,1}$ norm regularization to perform feature selection, all features from the same group will be selected simultaneously. Thus, only the "between group sparsity" is considered. However, some features from a selected group may be irrelevant to our prediction. In this case, the $\ell_{2,1} - \ell_1$ norm regularization (called sparse group Lasso) [121, 122] can be applied which simultaneously achieves the "between group" sparsity based on the $\ell_{2,1}$ norm and the "within group" sparsity based on the ℓ_1 norm as follows:

$$\text{Reg}(\mathbf{w}) = \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \sum_{i=1}^{S} \left\|\mathbf{w}_{\mathscr{G}_i}\right\|_2$$

Constructing a Pool of Diverse Classifiers The key idea of a successful voting system is to have a large and diverse pool of models, each of them with reasonable prediction power. In this study, I applied SVM with linear kernels from the LIBLINEAR [123] package, and 6 sparse learning algorithms (Lasso, group Lasso and sparse group Lasso with least square and logistic loss) from the SLEP [120] package. I then partition the annotated data set into two disjoint sets, namely, the "training set" where linear classifiers are learned and the "validation set" where the performance of the learned classifiers can be evaluated. Five different training set ratios (from 50% to 90%) are used to partition the data set and for each ratio, 30 random partitions are generated. The average performance in terms of accuracy on the validation sets for different classification algorithms and training ratios is shown in Table 5.2.

Table 5.2: Average classification accuracy on the validation set for different algorithms and different training ratio, when stage range information is not available. 7 algorithms are evaluated, including SVM with linear kernel (SVM) and 6 sparse learning methods: Lasso (lasso), group Lasso (glasso), sparse group Lasso (sglasso) with least square loss (least) and logistic loss (log).

Ratio	SVM	lasso(least)	glasso(least)	sglasso(least)	lasso(log)	glasso(log)	sglasso(log)
50.00%	77.54%	73.64%	74.72%	75.30%	78.34%	78.61%	78.36%
60.00%	77.53%	74.89%	75.08%	76.33%	78.93%	78.93%	78.81%
70.00%	77.58%	75.85%	76.22%	77.02%	79.27%	79.40%	79.18%
80.00%	78.02%	77.33%	77.78%	77.95%	79.55%	79.68%	79.49%
90.00%	77.30%	78.19%	77.95%	77.70%	79.79%	79.88%	79.82%

As I can see from Table 5.2, all 7 classifiers perform comparably and the three sparse learning methods using logistic loss perform slightly better. For my 15-class (stages 3 to 17) classification problem, an accuracy of 80% is reasonably good. I can also see that the validation accuracy generally increases as more samples are used in training, but the increase is not that significant after 70% of the annotated data (about 2600 images) are used for training. This indicates that the annotated data set has an adequate size.

In addition to obtaining a collection of "reasonable" models, I also need the models to be diverse such that the majority voting of the pool will provide robust results

for unseen subjects. For each different training ratio, I calculated the average rate that at least one of the algorithms does not agree with the others, and the results are summarized in Table 5.3.

Table 5.3: Average rate of disagreement among all 7 different algorithms for different training ratios.

Ratio	50%	60%	70%	80%	90%
Average rate of disagreement	29.16%	26.96%	24.31%	21.70%	20.12%

As we can see from Table 5.3, the average rate of disagreement varies from 30% to 20% as the training ratio increases. Therefore, I have built a pool of 1050 (7 algorithms times 5 training ratios times 30 random partitions) diverse models, each of which achieves reasonably good classification performance.

Voting for Stage Annotation and beyond

In this sub-section, I will discuss in detail the voting scheme I designed for annotating the remaining BDGP images in our Fly-Express database.

Stage Annotation by Majority Voting

For a given un-labeled image, I denote the prediction vector for this image based on the i-th model as $\mathbf{y}^i \in \{0,1\}^{15}$, where $\mathbf{y}^i$ is a 15-dimensional binary vector indicating the stage prediction of the i-th model. Specifically, $\mathbf{y}^i[j] = 1$ indicates that the i-th model determines that this image belongs to the j-th stage. I also assign a "confidence level" of the current model as a^i, which is set to be the classification accuracy of this model on the validation set. I then summarize all the predictions from the 1050 models, and obtain a prediction histogram defined as $\mathbf{h} = \sum_{i=1}^{1} 050 a^i \mathbf{y}^i$. Then, the entry with the highest voting will be the stage assigned by the ensemble of the pool of models. That is, the final annotation is defined as $S = \arg\max_i \mathbf{h}[i]$.

Embryogenesis as a Continuous Process

With the proposed majority voting scheme, I can annotate all the lateral BDGP images in our FlyExpress database with precise stages. This is a significant refinement from the currently available stage-range information. However, embryogenesis is a continuous process and a natural question to ask is: based on current information (training set annotated with stages), can we provide further refinement? Here I will present some intermediate results that provide us with suggestions to move beyond stage annotation to even further refinement. I first summarize the voting confidence level for all the models on the training data set. For example, for an image that is known to be from stage 4, I will calculate the actual predictions (they may vary from stage 3 to stage 17) from all the models. I then repeat the process for all the labeled images for stage 4, so that I will generate a ”voting distribution” for images known to be stage 4. The voting distribution for all 15 stages is illustrated in Figure 5.11.

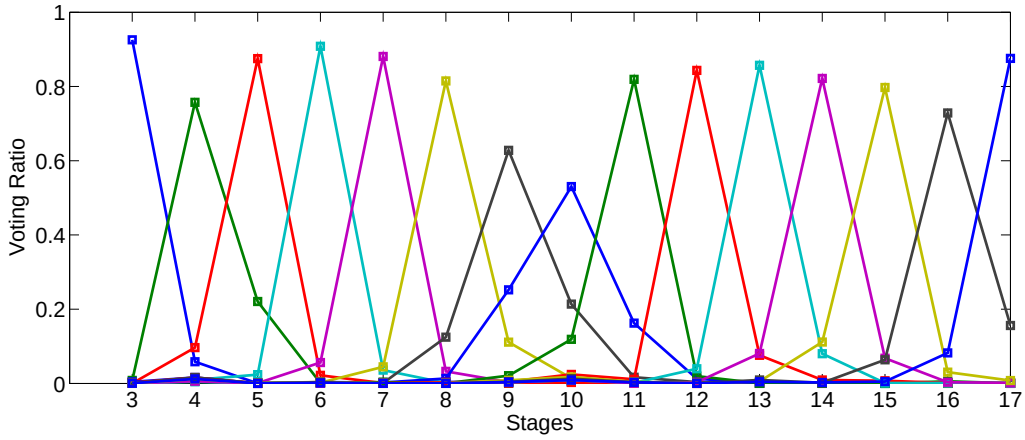


Figure 5.11: Summary of total voting for each stage in the labeled data set. For each stage in the training dataset, I summarize the predicted stages from all the models on all the images to obtain a voting distribution for that stage. The voting distribution will always peak at the corresponding stage.

We can see from Figure 5.11 that the highest ratio always appears in the corresponding stage, which is intuitive since all the models yield reasonable performance.

More interestingly, the second and third highest ratios always appear in the adjacent stages. For example, in the distribution for stage 9 (black line with a peak at stage 9), the second and third highest voting ratio are stages 10 and 8 respectively. This shows that when a model disagrees with the actual stage annotation, it will mostly assign that image to an adjacent stage. This corresponds perfectly to the continuity of the embryogenesis process, and motivates us to make additional use of the prediction histogram **h** to further refine the stage annotation.

Sub-stage Annotation and Decimal Based Embryo Ordering

To illustrate my method of refining stage annotation to sub-stages and the decimal based embryo ordering scheme, I first provide an example of the prediction histogram for a specific image in Figure 5.1. In my current system, only images assigned to stage 4 to 16 have refined stage annotation.

As expected, stage 10 gets the most votes among all 15 stages, and therefore this image will be annotated as stage 10. I then compare the voting scores for the two adjacent stages 9 and 11, and observe that $\mathbf{h}[9] > \mathbf{h}[11]$. Therefore according to my system, this stage 10 image is more similar to stage 9 compared to stage 11. Thus, I will annotate this image as stage 10E (early 10).

In addition to the order information of the prediction histogram, I can assign a continuous stage value for the image. Using Figure 5.1 as an example, I calculate the "stage score" for this image as:

$$SS = 10 - \frac{\mathbf{h}[9]}{\mathbf{h}[9] + \mathbf{h}[10]}$$

The intuition is as follows: the higher value of $\mathbf{h}[9]$ with respect to $\mathbf{h}[10]$ is, the "earlier" this embryo is among all the stage 10E images. Note that this decimal stage value can only be used to suggest a relative order within each sub-stage. For example, in terms of

developmental time, a stage 7.9 image is not necessarily closer to stage 8 than a stage 6.7 image is to stage 7.

With the help of the embryo ordering scheme, I can obtain even more refined stages. For example, I can further divide stage 10E into three sub-sub-stages as follows: Firstly, I sort all the decimal stage values of all the images assigned to stage 10E. I then evenly split the sorted images into three groups, with the first group annotated as stage 10E-a, second as 10E-b and third as 10E-c.

With or Without Stage Range Information

In the current FlyExpress database, all the BDGP images are annotated with stage ranges. Clearly the additional stage range information can be used as prior knowledge to make the prediction of exact stages more reliable. When no range information is available, the problem is a 15-class classification problem. With the additional stage range information, the problem can be reduced into a binary-class, 3-class or 5-class problem depending on which stage range the corresponding image belongs to.

I use similar experimental settings as in Table 5.2, and calculate the average classification performance when the stage range is given. The results are summarized in Table 5.4.

Table 5.4: Average classification accuracy on the validation set for different algorithms and different training ratios, when stage range information is available. 7 algorithms are evaluated, including SVM with linear kernel (SVM) and 6 sparse learning methods: Lasso (lasso), group Lasso (glasso), sparse group Lasso (sglasso) with least square loss (least) and logistic loss (log).

Ratio	SVM	lasso(least)	glasso(least)	sglasso(least)	lasso(log)	glasso(log)	sglasso(log)
50.00%	86.27%	84.78%	85.55%	85.53%	85.34%	85.44%	85.46%
60.00%	86.31%	85.18%	85.85%	85.90%	85.60%	85.85%	85.72%
70.00%	86.46%	85.82%	86.06%	86.01%	85.72%	85.99%	85.97%
80.00%	86.87%	86.34%	86.49%	86.61%	86.29%	86.05%	86.32%
90.00%	87.10%	86.41%	86.47%	86.88%	86.57%	86.46%	86.87%

As we can see from Table 5.4, when the stage range is given, the classification performance for all cases significantly improves. The only drawback of using the stage

range information is that one cannot provide a complete prediction histogram for all 15 stages to perform sub-stage annotation (only the histogram for the corresponding stage range). In my study, I combine the results from both with and without range classifiers. For an un-annotated image, I first use with-range classifiers to determine the stage of this image, and then use the prediction histogram $\mathbf{h}$ of without-range classifiers to perform sub-stage annotation. For example, if the with-range classifiers decide that this image belongs to stage i, I will then use $\mathbf{h}[i-1]$, $\mathbf{h}[i]$ and $\mathbf{h}[i+1]$ to calculate the sub-stage and the decimal stage value.

In the previous evaluations, I use the overall accuracy across all stages to illustrate the performance of my system. It is also interesting to see the differences between stage ranges in terms of the annotation accuracy. I use the linear SVM classifier as an example to summarize the average annotation accuracies for all 5 different stage ranges with different training ratios. The results are presented in Table 5.5. I can conclude from Table 5.5 that the stage range with the best classification performance is stages 11-12, where the accuracy is as high as 97% when 90% of the data is used as training. The most challenging stage range for my system is stages 9-10, where only 76% of the time an accurate prediction is made. This is consistent with my previous experiments.

Table 5.5: Average classification accuracy within each stage range for different training ratios. SVM with linear kernel is used as the classifier and stage range information is available.

Ratio	Stages 4-6	Stages 7-8	Stages 9-10	Stages 11-12	Stages 13-17
50.00%	86.43%	95.27%	74.10%	95.61%	83.64%
60.00%	86.08%	95.27%	75.13%	95.97%	83.48%
70.00%	87.05%	95.39%	74.08%	96.38%	83.53%
80.00%	87.33%	95.79%	75.23%	96.40%	83.89%
90.00%	87.70%	95.60%	76.47%	97.07%	82.76%

More on Model Ensemble

In my final annotation system, all 1050 models are used to form the ensemble. One interesting question to ask is: is it truly beneficial to include all of them? In this

subsection, I use the independent evaluation dataset discussed in the "Results and Discussion" section to validate my choice of large number of models.

Firstly, I show that combining different classification algorithms is essential for the success of model ensemble. I predict the stages of the images from the evaluation set using the ensemble of just the SVM models or just the sparse models. I use the same evaluation criteria as in the "Results and Discussion" section, and the results are summarized in Table 5.6. Formally, I define the criteria as follows:

- **Sub-stage Accuracy** ($\text{Acc}_{0.5}$). Only the images that are annotated with the correct sub-stage are considered accurate. For example, if an "early stage 7" image is annotated as stage 7E by my system, then the annotation is considered correct.
- **Stage Accuracy** ($\text{Acc}_{\text{Stage}}$). The images that are annotated with the correct stage are considered accurate. For example, if an "early stage 7" image is annotated as stage 7E or stage 7L by my system, then the annotation is considered correct.
- **Plus-Minus-Sub-stage Accuracy** ($\text{Acc}_{\pm 0.5}$). The images that are annotated with a sub-stage which is at most "a sub-stage away" from the manually annotated sub-stage are considered accurate. For example, if an "early stage 7" image is annotated as stage 6L, stage 7E or stage 7L by my system, then the annotation is considered correct.

Table 5.6: Performance evaluation of model ensemble using different subsets of the learning algorithms. Three evaluation criteria are used, namely, the sub-stage accuracy ($\text{Acc}_{0.5}$), the stage accuracy ($\text{Acc}_{\text{Stage}}$) and the plus-minus-half accuracy ($\text{Acc}_{\pm 0.5}$). Both with and without range information cases are considered.

	With Range Info Case			Without Range Info Case		
	$\text{Acc}_{0.5}$	$\text{Acc}_{\text{Stage}}$	$\text{Acc}_{\pm 0.5}$	$\text{Acc}_{0.5}$	$\text{Acc}_{\text{Stage}}$	$\text{Acc}_{\pm 0.5}$
SVM + Sparse Algorithms	75.21%	85.47%	94.87%	73.50%	81.20%	93.16%
SVM	57.26%	80.34%	92.31%	52.99%	76.07%	91.45%
Sparse Algorithms	70.09%	85.47%	94.87%	68.38%	82.05%	94.02%

As we can see from Table 5.6, neither SVM models nor sparse models yield competitive results, while the best performance is achieved by combining all of them together. This is especially true for the side-stage accuracy, where a large number of diverse models are essential for accurately predicting if an image is from the early or late part of a certain stage.

I have shown that combining different learning algorithms is beneficial. Next, I present another set of experiments where a subset of models (instead of a subset of algorithms) are selected to form the ensemble. This is often referred to as ensemble pruning [124]. Two model selection methods are tested:

- **Random Subset**. In this setting, a number of models are randomly selected from the trained models to form the model ensemble. This process is repeated 1000 times to report the average result.
- **Ranked by Validation Accuracy**. In this setting, the trained 1050 models are first ranked by their classification accuracy on the validation set, with the most accurate one on the top of the list. Then, only the top few of the models are selected to form the model ensemble.

I vary the number of models used for the ensemble, and report the performance using the independent evaluation dataset in Table 5.7. As we can observe from Table 5.7, 30 models are sufficient to accurately annotate the stage of a given image. By increasing the number of models, the stage accuracy ($\text{Acc}_{\text{Stage}}$) and the plus-minus-half accuracy ($\text{Acc}_{\pm 0.5}$) remain stable while the sub-stage accuracy ($\text{Acc}_{0.5}$) steadily improves from 63% to 75.5%. Thus, for accurate sub-stage annotation, combining all models is essential.

Table 5.7: Performance evaluation of model ensemble using different subsets of the trained models. In the "Random Subset" category, a number of models are randomly selected from the trained models to form the model ensemble. In the "Ranked by Validation Accuracy" category, the trained models are first ranked by their accuracy on the corresponding validation set such that the top models can be selected. Only the case without stage range information is considered.

	Random Subset			Ranked by Validation Accuracy		
Number of Models	$\text{Acc}_{0.5}$	$\text{Acc}_{\text{Stage}}$	$\text{Acc}_{\pm 0.5}$	$\text{Acc}_{0.5}$	$\text{Acc}_{\text{Stage}}$	$\text{Acc}_{\pm 0.5}$
30	62.82%	81.63%	93.49%	59.83%	82.05%	92.31%
50	64.56%	81.65%	93.56%	62.39%	81.20%	93.16%
100	66.96%	81.58%	93.59%	62.39%	81.20%	94.02%
200	69.22%	81.49%	93.52%	62.39%	81.20%	94.87%
500	71.41%	81.42%	93.44%	66.67%	82.05%	94.87%
1050	73.50%	81.20%	93.16%	73.50%	81.20%	93.16%

Chapter 6

Conclusion and Outlook

In this chapter, I summarize the major contributions made in this thesis and discuss possible future directions.

6.1 Summary of Contributions

The major theme of this thesis is to demonstrate that how structral sparse learning can help scientific discoveries in biomedical and biology fields. I present an efficient algorithm for solving the general group structured problems, the overlapping group lasso. In the area of Alzheimer's Disease research, I identified the problem of structured missing values, and present two approaches to achieve feature learning in such scenario without having to guess a complete block of missing data. In the analysis of *Drosophila* embryo gene expression images, I present a novel feature extraction method such that accurate image retrieval and term annotation can be achieved. Moreover, I built the first automatic system of embryo developmental stage annotation.

I consider the efficient optimization of the overlapping group Lasso penalized problem based on the accelerated gradient descent method. I reveal several key properties of the proximal operator associated with the overlapping group Lasso, and compute the proximal operator via solving the smooth and convex dual problem. Numerical experiments on both synthetic and the breast cancer data set demonstrate the efficiency of the proposed algorithm. Although with an inexact proximal operator, the optimal convergence rate of the accelerated gradient descent might not be guaranteed [92, 93], the algorithm performs quite well empirically. My algorithm is extended to tackle the generalized q norm, as well as a non-convex formulation based on the capped norm regularization. My preliminary results show that the capped norm leads to improved sparse pattern recovery.

In the feature learning with block-wise missing data, I propose two different multi-source learning schemes for block-wise incomplete data, and use them to construct automatic, robust classifiers. In my experiments, the proposed systems greatly improve the classification accuracy on the ADNI data set. My method (iMSF) has two major advantages: 1) All subjects, so long as at least one of the feature sources is available, can be used for feature learning, and all of them can contribute to the feature selection jointly; 2) the difficulty of guessing unknowns is bypassed, as the feature learning is only based on what data is available. To the best of our knowledge, in the ADNI data set, we are the first group who tried to utilize all the available information for classification by allowing the use of subjects with incomplete data. In our current pilot work, I assessed whether our multi-source learning models help to boost the statistical power in the whole data set. Except for some of the PET imaging measures (such as Pittsburgh compound B), I use all other measures that are publicly available at ADNI web site. I hope our work will increase interest in this important problem and that other groups might consider using this approach (not throwing out data) when performing future ADNI classification studies.

The spatial-sparse feature learning provides computational methods for annotating *Drosophila* gene expression pattern images, and identifying similar images based on gene patterning. In both tasks, images are represented as bags-of-words. The size of the bags is determined by the spatial properties of a representation. For both applications, a sparse learning framework was used. Results on the FlyExpress database indicate that the proposed annotation method outperforms the non-sparse, non-spatial bag-of-words method, as well as approaches that would use either a sparse or spatial framework.

In the automatic annotation of *Drosophila* embryo developmental stages, I use the annotated set of training images to build a large collection of diverse models. For any unseen expression image, these models will each give an individual annotation, which are aggregated into an annotation histogram. This histogram will not only provide the best

estimation for the corresponding developmental stage, it also enables us to assign a decimal based stage score that can be use to sort all the expression images in terms of developmental time. I thus developed a computational system that produces refined stage information applicable to existing images and available for future images. This advancement enables the biologically-meaningful mining of genes with similar spatial expression patterns and calculating the developmental trajectories of gene expression.

6.2 Future Directions

In the feature learning with block-wise missing data problem, one potential drawback of my proposed iMSF method is that I have to construct one linear model for each of the possible data source combination. When the number of data sources increases, the number of required models will increase exponentially. Thus, the training samples available for each model will drop dramatically. For example, if we have 10 different data sources, the number of linear models required for iMSF will increase to 1023. It will require a huge training set to provide adequate number of samples for each of these models. This will be an interesting and challenging problem for future research.

In the *Drosophila* developmental time annotation problem, the voting histogram $\mathbf{h}$ obtained for each individual image may not have the nice shape as shown in Figure 5.1. Sometimes, the adjacent voting value may be 0. For example, when $\mathbf{h}[10]$ holds the maximum value of the histogram, $\mathbf{h}[9]$ and $\mathbf{h}[11]$ may both be 0. In this case, my current system will simply yield a stage score of 10.0, and no sub-stage annotation can be done. One possible fix to this issue is to train multiple classifiers dedicated for sub-stage and stage score annotation. We may have a classifier for stage 3 versus 5, 4 versus 6, etc. When an embryo is assigned to stage 10, then another set of classifiers will be applied to determine if this embryo is more similar to 9 or 11. Other interesting topics for this stage annotation problem include stage annotation for other views, other databases and automatic identification of improper images (e.g., bad quality, wrong views).

REFERENCES

[1] R. Jenatton, J.-Y. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *arXiv:0904.3523*, 2009.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," 2010.

[3] X. Chen, Q. Lin, S. Kim, J. Carbonell, and E. Xing, "Smoothing proximal gradient method for general structured sparse learning," *Annals of Applied Statistics*, vol. 6, no. 2, pp. 719–752, 2012.

[4] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[5] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal Of The Royal Statistical Society Series B*, vol. 68, no. 1, pp. 49–67, 2006.

[6] J. Liu, L. Yuan, and J. Ye, "An efficient algorithm for a class of fused lasso problems," in *KDD*, 2010.

[7] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal Of The Royal Statistical Society Series B*, vol. 67, no. 1, pp. 91–108, 2005.

[8] J. Liu and J. Ye, "Moreau-yosida regularization for grouped tree structure learning," in *NIPS*, 2010.

[9] S. Yang, L. Yuan, Y.-C. Lai, X. Shen, P. Wonka, and J. Ye, "Feature grouping and selection over an undirected graph," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 922–930. [Online]. Available: http://doi.acm.org/10.1145/2339530.2339675

[10] C. Li and H. Li, "Network-constrained regularization and variable selection for analysis of genomic data," *Bioinformatics*, vol. 24, no. 9, pp. 1175–1182, 2008.

[11] H. Liu, M. Palatucci, and J. Zhang, "Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery," in *ICML*, 2009.

[12] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization," in *UAI*, 2009.

[13] L. Meier, S. Geer, and P. Bühlmann, "The group lasso for logistic regression," *Journal of the Royal Statistical Society: Series B*, vol. 70, pp. 53–71, 2008.

[14] L. Jacob, G. Obozinski, and J. Vert, "Group lasso with overlap and graph lasso," in *ICML*, 2009.

[15] H. D. Bondell and B. J. Reich, "Simultaneous regression shrinkage, variable selection and clustering of predictors with oscar," *Biometrics*, vol. 64, pp. 115–123, 2008.

[16] S. Kim and E. P. Xing, "Tree-guided group lasso for multi-task regression with structured sparsity," in *ICML*, 2010.

[17] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach, "Network flow algorithms for structured sparsity," in *NIPS*, 2010, pp. 1558–1566.

[18] P. Zhao, G. Rocha, and B. Yu., "The composite absolute penalties family for grouped and hierarchical variable selection," *Annals of Statistics*, vol. 37, no. 6A, pp. 3468–3497, 2009.

[19] R. Kuljiš, "Grand challenges in dementia 2010," *Frontiers in Neurology*, vol. 1, 2010.

[20] E. Reiman, J. Langbaum, and P. Tariot, "Alzheimer's prevention initiative: a proposal to evaluate presymptomatic treatments as quickly as possible," *Biomarkers*, vol. 4, no. 1, pp. 3–14, 2010.

[21] M. Braskie, A. Klunder, K. Hayashi, H. Protas, V. Kepe, K. Miller, S. Huang, J. Barrio, L. Ercoli, P. Siddarth *et al.*, "Plaque and tangle imaging and cognition in normal aging and Alzheimer's disease," *Neurobiology of Aging*, vol. 31, no. 10, pp. 1669–1678, 2010.

[22] C. Jack Jr, F. Barkhof, M. Bernstein, M. Cantillon, P. Cole, C. DeCarli, B. Dubois, S. Duchesne, N. Fox, G. Frisoni *et al.*, "Steps to standardization and validation of

hippocampal volumetry as a biomarker in clinical trials and diagnostic criterion for Alzheimer's disease," *Alzheimer's and Dementia*, vol. 7, no. 4, pp. 474–485, 2011.

[23] D. Devanand, G. Pradhaban, X. Liu, A. Khandji, S. De Santi, S. Segal, H. Rusinek, G. Pelton, L. Honig, R. Mayeux *et al.*, "Hippocampal and entorhinal atrophy in mild cognitive impairment," *Neurology*, vol. 68, no. 11, pp. 828–836, 2007.

[24] C. Fennema-Notestine, D. Hagler Jr, L. McEvoy, A. Fleisher, E. Wu, D. Karow, and A. Dale, "Structural MRI biomarkers for preclinical and mild Alzheimer's disease," *Human Brain Mapping*, vol. 30, no. 10, pp. 3238–3253, 2009.

[25] S. Mueller, M. Weiner, L. Thal, R. Petersen, C. Jack, W. Jagust, J. Trojanowski, A. Toga, and L. Beckett, "The Alzheimer's Disease Neuroimaging Initiative," *Neuroimaging Clinics of North America*, vol. 15, no. 4, p. 869, 2005.

[26] C. Jack Jr, M. Bernstein, N. Fox, P. Thompson, G. Alexander, D. Harvey, B. Borowski, P. Britson, J. L Whitwell, C. Ward *et al.*, "The Alzheimer's disease neuroimaging initiative (ADNI): MRI methods," *Journal of Magnetic Resonance Imaging*, vol. 27, no. 4, pp. 685–691, 2008.

[27] V. Ibanez, P. Pietrini, G. Alexander, M. Furey, D. Teichberg, J. Rajapakse, S. Rapoport, M. Schapiro, and B. Horwitz, "Regional glucose metabolic abnormalities are not the result of atrophy in Alzheimer's disease," *Neurology*, vol. 50, no. 6, pp. 1585–1593, 1998.

[28] R. Casanova, R. Srikanth, A. Baer, P. Laurienti, J. Burdette, S. Hayasaka, L. Flowers, F. Wood, and J. Maldjian, "Biological parametric mapping: a statistical toolbox for multimodality brain image analysis," *Neuroimage*, vol. 34, no. 1, pp. 137–143, 2007.

[29] C. Jack, V. Lowe, M. Senjem, S. Weigand, B. Kemp, M. Shiung, D. Knopman, B. Boeve, W. Klunk, C. Mathis *et al.*, "[11C]-PiB and structural MRI provide complementary information in imaging of Alzheimer's disease and amnestic mild cognitive impairment," *Brain*, vol. 131, no. 3, p. 665, 2008.

[30] S. Landau, D. Harvey, C. Madison, R. Koeppe, E. Reiman, N. Foster, M. Weiner, and W. Jagust, "Associations between cognitive, functional, and FDG-PET measures of decline in AD and MCI," *Neurobiology of Aging*, vol. 32, no. 7, pp. 1207–1218, 2011.

[31] K. Worsley, J. Poline, K. Friston, and A. Evans, "Characterizing the response of PET and fMRI data using multivariate linear models," *Neuroimage*, vol. 6, no. 4, pp. 305–319, 1997.

[32] E. Martinez-Montes, P. Valdés-Sosa, F. Miwakeichi, R. Goldman, and M. Cohen, "Concurrent EEG/fMRI analysis by multiway partial least squares," *Neuroimage*, vol. 22, no. 3, pp. 1023–1034, 2004.

[33] Y. Fan, S. Resnick, X. Wu, and C. Davatzikos, "Structural and functional biomarkers of prodromal Alzheimer's disease: a high-dimensional pattern classification study," *Neuroimage*, vol. 41, no. 2, pp. 277–285, 2008.

[34] J. Ye, K. Chen, T. Wu, J. Li, Z. Zhao, R. Patel, M. Bae, R. Janardan, H. Liu, G. Alexander *et al.*, "Heterogeneous data fusion for Alzheimer's disease study," in *Proceeding of the 14th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*. ACM, 2008, pp. 1025–1033.

[35] P. Vemuri, H. Wiste, S. Weigand, L. Shaw, J. Trojanowski, M. Weiner, D. Knopman, R. Petersen, C. J. Jack, and ADNI, "MRI and CSF biomarkers in normal, MCI, and AD subjects: diagnostic discrimination and cognitive correlations," *Neurology*, vol. 73(4), pp. 287–293, 2009.

[36] O. Kohannim, X. Hua, D. Hibar, S. Lee, Y. Chou, A. Toga, C. Jack Jr, M. Weiner, P. Thompson *et al.*, "Boosting power for clinical trials using classifiers based on multiple biomarkers," *Neurobiology of Aging*, vol. 31, no. 8, pp. 1429–1442, 2010.

[37] Y. Wang, Y. Fan, P. Bhatt, and C. Davatzikos, "High-dimensional pattern regression using machine learning: From medical images to continuous clinical variables," *Neuroimage*, vol. 50, no. 4, pp. 1519–1535, 2010.

[38] S. Lemm, B. Blankertz, T. Dickhaus, and K. Muller, "Introduction to machine learning for brain imaging," *Neuroimage*, 2010.

[39] D. Zhang, Y. Wang, L. Zhou, H. Yuan, D. Shen *et al.*, "Multimodal classification of Alzheimer's disease and mild cognitive impairment," *Neuroimage*, 2011.

[40] P. Van Ness, T. Murphy, K. Araujo, M. Pisani, and H. Allore, "The use of missingness screens in clinical epidemiologic research has implications for regression modeling," *Journal of clinical epidemiology*, vol. 60, no. 12, pp. 1239–1245, 2007.

[41] S. Hardy, H. Allore, and S. Studenski, "Missing data: A special challenge in aging research," *Journal of the American Geriatrics Society*, vol. 57, no. 4, pp. 722–729, 2009.

[42] R. Palmer and D. Royall, "Missing data? Plan on it!" *Journal of the American Geriatrics Society*, vol. 58, pp. S343–S348, 2010.

[43] T. Hastie, R. Tibshirani, G. Sherlock, M. Eisen, P. Brown, and D. Botstein, "Imputing missing data for gene expression arrays," *Technical Report*, 1999.

[44] T. Schneider, "Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values," *Journal of Climate*, vol. 14, no. 5, pp. 853–871, 2001.

[45] S. Gao, "A shared random effect parameter approach for longitudinal dementia data with non-ignorable missing data," *Statistics in Medicine*, vol. 23, no. 2, pp. 211–219, 2004.

[46] J. Schott, J. Bartlett, J. Barnes, K. Leung, S. Ourselin, N. Fox, and ADNI, "Reduced sample sizes for atrophy outcomes in Alzheimer's disease trials: baseline adjustment," *Neurobiology of Aging*, vol. 31, no. 8, p. 1452, 2010.

[47] J. Cai, E. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal of Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[48] E. Candes and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *Information Theory, IEEE Transactions on*, vol. 56, no. 5, pp. 2053–2080, 2010.

[49] P. Tomancak, A. Beaton, R. Weiszmann, E. Kwan, S. Shu, S. E. Lewis, S. Richards, M. Ashburner, V. Hartenstein, S. E. Celniker, and G. M. Rubin, "Systematic determination of patterns of gene expression during *drosophila* embryogenesis," *Genome Biology*, vol. 3, no. 12, 2002.

[50] P. Tomancak, B. Berman, A. Beaton, R. Weiszmann, E. Kwan, V. Hartenstein, S. Celniker, and G. Rubin, "Global analysis of patterns of gene expression during *drosophila* embryogenesis," *Genome Biology*, vol. 8, no. 7, p. R145, 2007.

[51] G. Grumbling, V. Strelets, and The FlyBase Consortium, "FlyBase: anatomical data, images and queries," *Nucleic Acids Research*, vol. 34, pp. D484–488, 2006.

[52] C. C. Fowlkes, C. L. Luengo Hendriks, S. V. Keränen, G. H. Weber, O. Rübel, M.-Y. Huang, S. Chatoor, A. H. DePace, L. Simirenko, C. Henriquez, A. Beaton, R. Weiszmann, S. Celniker, B. Hamann, D. W. Knowles, M. D. Biggin, M. B. Eisen, and J. Malik, "A quantitative spatiotemporal atlas of gene expression in the *drosophila* blastoderm," *Cell*, vol. 133, no. 2, pp. 364–374, April 2008.

[53] E. Lécuyer, H. Yoshida, N. Parthasarathy, C. Alm, T. Babak, T. Cerovina, T. R. Hughes, P. Tomancak, and H. M. Krause, "Global analysis of mRNA localization reveals a prominent role in organizing cellular architecture and function," *Cell*, vol. 131, pp. 174–187, 2007.

[54] A. A. Samsonova, M. Niranjan, S. Russell, and A. Brazma, "Prediction of gene expression in embryonic structures of drosophila melanogaster," *PLoS Comput Biol*, vol. 3, no. 7, p. e144, 07 2007. [Online]. Available: http://dx.plos.org/10.1371%2Fjournal.pcbi.0030144

[55] C. Luengo Hendriks, S. Keranen, C. Fowlkes, L. Simirenko, G. Weber, A. DePace, C. Henriquez, D. Kaszuba, B. Hamann, M. Eisen, J. Malik, D. Sudar, M. Biggin, and D. Knowles, "Three-dimensional morphology and gene expression in the *drosophila* blastoderm at cellular resolution i: data acquisition pipeline," *Genome Biology*, vol. 7, no. 12, p. R123, 2006.

[56] S. Keranen, C. Fowlkes, C. Luengo Hendriks, D. Sudar, D. Knowles, J. Malik, and M. Biggin, "Three-dimensional morphology and gene expression in the *drosophila* blastoderm at cellular resolution ii: dynamics," *Genome Biology*, vol. 7, no. 12, p. R124, 2006.

[57] G. H. Weber, O. Rubel, M.-Y. Huang, A. H. DePace, C. C. Fowlkes, S. V. E. Keranen, C. L. Luengo Hendriks, H. Hagen, D. W. Knowles, J. Malik, M. D. Biggin, and B. Hamann, "Visual exploration of three-dimensional gene expression using physical views and linked abstract views," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 99, no. 1, 2008.

[58] E. Frise, A. S. Hammonds, and S. E. Celniker, "Systematic image-driven analysis of the spatial *drosophila* embryonic expression landscape," *Molecular Systems Biology*, vol. 6, no. 345, 2010.

[59] D. L. Mace, N. Varnado, W. Zhang, E. Frise, and U. Ohler, "Extraction and comparison of gene expression patterns from 2D RNA in situ hybridization images," *Bioinformatics*, vol. 26, no. 6, pp. 761–769, 2010.

[60] K. Puniyani, C. Faloutsos, and E. P. Xing, "SPEX2: automated concise extraction of spatial gene expression patterns from Fly embryo ISH images," *Bioinformatics*, vol. 26, no. 12, pp. i47–i56, 2010. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/26/12/i47.abstract

[61] S. Kumar, K. Jayaraman, S. Panchanathan, R. Gurunathan, A. Marti-Subirana, and S. J. Newfeld, "Best: A novel computational approach for comparing gene expression patterns from early stages of drosophila melanogaster development," *Genetics*, vol. 162, no. 4, pp. 2037–2047, 2002. [Online]. Available: http://www.genetics.org/cgi/content/abstract/162/4/2037

[62] R. Gurunathan, B. V. Emden, S. Panchanathan, and S. Kumar, "Identifying spatially similar gene expression patterns in early stage fruit fly embryo images: binary feature versus invariant moment digital representations," *BMC Bioinformatics*, vol. 5, no. 202, p. 13, 2004.

[63] C. E. Konikoff, T. L. Karr, M. McCutchan, S. J. Newfeld, and S. Kumar, "Comparison of embryonic expression within multigene families using the flyexpress discovery platform reveals more spatial than temporal divergence," *Dev Dyn*, vol. 241, no. 1, pp. 150–60, 2012.

[64] S. Ji, L. Sun, R. Jin, S. Kumar, and J. Ye, "Automated annotation of *drosophila* gene expression patterns using a controlled vocabulary," *Bioinformatics*, vol. 24, no. 17, pp. 1881–1888, 2008.

[65] E. Lécuyer and P. Tomancak, "Mapping the gene expression universe," *Current Opinion in Genetics & Development*, pp. 506–512, 2008.

[66] J. Ye, J. Chen, R. Janardan, and S. Kumar, "Developmental stage annotation of *drosophila* gene expression pattern images via an entire solution path for LDA," *ACM Transactions Knowledge Discovery from Data*, vol. 2, no. 1, pp. 1–21, 2008.

[67] S. Ji, Y.-X. Li, Z.-H. Zhou, S. Kumar, and J. Ye, "A bag-of-words approach for drosophila gene expression pattern annotation," *BMC Bioinformatics*, vol. 10, no. 1, p. 119, 2009.

[68] S. Ji, L. Yuan, Y.-X. Li, Z.-H. Zhou, S. Kumar, and J. Ye, "*Drosophila* gene expression pattern annotation using sparse features and term-term interactions," in *Proceedings of the Fifteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 407–416.

[69] Y.-X. Li, S. Ji, S. Kumar, J. Ye, and Z.-H. Zhou, "*Drosophila* gene expression pattern annotation through multi-instance multi-label learning," in *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*, 2009, pp. 1445–1450.

[70] S. Ji, L. Sun, R. Jin, and J. Ye, "Multi-label multiple kernel learning," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., 2009, pp. 777–784.

[71] P. Tomancak, A. Beaton, R. Weiszmann, E. Kwan, S. Shu, S. E. Lewis, S. Richards, M. Ashburner, V. Hartenstein, S. E. Celniker, and G. M. Rubin, "Systematic determination of patterns of gene expression during drosophila embryogenesis," *Genome Biology*, vol. 3, no. 12, 2002.

[72] T. Walter, D. W. Shattuck, R. Baldock, M. E. Bastin, A. E. Carpenter, S. Duce, J. Ellenberg, A. Fraser, N. Hamilton, S. Pieper, M. A. Ragan, J. E. Schneider, P. Tomancak, and J. K. Heriche, "Visualization of image data from cells to organisms," *Nat Methods*, vol. 7, no. 3 Suppl, pp. S26–41, 2010.

[73] A. T. Kalinka, K. M. Varga, D. T. Gerrard, S. Preibisch, D. L. Corcoran, J. Jarrells, U. Ohler, C. M. Bergman, and P. Tomancak, "Gene expression divergence recapitulates the developmental hourglass model," *Nature*, vol. 468, no. 7325, pp. 811–814, 2010.

[74] M. Osterfield, X. Du, T. Schupbach, E. Wieschaus, and S. Y. Shvartsman, "Three-dimensional epithelial morphogenesis in the developing drosophila egg," *Dev Cell*, vol. 24, no. 4, pp. 400–10, 2013.

[75] J. Campos-Ortega and V. Hartenstein, *The embryonic development of Drosophila melanogaster*. Springer, 1997.

[76] J. Ye, J. Chen, Q. Li, and S. Kumar, "Classification of *drosophila* embryonic developmental stage range based on gene expression pattern images," in *Proceedings of the Computational Systems Bioinformatics Conference*, 2006, pp. 293–298.

[77] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[78] A. Nemirovski, *Efficient methods in convex programming*. Lecture Notes, 1994.

[79] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

[80] J.-J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bull. Soc. Math. France*, vol. 93, pp. 273–299, 1965.

[81] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for sparse hierarchical dictionary learning," in *ICML*, 2010.

[82] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[83] J. Liu and J. Ye, "Efficient l1/lq norm regularization," *Arxiv preprint arXiv:1009.4766*, 2010.

[84] T. Zhang, "Multi-stage convex relaxation for feature selection," *Arxiv preprint arXiv:1106.0565*, 2011.

[85] X. Shen, W. Pan, and Y. Zhu, "Likelihood-based selection and sharp parameter estimation," *Journal of the American Statistical Association*, vol. 107, no. 497, pp. 223–232, 2012.

[86] T. Zhang, "Analysis of multi-stage convex relaxation for sparse regularization," *The Journal of Machine Learning Research*, vol. 11, pp. 1081–1107, 2010.

[87] J. Liu and J. Ye, *SLEP: Sparse Learning with Efficient Projections*, Arizona State University, 2009. [Online]. Available: http://www.public.asu.edu/~jye02/Software/SLEP

[88] A. Argyriou, C. Micchelli, M. Pontil, L. Shen, and Y. Xu, "Efficient first order methods for linear composite regularizers," *Arxiv preprint arXiv:1104.1436*, 2011.

[89] M. J. Van de Vijver and et al., "A gene-expression signature as a predictor of survival in breast cancer," *The New England Journal of Medicine*, vol. 347, no. 25, pp. 1999–2009, 2002.

[90] A. Subramanian and et al., "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles," *Proceedings of the National Academy of Sciences*, vol. 102, no. 43, pp. 15 545–15 550, 2005.

[91] H. Y. Chuang, E. Lee, Y. T. Liu, D. Lee, and T. Ideker, "Network-based classification of breast cancer metastasis," *Molecular Systems Biology*, vol. 3, no. 140, 2007.

[92] R. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM Journal on Control and Optimization*, vol. 14, p. 877, 1976.

[93] B. He and X. Yuan, "An accelerated inexact proximal point algorithm for convex minimization," 2010.

[94] M. Schmidt, N. L. Roux, and F. Bach, "Convergence rates of inexact proximal-gradient methods for convex optimization," in *NIPS*, 2011.

[95] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[96] R. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *The Journal of Machine Learning Research*, vol. 6, pp. 1817–1853, 2005.

[97] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.

[98] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient L2,1-Norm minimization," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009, pp. 339–348.

[99] J. Zhou, J. Chen, and J. Ye, *MALSAR: Multi-tAsk Learning via StructurAl Regularization*, Arizona State University, 2011. [Online]. Available: http://www.public.asu.edu/~jye02/Software/MALSAR

[100] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[101] J. Liu, L. Yuan, and J. Ye, "An efficient algorithm for a class of fused lasso problems," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*. ACM, 2010, pp. 323–332.

[102] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[103] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 591–606, 2009.

[104] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.

[105] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, pp. 43–52, 2010, 10.1007/s13042-010-0001-0. [Online]. Available: http://dx.doi.org/10.1007/s13042-010-0001-0

[106] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[107] F. Moosmann, E. Nowak, and F. Jurie, "Randomized clustering forests for image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1632–1646, 2008.

[108] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2169–2178.

[109] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[110] J. Zhou and H. Peng, "Automatic recognition and annotation of gene expression patterns of fly embryos," *Bioinformatics*, vol. 23, no. 5, pp. 589–596, 2007.

[111] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[112] S. Kumar, C. Konikoff, B. Van Emden, C. Busick, K. Davis, S. Ji, L. Wu, H. Ramos, T. Brody, and S. Panchanathan, "Flyexpress: visual mining of spatiotemporal patterns for genes and publications in drosophila embryogenesis," *Bioinformatics*, vol. 27, no. 23, pp. 3319–3320, 2011.

[113] V. Hartenstein, *Atlas of Drosophila Development*. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press, 1993.

[114] S. Kumar, K. Boccia, M. McCutchan, and J. Ye, "Exploring spatial patterns of gene expression from fruit fly embryogenesis on the iphone," *Bioinformatics*, 2012.

[115] J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision Research*, vol. 20, no. 10, pp. 847–856, 1980. [Online]. Available: http://www.sciencedirect.com/science/article/B6T0W-4846HB5-HV/2/76c2432440d609a6f214c88d41a453e5

[116] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *Journal of The Optical Society of America A*, vol. 4, no. 12, pp. 2379–2394, 1987.

[117] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2009.

[118] V. Vapnik, *The nature of statistical learning theory*. Springer-Verlag New York Inc, 2000.

[119] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[120] J. Liu, S. Ji, and J. Ye, *SLEP: Sparse Learning with Efficient Projections*, Arizona State University, 2009. [Online]. Available: http://www.public.asu.edu/~jye02/Software/SLEP

[121] J. Liu and J. Ye, "Moreau-yosida regularization for grouped tree structure learning," *Advances in Neural Information Processing Systems*, vol. 23, pp. 1459–1467, 2010.

[122] J. Friedman, T. Hastie, and R. Tibshirani, "A note on the group lasso and a sparse group lasso," *Arxiv preprint arXiv:1001.0736*, 2010.

[123] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[124] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall, 2012.