SEMI-SUPERVISED AND TRANSDUCTIVE LEARNING ALGORITHMS FOR PREDICTING ALTERNATIVE SPLICING EVENTS IN GENES

by

KARTHIK TANGIRALA

B.E., JNTU, India, 2009

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2011

Approved by:

Major Professor
Doina Caragea

Copyright

Karthik Tangirala

2011

Abstract

As genomes are sequenced, a major challenge is their annotation – the identification of genes and regulatory elements, their locations and their functions. For years, it was believed that one gene corresponds to one protein, but the discovery of alternative splicing provided a mechanism for generating different gene transcripts (isoforms) from the same genomic sequence. In the recent years, it has become obvious that a large fraction of genes undergoes alternative splicing. Thus, understanding alternative splicing is a problem of great interest to biologists. Supervised machine learning approaches can be used to predict alternative splicing events at genome level. However, supervised approaches require large amounts of labeled data to produce accurate classifiers. While large amounts of genomic data are produced by the new sequencing technologies, labeling these data can be costly and time consuming. Therefore, semi-supervised learning approaches that can make use of large amounts of unlabeled data, in addition to small amounts of labeled data are highly desirable. In this work, we study the usefulness of a semi-supervised learning approach, co-training, for classifying exons as alternatively spliced or constitutive. The co-training algorithm makes use of two views of the data to iteratively learn two classifiers that can inform each other, at each step, with their best predictions on the unlabeled data. We consider three sets of features for constructing views for the problem of predicting alternatively spliced exons: lengths of the exon of interest and its flanking introns, exonic splicing enhancers (a.k.a., ESE motifs) and intronic regulatory sequences (a.k.a., IRS motifs). Naive Bayes and Support Vector Machine (SVM) algorithms are used as based classifiers in our study. Experimental results show that the usage of the unlabeled data can result in better classifiers as compared to those obtained from the small amount of labeled data alone. In addition to semi-supervised approaches, we also also study the usefulness of graph based

transductive learning approaches for predicting alternatively spliced exons. Similar to the semi-supervised learning algorithms, transductive learning algorithms can make use of unlabeled data, together with labeled data, to produce labels for the unlabeled data. However, a classification model that could be used to classify new unlabeled data is not learned in this case. Experimental results show that graph based transductive approaches can make effective use of the unlabeled data.

Table of Contents

List of Figures

List of Tables

Acknowledgments

The following thesis, while an individual work, would not have been possible without the help and supervision of several people. I would like to thank all people who have helped and inspired me during my M.S. years.

First and foremost, I am greatly indebted to my adviser, Dr. Doina Caragea. It has been an honor to be her student during my M.S. This thesis would not have been completed without her patience and steadfast encouragement. The instructive and timely comments on chapter drafts represent themselves a course in critical thought upon which I will always draw. The support that she has shown during my thesis work cannot be overstated. I appreciate all her contributions of time and ideas that helped me have a two productive years at KSU and great experience. Dr. Caragea was a fabulous advisor: sharp, perceptive, and mindful of the things that truly matter. Thank you does not seem sufficient but it is said with appreciation and respect.

I am grateful to be a student of Dr. Susan J. Brown and to have her as my M.S. committee member. Her knowledge and ideas in the field of bioinformatics motivated me during my early stages of my M.S. studies. Dr. Brown was always there to listen and give advice. Her insights into biological concepts are very much appreciated. I am also thankful to my M.S. committee member, Dr. Torben Amtoft, for all his support and guidance right from the beginning of my years at KSU. His knowledge in the field of algorithms helped me to gain insights in determining ways to tackle some of the hardest problems.

My deepest gratitude to my parents, Mr. Venkata Gopala Krishna Murthy Tangirala and Mrs. Venkata Satya Kumari Kambhampati, and to my sister, Harika Tangirala, for their love and support at every stage of my life. It is because of their motivation and encouragement that I am able to complete my M.S. and start a Ph.D.

I finally thank my friends, especially Rohit, Ana, Surbhi and Vishal for helping me in the early days of my M.S. and for valuable discussions.

Chapter 1

Introduction

In this chapter, we begin by providing motivation for this work along with a brief problem definition in Sections 1.1 and 1.2, respectively. Then, in Section 1.3, we give a brief overview of the approaches used.

1.1 Motivation

Machine Learning [Mitchell, 1997] is a branch of artificial intelligence focused on the design and development of algorithms for learning models from data. Supervised learning is a type of learning, where a model is learned from labeled data and is used to predict unlabeled data. Predicting a new problem is a difficult task even for humans. However, humans use past experiences to gain knowledge on a particular problem and then they predict it. Similarly, computers need existing known data or examples to capture "past experiences." For supervised learning, the examples are pairs of objects and their corresponding labels, generally referred to as training examples. Using the training data, a machine learning algorithm infers a classification or regression function, a.k.a. model, which is further used to predict new data.

Supervised machine learning has been used in a variety of domains including information retrieval, natural language processing, social networking and bioinformatics. However, the success of supervised learning depends on the availability of large amounts of labeled data. For many application domains, the amount of labeled data is very limited, while large

amounts of unlabeled data are easily available. This motivates the need for semi-supervised learning algorithms [Xiaojin, 2006], which can make use of large amounts of unlabeled data together with small amounts of labeled data to learn models that can accurately predict new, unlabeled data. Similar to semi-supervised learning algorithms, transductive learning algorithms [Xiaojin, 2006] use both labeled and unlabeled data; however, instead of learning a model that can predict new, unlabeled data, transductive algorithms produce labels for the originally provided unlabeled data, but are not able to predict labels for new data. Both semi-supervised and transductive learning algorithms have received a lot of attention in the last few years. They have been successfully used in several application domains including: text classification, natural language processing and sentiment categorization [Nigam et al., 2000], [Joachims, 1999], [Collins and Singer, 1999], [Gupta and Ratinov, 2008], [Dai et al., 2007], [Goldberg and Zhu, 2006].

However, semi-supervised and transductive learning algorithms have not been much studied in the bioinformatics domain, with a few notable exceptions [Kall et al., 2007], [Weston et al., 2006], [Weston et al., 2005]. Given the recent advances in next generation sequencing technologies, large amounts of biological sequence data are produced. Labeling these data can be very expensive. Algorithms that can make use of unlabeled data are greatly needed in bioinformatics.

1.2 Problem Definition

The work in this thesis is focused on the *study of semi-supervised and transductive learning algorithms in the context of bioinformatics classification problems*. Specifically, we consider the problem of *predicting alternative splicing events in genes*. We briefly describe the alternative splicing problem below and further detail it in Section 2.1.2.

Genes undergo transcription and translation in the process of protein synthesis. Splicing is a stage in between transcription and translation, in which the coding regions are separated from the non-coding regions to form mRNA (messenger-RNA). Alternative splicing (e.g.,

exon skipping or intron retention) is a mechanism by which multiple mRNA transcripts are generated from a single gene [Black, 2003]. Thus, alternative splicing can be seen as an important mean for increasing proteome diversity. Alternative splicing is believed to be regulated by splicing factors that bind to regulatory elements, called splicing motifs or enhancers/silencers, present in exons and/or their flanking introns. Furthermore, the lengths of exons and introns are known to be important with respect to alternative splicing prediction.

Therefore, in this work, we aim to learn to discriminate between alternative spliced exons and constitutive exons, and use spicing motifs and length features to represent data instances.

1.3 Overview of the Approaches Used

Traditional approaches to the identification of alternative splicing events have relied on transcriptome-to-genome alignments (see [Hongchao et al., 2009], [Bonizzoni et al., 2008] for reviews). While such approaches can help identify alternative splicing events accurately, they are limited to genomes for which a large amount of transcript data is available, as different transcripts can be found in different tissues or cell types, at different development stages or induced by external stimuli. As a complement to traditional alignment-based approaches, supervised machine learning approaches have been successfully applied to the problem of predicting alternative splicing events (most commonly, prediction of alternatively spliced exons) [Ratsch et al., 2005]. Such machine learning approaches make use of local sequence features to distinguish between alternatively spliced and constitutive exons including [Ratsch et al., 2005], [Dror et al., 2005], [Xia et al., 2010]: lengths of the exon and the flanking introns; exon divisibility by 3; sequence conservation in the exon and in the flanking introns (i.e., conserved motifs).

Lack of labeled data for many, but a few model organisms, makes the application of supervised machine learning algorithms impractical. Therefore, the use of semi-supervised

and transductive learning algorithms that can make use of large amounts of unlabeled data, in addition to small amounts of labeled data, is highly desirable. As discussed above, two main classes of approaches that make use of unlabeled data exist: semi-supervised and transductive learning approaches.

In the semi-supervised learning framework, co-training [Blum and Mitchell, 1998] and EM [Mclachlan and Krishnan, 2009] are two prominent algorithms. In this work, we will explore the use of the co-training algorithm to learn a model for distinguishing between alternatively spliced exons and constitutive exons in the semi-supervised framework. As originally described by Blum and Mitchell [1998], co-training is a two-view iterative learning technique, which uses two or more independent and sufficient feature representations, or views, of the same data, to learn two different classifiers. At each iteration, the training data of each classifier is augmented with the best predictions that the other classifier makes on the unlabeled data. As a consequence, the classifiers improve at each iteration. The final classifiers are used together to predict labels for new data.

In the transductive learning category, we will explore the use of graph based learning algorithms to propagate labels from labeled exons to unlabeled exons. As opposed to semi-supervised learning, no model is learned in this case.

The rest of this thesis is organized as follows: Chapter 2 provides the background information related to this project, both from biological as well as machine learning perspectives. The algorithms which are used for this approach are also discussed in Chapter 2. A discussion of the related work can be found in Chapter 3. Chapter 4 describes in detail the problem addressed in this work and how the approached used are applied to this problem. In Chapter 5 , we describe the data used in our study, the experimental setup, the research questions that are addressed in this work and finally the experiments performed. Chapter 6 presents the results obtained for our experiments.Finally, Chapter 7 provides several directions for future work and summarizes conclusions drawn from this work.

Chapter 2

Background

2.1 Biological Background

2.1.1 What is a Gene?

A *gene* is the hereditary unit of every living organism. Genes are segments of DNA (Deoxyribonucleic acid) consisting of two complementary strands, which are held together by hydrogen bonds. A DNA molecule is made up of four different nucleotides - guanine (G), cytosine (C), adenine (A) and thymine (T). Nucleotide G pairs with C and A with T. A gene consists of exons, representing the protein coding information and intervening sequences called introns. The beginning of an intron is marked by a *donor site*, while the end of the intron is marked by an *acceptor site*. Figure 2.1 shows a simplified picture of the gene structure, specifically the components of a gene which are relevant to this work.

Genes are responsible for growth and development, and they play a crucial role in protein synthesis. Protein synthesis consists of several stages. Gene transcription is the first stage in protein synthesis. During transcription, the RNA polymerase traverses across the gene to generate the pre-mRNA, which is the unprocessed mRNA. Transcription starts at transcription start site and ends at transcription stop site. The pre-mRNA is then transformed into mRNA during splicing, a process which occurs in between the phases of transcription and translation. Finally, during the translation phase, the mRNA is transformed into a protein. Translation starts at translation start site and ends at translation stop site. The whole

process is known as the Central Dogma of Molecular Biology and is shown in Figure 2.2. The mRNA contains the protein coding information. Figure 2.3 shows in more detail the splicing step in the protein synthesis process. In this picture, the pre-mRNA corresponding to a gene consisting of four exons is transformed into mRNA by splicing out the introns. As can be seen in the figure, all exons are retained, and all introns are skipped.

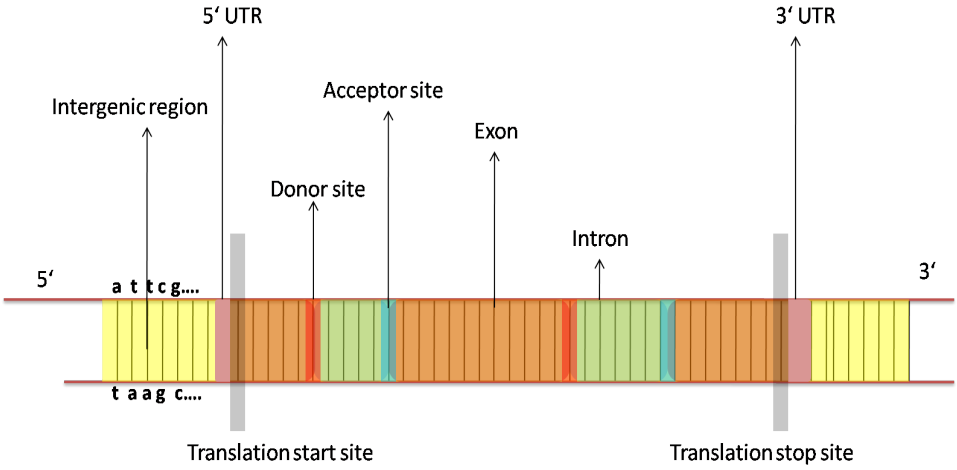


Figure 2.1: *Simplified gene structure: the components of a gene relevant to this work are highlighted.*

2.1.2 Alternative Splicing

For many years, it was believed that one gene corresponds to one protein. However, the discovery of alternative splicing [Black, 2003], have provided an explanation for protein diversity.

Alternative splicing is a mechanism responsible for the formation of multiple proteins from a single gene. That means that several mRNA *isoforms* can be generated from the pre-mRNA corresponding to a gene. Several types of alternative splicing events are known to exist. A skipped exon, a retained intron, alternative 5‘donors, alternative 3‘ acceptors and mutual exclusive exons as discussed in [Black, 2003] are the possible alternative splicing events.

In this thesis, we will study one such kind of alternative splicing event, precisely we

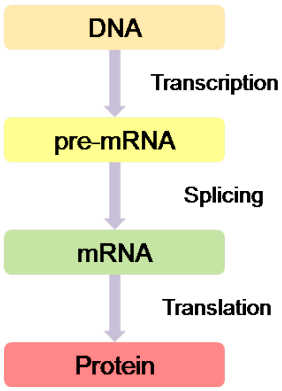


Figure 2.2: *Central Dogma of Molecular Biology: main steps involved in protein synthesis.*

consider *alternatively spliced exons*, i.e. exons that are included in some transcripts and skipped from other transcripts. Exons that appear in all transcripts are called *constitutive exons*. Figure 2.4 shows an example of an alternative splicing event, where exon 3 is retained in one transcript and skipped in another transcript, therefore alternatively spliced. Exons 1, 2 and 4 appear in both transcripts, therefore they are constitutive exons. Recent studies have found that approximately 95% of human genes are alternatively spliced [Pan et al., 2008]. Predicting alternative splicing events is important as they contributes significantly to protein diversity.

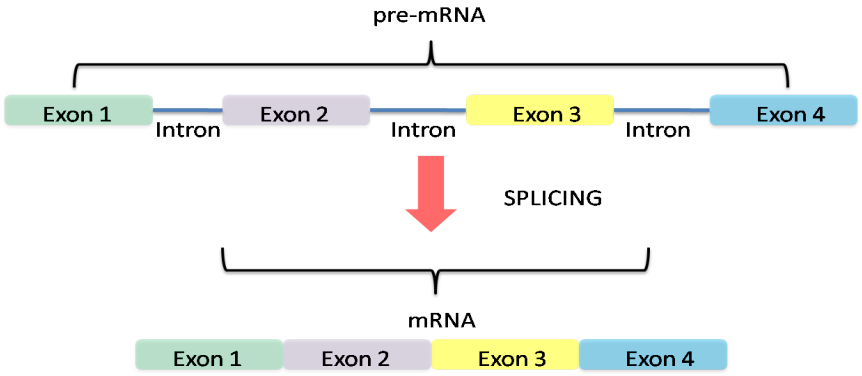


Figure 2.3: *Splicing phase of gene resulting in the formation of mRNA.*

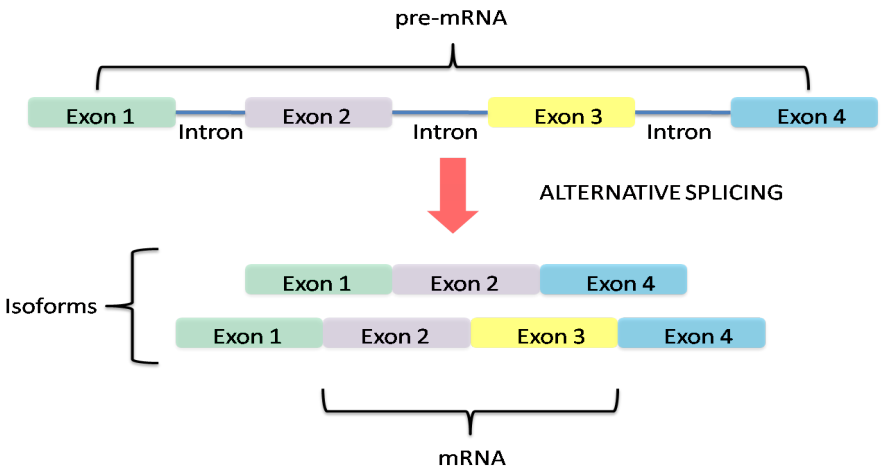


Figure 2.4: *Alternatively spliced versus constitutive exons: Exon 3 is skipped in the first transcript and retained in the second transcript, therefore, alternatively spliced. Exons 1, 2 and 4 are constitutive exons.*

2.2 Machine Learning Background

2.2.1 Learning Frameworks

Supervised learning uses all the available labeled data to train or learn a model. The model is further used in predicting new unseen data. Supervised learning cannot use unlabeled data for training the model and performs well if we have sufficient labeled data. Scarce labeled data can degrade the performance of supervised learning. Figure 2.5 shows an abstract working of supervised learning.

Semi-supervised learning uses the knowledge from both labeled and unlabeled data to learn models that can be used to predict the unseen data. Semi-supervised learning is efficient when we have very little labeled data and large amount of unlabeled data. Figure 2.6 shows an abstract working of semi-supervised learning. Co-training and EM are two prominent approaches under the semi-supervised learning framework.

Transductive learning is similar to semi-supervised learning in using the knowledge of unlabeled data. However, transductive learning does not learn a model for predicting new unseen data. Instead, it classifies all the initially available unlabeled data. Thus,

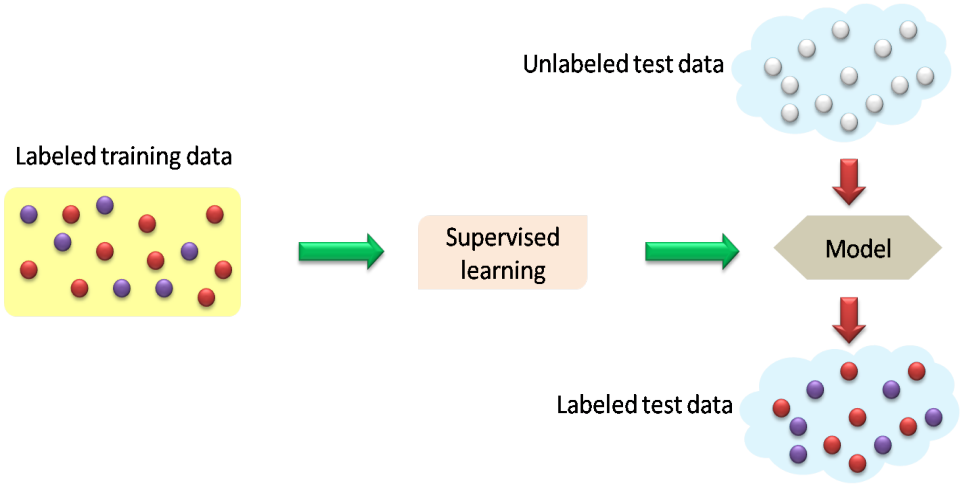


Figure 2.5: *Supervised learning overview*

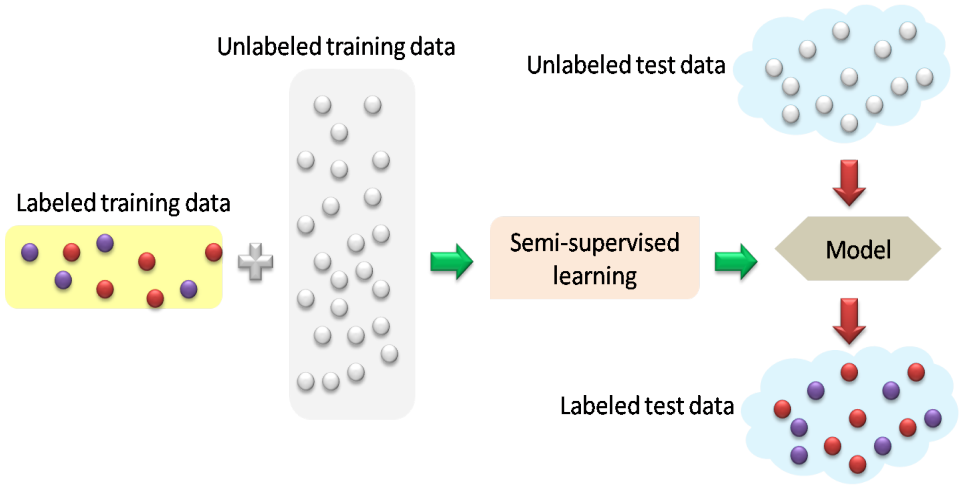


Figure 2.6: *Semi-supervised learning overview*

this learning framework uses the labeled data as well as the unlabeled data to predict the unlabeled data. Figure 2.7 shows an abstract working of transductive learning. Graph-based approaches and transductive SVM are some examples of approaches under this category.

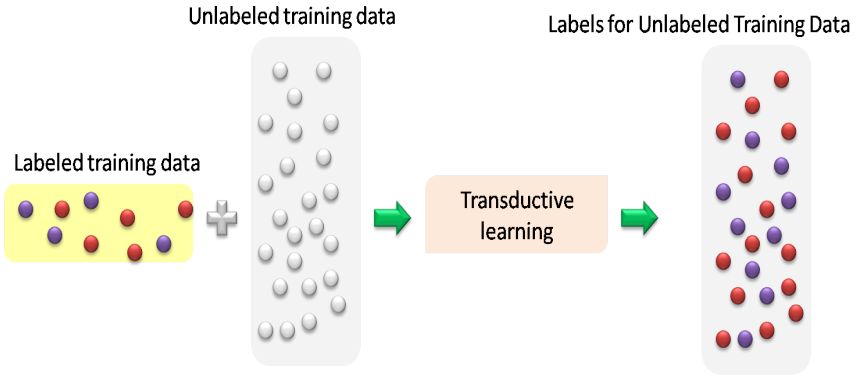


Figure 2.7: *Transductive learning overview*

2.2.2 Co-Training based Semi-Supervised Learning

Co-training is a semi-supervised learning algorithm, which has seen great research interest in recent days. Co-training has been shown experimentally to work well in various domains. The goal of this work is to study the co-training algorithm in the context of predicting alternative splicing in genes. This algorithm was first designed and implemented by Blum and Mitchell [1998]. Co-training uses the information of the labeled and unlabeled data to learn accurate models for predicting new unseen data.

The main idea of co-training is to represent data using two (or more) views, where each view is defined by a subset of features that can be used to describe the data. The views are assumed to be independent and sufficient. Two different classifiers are learned from the two views and their predictions on the unlabeled data are used to iteratively improve each other, as explained below.

Co-Training Implementation ([Blum and Mitchell, 1998], Figure 2.8)

The process is initiated by defining two different views V1 and V2 for the data available. Using these views, we represent labeled as L1 in view V1 and L2 in view V2. Similarly,

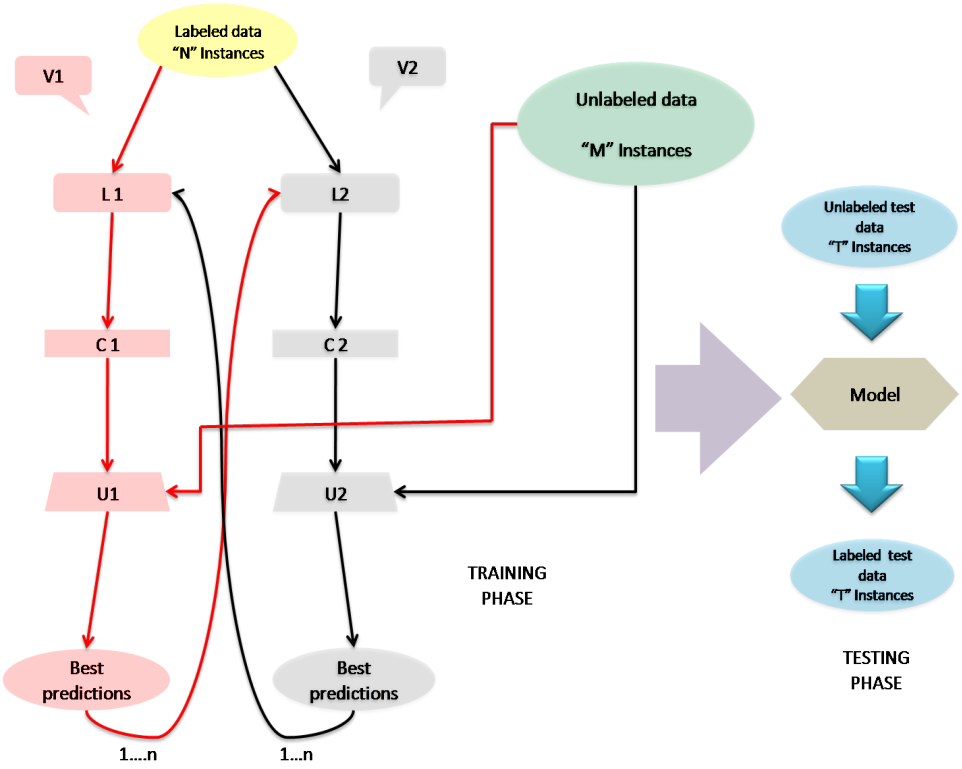


Figure 2.8: *Brief overview of the co-training algorithm. Classifiers C1 and C2 can be any base classifiers, e.g. Naive Bayes and SVM classifiers.*

unlabeled data is also represented in the two views: U1 in view V1 and U2 in view V2. Using the known labeled data L1 and L2 (training sets), two different classifiers C1 and C2 corresponding to the two views V1 and V2 are learned.

These classifiers then predict the unlabeled data U1 and U2 corresponding to the two views. As a result, all the unlabeled instances are classified with a probability. We consider the best predicted instances from U1 and add them to the training set of C2. Similarly, we take the best predictions from U2 and add them to the training set of C1. As soon as we add a particular instance to the training set, we delete the instance from the unlabeled instance set. After adding the best predictions to the training sets of opposite views, we rerun the whole process. This can be done for a certain number of iterations or until all the unlabeled data is utilized by the algorithm. At the end of this process all, we have two classifiers that can be used together to predict new test data. The new samples are classified by considering the predictions of both classifiers jointly. This is done by taking into account the probability of predictions. More precisely, we add the probabilities for each class obtained from the two different classifiers corresponding to the two views. The final label is assigned based on the most probable class. Figure 2.9 shows the pseudocode for co-training.

Adding the best predictions to the original training data will improve the classifiers at each iteration. Intuitively, the instances which are best predicted by one classifier in one view, may not be well predicted by the classifier in the other view. Transferring information to other classifier in this way can increase the performance of both classifiers. Indeed, experimental results show that this kind of knowledge transfer between two classifiers using the unlabeled data can be very successful in several domains.

We explore the use of this particular algorithm on biological data and study its ability to outperforms supervised algorithms trained on small amounts of labeled data. One reason for using this particular algorithm is because of its capability to make use of information from two independent views of data. For our problem, the data available can be represented using various combinations of features.

Co-training pseudo code

INPUT: Labeled data, unlabeled data and test data.

Output: Labels for all the test data

Algorithm:

Step1 : Two views V1 and V2 are defined for the available data. L1 , L2 corresponding to labeled data and U1, U2 corresponding to unlabeled data are obtained according to the two views.

Step2: Learn classifiers C1 and C2 using L1 and L2.

Step 3: Use C1 and C2 to predict U1 and U2.

Step 4: Take the best predictions from U1, add to L2. Similarly, take the best predictions from U2 and add to L1.

Step 5: Go to Step 2. Do this iteratively either for certain number of iterations or until the completion of unlabeled data.

Step 6: Use C1 and C2 to predict the test data.

Figure 2.9: *Pseudocode for the co-training algorithm*

2.2.3 Graph Based Transductive Learning

As an alternative to semi-supervised learning using co-training, we will also study a graph-based transductive approach. Graph based transductive learning also uses the knowledge from both labeled and unlabeled data. Unlike co-training, the graph-based approach restricts its predictions only to the unlabeled data available initially. Instead of building a model which can classify new instances, the goal is to classify all the unlabeled data. Graph based transductive algorithms define a graph, where the nodes correspond to the labeled and unlabeled data instances and the edges between the nodes are weighted based on the similarity between the corresponding instances. The goal is to define a function, which is smooth over the whole data. Our work uses the local and global consistency graph based approach introduced by Zhou et al. [2004]. We will review this algorithm in what follows.

Given the data, our primary task is to construct a weight matrix, which captures the similarity between every pair of instances (or nodes), with rows and columns corresponding to instances, including both labeled and unlabeled instances. Let us suppose that we have N labeled instances $\{x_1, x_2, ...x_N\}$ and M unlabeled instances $\{x_{1+N}, x_{2+N}, ...x_{M+N}\}$ categorized into c classes from the set $C = \{c_1, c_2, ...c_c\}$. We construct a matrix $W_{i,j}$ of size $(N + M) \times (N + M)$, where each cell represents the similarity between the instances corresponding to that particular cell. Let the matrix $F = \{{f_1}^T, {f_2}^T, ...{f_{N+M}}^T\}^T$ be a classification matrix, where the number of rows is equal to the number of instances and the number of columns is equal to the number of classes. Thus, the size of F is $(M + N) \times c$.

Matrix F is updated at every step using the iterative equation given by

$$F(t + 1) = \alpha \cdot S \cdot F(t) + (1 - \alpha) \cdot Y \tag{2.1}$$

(Matrices Y and S are defined further in this section.)

At each iteration, each instance gains some information from its neighbors. As a result, the matrix F improves at each iterations and converges to a fix point. Convergence is achieved when the current iteration will produce the same result as the previous iteration.

The detailed working of the algorithm is as follows:

Let Y be a matrix with the number of rows equal to the number of instances and the number of columns equal to the number of classes. $Y_{i,j}$ is set to 1 if the instance x_i belongs to class c_j and is set to 0 in all other cases. Unlabeled instances will have all the columns set to 0. That means,

$$Y_{i,j} = \begin{cases} 1, \text{ if } x_i \text{ is labeled as } c_j \\ 0, \text{ otherwise} \end{cases} \tag{2.2}$$

Let W be the weight matrix, where $W_{i,j}$ represents the similarity between the instances x_i and x_j. We use a *Gaussian kernel* to calculate the distance between two instances, thus generating the weight matrix. Precisely, the values of W are calculated as follows:

$$W_{i,j} = \begin{cases} 0, \text{ if i} = \text{j} \\ \exp\left(\dfrac{-\|x_i - x_j\|^2}{2\sigma^2}\right), \text{ if i} \neq \text{j} \end{cases} \tag{2.3}$$

Using the weight matrix W, we construct a normalized weight matrix S, by symmetrically normalizing W (the normalization is necessary for the convergence of the function F). The normalized weight matrix is computed as:

$$S = D^{-1/2} W D^{-1/2} \tag{2.4}$$

where D is a diagonal matrix with each diagonal element equal to the sum of all the elements of the corresponding row:

$$D_{i,j} = \begin{cases} \sum_{i=0}^{M+N} x_i, \text{ if i} = \text{j} \\ 0, \text{ if i} \neq \text{j} \end{cases} \tag{2.5}$$

After generating all the matrices, the next step is to calculate the following function:

$$F(t) = (\alpha S)^{t-1} Y + (1 - \alpha) \sum_{i=0}^{t-1} (\alpha S)^i Y \tag{2.6}$$

which is proved to be convergent [Zhou et al., 2004] and the point of convergence is

$$F^* = (I - \alpha S)^{-1} Y, \tag{2.7}$$

Zhou et al. [2004] also define two variants of this particular function. Let $P = D^{-1}W$. As can be seen from Equation (2.4), P is similar to S. Therefore, we can replace S in the formula above with P or P^T, which gives rise to the following two variants:

- Variant 1: $F^* = (I - \alpha P)^{-1}Y$
- Variant 2: $F^* = (I - \alpha P^T)^{-1}Y$, which is equivalent to $F^* = (D - \alpha W)^{-1}Y$.

The values of the matrix F^* can be seen as weights representing the confidence with which a particular instance (row) belongs to a particular class (column). Using this matrix, we assign labels to the unlabeled instances using the following formula: $y_i = \arg\max_{j \leqslant c} {F^*}_{ij}$. The pseudocode for the graph-based transductive approach is shown in Figure 2.10.

Graph-based transductive approach pseudo code

Input: Instances of both labeled and unlabeled data represented as feature vectors.

Output: Labels for all the unlabeled instances.

Algorithm:

Step 1: Calculate the weight matrix using Equation 2.3.

Step 2: Compute matrix Y by assigning 1 to all the labeled instances for the corresponding columns and 0 to the rest of the instances according to Equation 2.2.

Step 3: Compute matrix S using Equation 2.4 and P as defined.

Step 4: Compute F using the consistency method as defined in Equation 2.7 as well as from variant methods 1 and 2 defined.

Step 5: The weight of a cell in matrix F represents the similarity of a particular instance (Row) to a particular class (Column).

Step 6: We assign the labels by $c_i = argmax_{j \leqslant c_i} {F^*}_{ij}$

Figure 2.10: *Pseudocode for the graph-based approach.*

Chapter 3

Related Work

Semi-supervised and transductive learning algorithms have received a lot of attention in the last few years, as large amounts of unlabeled data have become available for classification tasks. We will review some of the work on co-training in Section 3.1 and work on graph based transductive approaches in Section 3.2. Section 3.3 deals with the previous work on semi-supervised learning algorithms for bioinformatics. Finally, in Section 3.4 we will review previous research on the problem of predicting alternative splicing events in genes.

3.1 Co-Training Based Semi-Supervised Learning

As mentioned before, co-training is a semi-supervised learning algorithm which utilizes the knowledge of labeled data together with unlabeled data through an iterative transfer of knowledge between two different classifiers. Co-training was first introduced by Blum and Mitchell [1998]. In this paper, the authors applied co-training to the problem of classifying web-pages as course home pages and compared the results with the supervised implementation of Naive Bayes algorithm. The results showed that co-training has a smaller error rate when compared to the supervised implementation. Nigam and Rayid [2000] studied the effectiveness of co-training by applying it to various real world data sets such as WebKB course dataset, News 2 $\times$ 2 dataset and News 5 dataset. In many cases, co-training outperforms the supervised learning algorithms and also the EM algorithm. Svetlana and Matwin [2001] evaluated the performance of co-training when various base classifiers such as Naive

Bayes and SVM are used. The results showed that, for the problem of classifying emails, co-training with SVM as a base classifier outperforms co-training with Naive Bayes multinomial as a base classifier. The results also showed that, for unbalanced data problems, balancing the training set has as effect an increase in the performance of co-training. Xu et al. [2009] worked on the problem of predicting protein localization using a co-forest approach, which is similar to co-training. The approach enhanced the state-of-the-art prediction results of SVM classifiers by 10%.

3.2 Graph-Based Transductive Learning

The goal of any graph based transductive learning is to create a smooth function over the intrinsic graph structure obtained by combining known labeled and unlabeled data points. Graph based transductive approaches make the assumption that nearby data points in a high density region share a common label (called *smoothness assumption*). Furthermore, data points within the same cluster also share a common label (called *cluster assumption*). Zhou et al. [2004] defined a smooth classification function with respect to the points corresponding to labeled and unlabeled data. Their work is similar to the implementation of spreading activation networks. Initially, the source nodes are assigned labels (based on the labeled data). The labels are then propagated across the unlabeled data using the knowledge of similarity between the neighboring instances. The similarity between instances is computed using a kernel function. An iterative function, which captures the information gain for each unlabeled instance on every iteration is defined. After convergence, unlabeled instances will be assigned weights based on their similarity with a particular class. The more similar the instances are to a particular class, the higher is the weight related to that class. This approach dominated supervised approaches on several toy problems, text classification and also digit recognition. Fei and Changshui [2008] defined a similar approach of deriving a smooth function, except that they used the Laplacian transformation of the weight matrix. The function propagates across the graph using the linear neighborhood,

maintaining the smoothness. The results of this approach when applied on a toy problem, text classification and digit recognition outperformed the supervised baselines. Jebara and Chang [2009] defined an improved version of the graph based approach, where they consider the sparseness of the matrix by using two approaches. One of them is k-NN, the other is b-matching. B-matching is an extension to the k-NN approach of labeling, where they restricted the number of edges for a particular node to be 'b'. Jebara and Chang [2009] also considered the concept of graph edge re-weighting in their approach. In this thesis, we use the local and global consistency method and its two variants described in [Zhou et al., 2004] for a bioinformatics problem.

3.3 Semi-Supervised Learning in Bioinformatics

Supervised machine learning has been successful in solving many biological problems [Larraaga et al., 2006]. In the recent years, we have also seen several applications of semi-supervised learning algorithms to bioinformatics. Weston et al. [2005] provided an approach of using semi-supervised learning for the problem of classifying proteins. The authors used various cluster kernels along with SVM to classify proteins. Weston et al. [2005] were successful in classifying proteins by using the little labeled data along with unlabeled data under semi-supervised learning framework. Kall et al. [2007] designed a tool named "Percolator", which uses semi-supervised machine learning algorithms for differentiating correct and incorrect spectrum identifications (problem of assigning peptides to spectra). Weston et al. [2006] provided a tool "RankProp", which uses the global network structure of similarity relationships among proteins in a database by performing diffusion operation. Weston et al. [2006] extended the unsupervised old system with a semi-supervised structure, which uses the labeled data. This tool has been successful and also proved to be better than local network search algorithms such as PSI-BLAST. Besides these, several other semi-supervised learning approaches were applied on biological problems. However, there is no study on predicting alternative splicing events in genes using semi-supervised learning algorithms.

3.4 Alternative Splicing Using Machine Learning

Predicting alternative splicing is an important problem in biology. Several approaches have been proposed for predicting alternative splicing events in genes. Particularly in the field of machine learning, various approaches have been proposed under the category of supervised learning. Ratsch et al. [2005] have proposed an approach of predicting alternative splicing in *C.elegans* by using SVM under the supervised learning scenario. In the paper, the authors used features derived from the lengths of exons and introns in addition to the weighted degree kernel which captures sequence similarities. The average AUC obtained with this approach is around 90%. Dror et al. [2005] applied SVM under supervised scenario for predicting alternative splicing events in humans. The authors used features derived by considering the length of exon and introns along with conservational information between human and mouse, as well as biologically significant motifs such as upstream and downstream intronic sequence motifs. However, their approach restricted the predictions of alternative splicing events to those which are common in humans and mouse as the approach is relying on the conservational features. This implementation has a true positive rate of 50% at a false positive rate of 0.5%. Xia et al. [2010] also used SVM in a supervised framework for predicting alternative splicing events. Sequence dependent features were used in their approach for predicting alternative splicing. Features based on GC content, exonic splicing enhancer motifs, intronic regulatory splicing motifs and various other features which are just based on the sequences unlike previously defined features which are based on alignments and conservations have been used for predicting the alternative splicing events. We use semi-supervised and transductive learning algorithms for the problem of predicting alternative splicing events in genes. Using semi-supervised learning has the advantage of learning from very little labeled data together with unlabeled data.

To the best of my knowledge, there is no research on applying semi-supervised learning algorithms in predicting the events of alternative splicing. Our approach is an attempt to use semi-supervised learning approaches to predict the alternative splicing events in genes.

Chapter 4

Problem Definition and Approaches

This chapter describes the task of predicting alternative splicing events in genes, specifically alternatively spliced exons in genes from the organism *C.elegans*, and the approaches used in this work. Our primary goal is to study how predictive algorithms work on biological data when the amount of labeled data is very small, as compared to the amount unlabeled data. Thus, in our work, we mainly focus on applying semi-supervised as well as transductive learning algorithms to the problem of predicting alternative splicing.

We begin this chapter by describing the alternative splicing prediction problem and the format of the original data. Following this, we describe the features used to represent instances in a vectorial format in Section 4.2. Section 4.3 describes the machine learning algorithms, which are used as base classifiers for performing experiments using co-training. At last, Section 4.4 describes how co-training and graph-based transductive learning approaches are used with various feature set combinations and various classifiers to predict alternatively spliced exons.

4.1 Alternative Splicing Prediction Problem

We study the application of co-training and graph-based approaches to the problem of predicting alternatively spliced exons. In other words, our task is to classify an exon as either alternatively spliced or constitutive. Thus, each instance corresponds to an exon. As the flanking introns are also relevant with respect to the prediction problem considered,

an instance is given by a sequence consisting of the upstream intron, exon of interest and the downstream intron. However, most signals predictive of alternative splicing are found around the splice sites; therefore, we represent an instance using DNA sequence fragments centered around the donor and acceptor sites. Precisely, we consider 200 base pairs on each side of the splice sites and we end up with sequences that are 802 base pairs long. The regions around the acceptor and donor sites are separated by "@". Figure 4.1 shows examples of two instances, one corresponding to an alternatively spliced exon and the other to a constitutive exon.

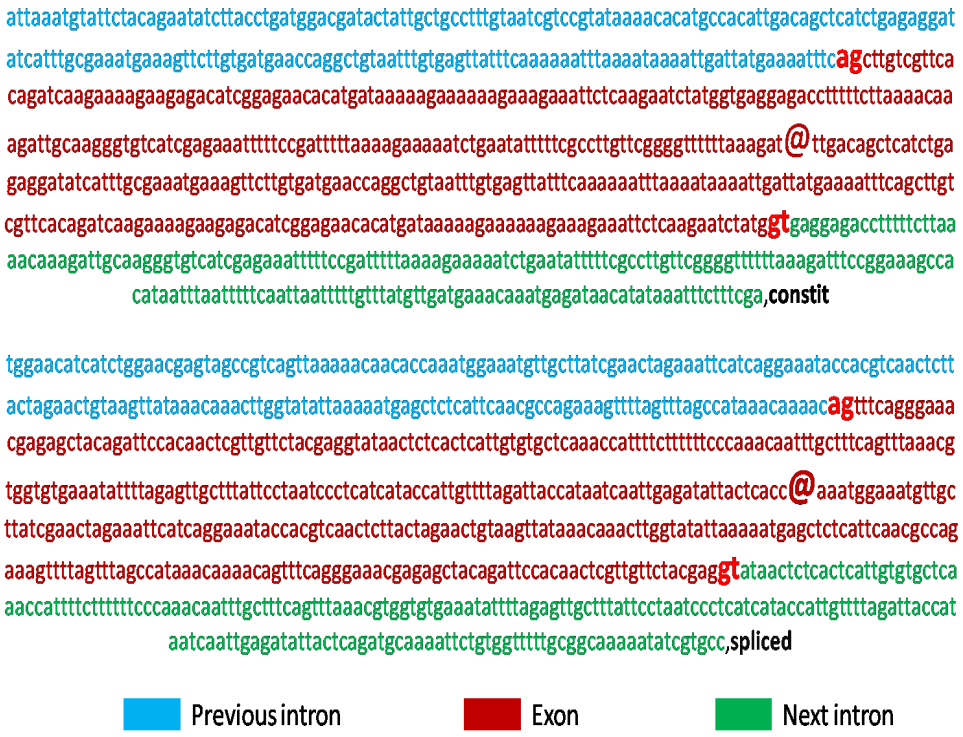


Figure 4.1: *Instances belonging to the spliced and constitutive classes, respectively.*

4.2 Feature Sets Used

Many machine learning algorithms assume data to be represented in the form of feature vectors, which means that we need to construct features for our instances. Supervised

learning of alternative splicing has shown that the lengths of an exon and its flanking introns can be used as predictive features. In addition to length features, biological motifs found in intronic and exonic regions, are also predictive for alternative splicing. Therefore, we in this work, we will represent instances using length and motif features, as described below.

4.2.1 Length Features

The lengths of spliced exons and their flanking introns are generally different from the lengths of the constitutive exons and their flanking introns [Dror et al., 2005]. As a consequence, length information can be used to discriminate between the two classes of exons. Ratsch et al. [2005] defined a feature representation which captures the length information, by considering 30 logarithmically spaced bins. Given a sequence x (which could be an exon or an intron), the value in each bin is calculated according to the following formula:

$$[f(x)]_j = \begin{cases} 1, \text{ if } l(x) \leq L_j, \\ \frac{l(x)-L_j}{L_{j+1}-L_j}, \text{ if } L_j \leq l(x) \leq L_{j+1}, \\ 0, \text{ if } otherwise. \end{cases} \tag{4.1}$$

where $l(x)$ represents the length of sequence x, and L_j and L_{j+1} define the jth bin. We have a total of 90 length features corresponding to the lengths of an exon and its flanking introns. Furthermore, we use 15 more features to indicate which of the three frames of an exon contain at least a stop codon. The length features corresponding to the instance belonging to the class "spliced" of Figure 4.1 are shown in Figure 4.2.

4.2.2 Exonic Splicing Enhancers (ESE motifs)

Splicing regulators are motifs which are highly responsible for the occurrence of alternative splicing events. These splicing regulators can occur in both exons and introns. Enhancing spicing regulators that occur in exons are called ESE motifs. In this work, we used a set of 77 ESE motifs (hexamers, i.e., sequences of length 6 bp) derived by Xia et al. [2010]. The

1.000000,1.000000,0.409253,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.0
00000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.0000
00,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,
0.000000,0.000000,1.000000,1.000000,1.000000,1.000000,1.000000,1.000000,1.000000,1.0
00000,1.000000,1.000000,1.000000,1.000000,1.000000,1.000000,1.000000,1.000000,1.0000
00,1.000000,0.587709,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,
0.000000,0.000000,0.000000,0.000000,1.000000,1.000000,1.000000,1.000000,1.000000,1.0
00000,1.000000,1.000000,1.000000,1.000000,1.000000,1.000000,1.000000,1.000000,1.0000
00,1.000000,1.000000,1.000000,1.000000,1.000000,1.000000,1.000000,1.000000,1.000000,
1.000000,0.501929,0.000000,0.000000,0.000000,0.000000,1.000000,0.000000,0.000000,0.0
00000,0.000000,0.000000,0.000000,0.000000,0.000000,1.000000,0.000000,0.000000,0.0000
00,0.000000,1.000000, spliced

Figure 4.2: *Length features for the spliced instance shown in Figure 4.1*

ESE motifs represent hexamers which occur more frequently in exons than in introns. We run a sliding window to obtain the counts of the 77 ESE motifs in a particular sequence.

Figure 4.3 shows the approach used in converting the motif features into the numeric count representation. Given a set of motifs and a sequence of nucleotides, we compute the frequency of occurrence of each motif in the sequence. We use a sliding window of a fixed size (based on the size of the motif) to compute the frequency of that particular motif in the sequence.

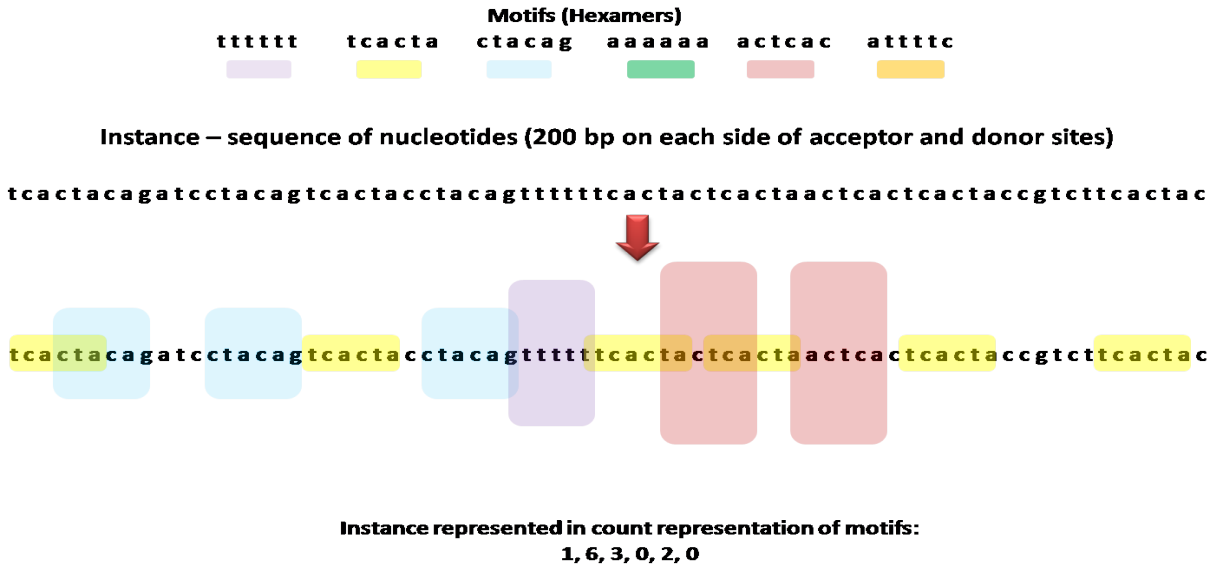


Figure 4.3: *Count representation of motifs.*

Figure 4.4 shows the set of 77 ESE motifs used in this work. For the input data in Figure 4.1, the count representation using these motifs is shown in Figure 4.5.

ESE MOTIFS:

aagttt, accacc, acgatg, agtaag, taagtt, caccac, cagcag, ccacca, cggcaa, gacgac, gtaagt, gtgagt, atatat, attttt, acgatg, acgaca, tttttt, tttcag, ttcagg, tcatca, tcagtg, tcgaaa, catcac, cagtga, cgaaag, gatgat, gacgac, gctcca, aaaagc, aagatg, aagcaa, atggaa, ttggaa, tcaaat, tctcaa, tggaaa, tggaat, caattt, cttgga, gaattc, gctcaa, aaaaga, agaagc, tttgaa, ttgaag, tgaaga, tggaaa, ctggaa

Figure 4.4: *ESE motifs used to capture exonic information related to alternative splicing.*

0,1,1,1,2,0,0,0,0,1,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0
,0,1,1,1,0,1,0,0,0,0, spliced

0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,2,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,2
,0,0,0,0,0,0,0,0,0,0, constit

Figure 4.5: *The ESE count representation for the two instances from Figure 4.1*

4.2.3 Intronic Regulatory Sequences (IRS motifs)

Intronic regulatory sequences (IRS motifs) are regulatory splicing motifs which frequently occur in the intronic regions of genes. We used the set of IRS motifs that was previously used in [Xia et al., 2010]. These motifs have been derived based on the observation that intronic sequences that are relevant for alternative splicing are highly conserved among closely related species. To form the set of IRS motifs, we combined both the upstream and downstream motifs and removed the duplicate motifs. This resulted in a total of 165 IRS motifs which are assumed to be informative for alternative splicing prediction. As in the case of the ESE motifs, we consider a count representation for the IRS motifs in every sequence. This will give rise to a vector of length 165 per sequence.

4.3 Base Classifiers Used with Co-Training

- Naive Bayes (NB) and Naive Bayes Multinomial algorithms (NBM):

  NB and NBM are probabilistic generative learning algorithms, which make the as-

sumption that features are independent given the class. Under this assumption, the probability of an instance given a class can be written as the product of probabilities of features given the class. Once we calculate the probability of an instance given the class, we can label the instance with the class which is having the maximum posterior probability. NBM differs from NB based on the type of the features that they use. NB works with discrete or continuous features. NBM works with count representations. Thus, NB can be used for length features, while NBM can be used for count representations of motifs. Both NB and NBM algorithms perform well on text classification problems, using a binary or count representation, respectively. Here, we explore the usefulness of these algorithms as base classifiers for co-training, when learning to predict alternatively splicing events in a semi-supervised framework.

- Support Vector Machine (SVM) algorithm:

  SVM is a discriminative learning algorithm used especially in binary classification and regression problems [Corinna and Vladimir, 1995]. If the data is linearly separable, SVM works by constructing a hyperplane that separates the positive class from the negative class. If the data is not linearly separable, SVM is using the kernel "trick" to map data from a lower dimensional space to a higher dimensional space, where the data becomes linearly separable, and a separating hyperplane is identified in that space. SVM can associate a confidence value with its prediction for an instance, based on the distance from the instance to the separating hyperplace. SVM has been successfully used to predict alternative splicing in a supervised framework. Here, we explore its usefulness as a base classifier for co-training, in a semi-supervised framework.

4.4 Approaches Used

Given the features described in Section 4.2 and the base classifiers described in Section 4.3, our task is to use these resources efficiently to obtain the best possible predictions from co-training and graph-based transductive approaches. This section describes the details of the

application of co-training and graph-based approaches to the problem of alternative splicing prediction.

4.4.1 Semi-Supervised Learning Using Co-Training

As discussed in Section 2.2.2, co-training needs two views (corresponding to two sets of features) for learning a model. Given the features described above, we can construct many combinations of features. However, as co-training works best when the views are independent and sufficient, we experiment with the following combinations of features: IRS versus Length, IRS+ESE versus Length, ESE versus Length, and IRS versus ESE. We also experiment with various combinations of base classifiers, according to the type of features used in a view. For example, if we are working with Length features, we can only have SVM and NB as base classifier. We cannot use NBM as this algorithm requires a count representation. However, if we are using motifs (IRS and ESE), we can use either NBM or SVM as base classifiers.

4.4.2 Graph-based Transductive Learning

Graph-based transductive learning requires sequences represented according to a set of features. To identify the best set of features for graph-based approaches, we experiment with the same combinations of features as in the case of co-training. In other words, we represent instances using IRS and Length features, IRS+ESE features, etc. Distance between features is calculated using a Gaussian kernel. This will result in an original distance matrix, needed for further computations. The graph-based procedure will produce labels for all the unlabeled data.

Chapter 5

Experimental Setup

We begin this chapter by describing the dataset used for evaluating the semi-supervised and transductive approaches in Section 5.1. Following this, in Section 5.2, we will discuss the the 5-fold cross validation framework used in the experiments and the implementation of co-training and graph-based transductive approaches under the 5-fold cross validation framework. Further in this chapter, in Section 5.3, we list the research questions that we are addressing through this project. Finally, we describe the experiments that we performed in Section 5.4.

5.1 Data

Our task is to categorize an exon as either alternatively spliced or constitutive. We use the alternatively spliced exon dataset from [Ratsch et al., 2005]. The procedure that they followed in generating the dataset is as follows:

- First, all the available *C.elegans* ESTs and cDNAs from various sources are gathered together.
- After removing duplicates, the ESTs are aligned to the genomic DNA using BLAT.
- Only sequences which are aligned with a percentage of 90 or higher are used further (weaker alignments are filtered out in order to maintain accuracy).

- The result will be ESTs aligned to the exonic regions of the gene as shown in Figure 5.1. This enables us to find the exonic regions of the genes.

- Exons that appear in one transcript, but do not appear in other overlapping transcripts, are labeled as alternatively spliced exons. Exons that appear in all transcripts are labeled as constitutive exons. Figure 5.1 shows the alignment of two transcripts to the same genomic region. Here, exon E3 is labeled as alternatively spliced, while exons E1, E3 and E4 are labeled as constitutive.

- Exons which have flanking introns on both 3' and 5' ends are collected along with their labels (spliced or constitutive).

- Finally, a collection of 487 exons which are categorized as alternatively spliced and 2531 exons which are categorized as constitutive is developed.

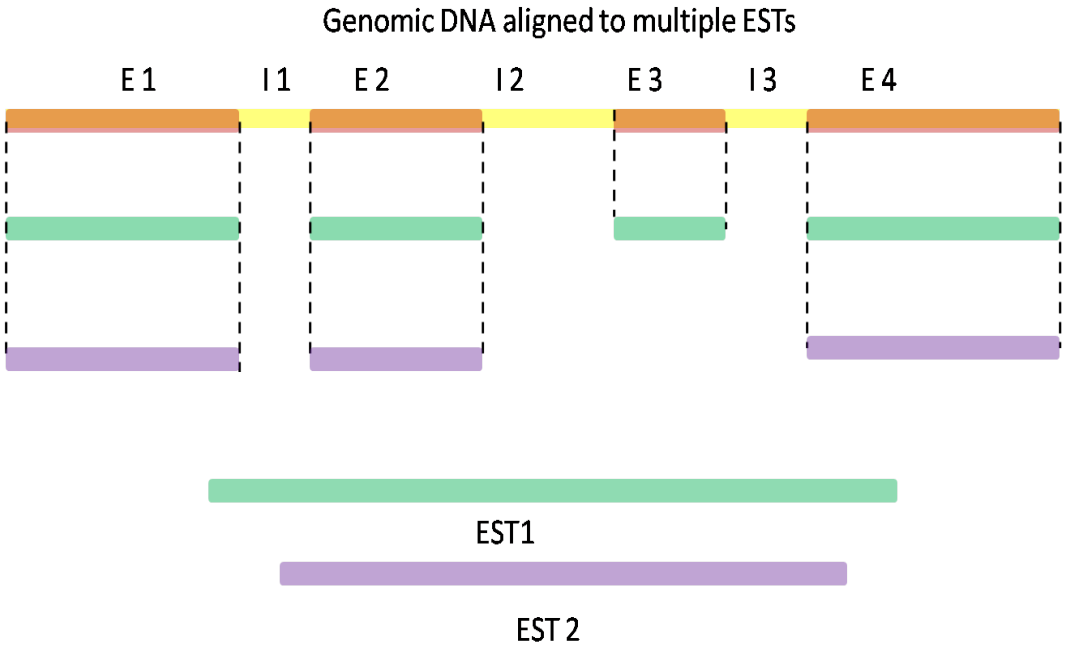


Figure 5.1: *Alignment of multiple ESTs to the same genomic sequence. Exon E3 is identified as alternatively spliced, while exons E1, E2, E3 are considered to be constitutive.*

We used a ±200 window around the acceptor and donor regions for every example in our dataset. The process of generating each instance from the original sequence is shown in Figure 5.2. As a result, we end up with instances that can be seen as sequences of 802 nucleotides labeled with the class (spliced, constit) to which they belong. Next, these instances are represented as feature vectors using length features, ESE motifs and IRS motifs.

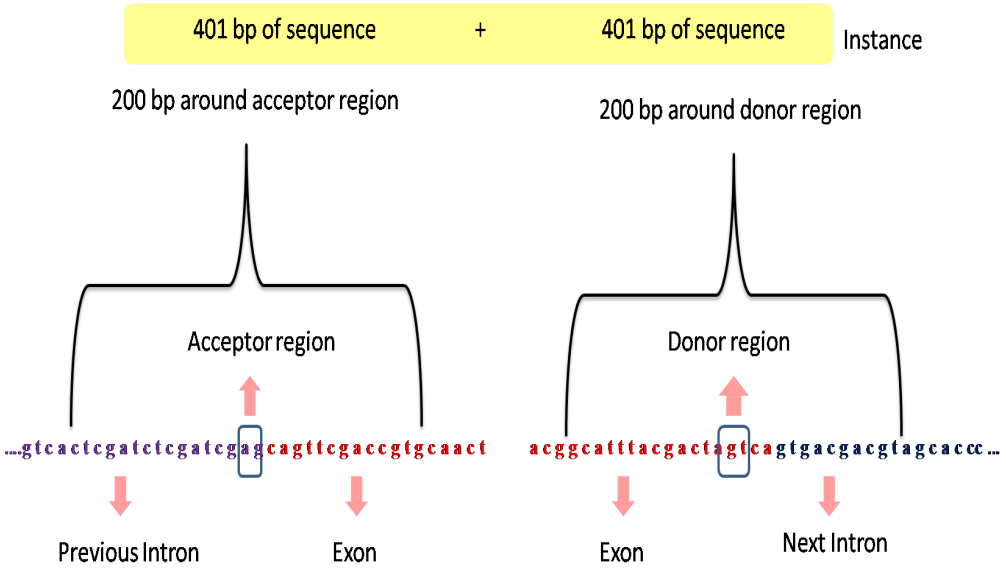


Figure 5.2: *Generation of an instance from the original sequence containing previous intron, exon and next intron.*

5.2 5-fold Cross Validation Setup

Once the data is represented as feature vectors, we split it for cross-validation purposes. We used a 5-fold cross validation procedure to run all the experiments. This section describes the details of the 5-fold cross validation scheme.

We divide the whole data into 5 splits. We run each algorithm 5 times. Each time, we consider one of the 5 splits for testing and the remaining 4 for training. By the end, all the splits will be used as test data once. To report the performance, we take the average of the predictions obtained in each of the 5 runs. Figure 5.3 shows the way we split the total

data into training and test data. The cross-validation procedure is used to reduce the bias in data sampling.

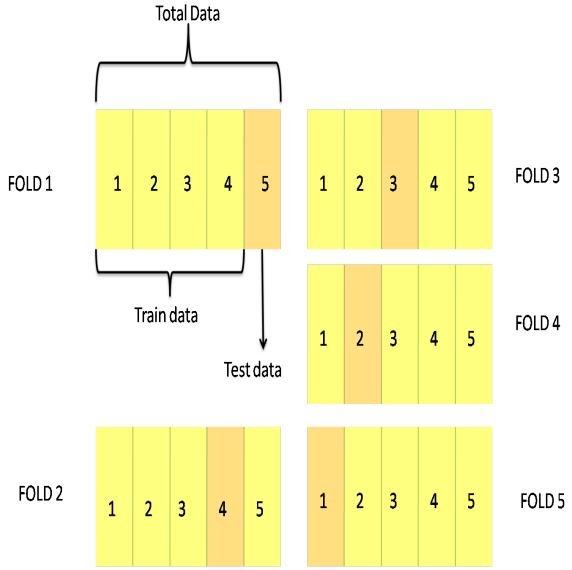


Figure 5.3: *5-fold cross validation*

Once we have all the 5 splits, as mentioned before, we consider one fold for testing and the remaining folds for training. However, as we implement semi-supervised/transductive approaches, we also split the training data into two subsets. We treat one of the two subsets as labeled, and the other one as unlabeled. To ensure the data assumption of these approaches, we make sure that the amount of labeled data is smaller than the amount of unlabeled data. We use various combinations of labeled and unlabeled percentage splits. Figure 5.4 shows an example of labeled and unlabeled percentage splits. Once we have test data, training labeled data and unlabeled data, we use these sets to run experiments using the co-training as well as graph-based transductive learning algorithms.

5-fold Cross Validation for Co-training:

The first and basic principle for using the co-training algorithm is to have data represented in two different views. For this purpose, we used various combination of feature sets. Feature combinations such as IRS vs ESE, IRS vs Length are used for the two corresponding views. After representing the whole data using two views, we apply the same 5-fold cross

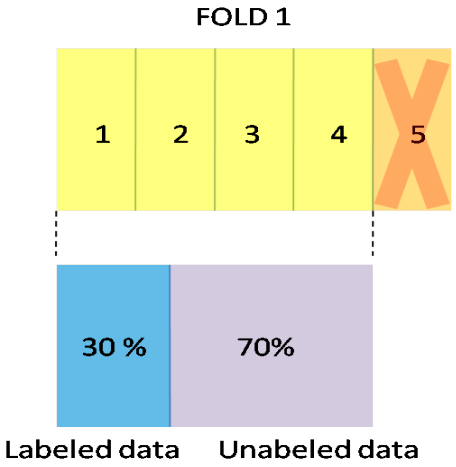


Figure 5.4: *The training set of each split is further divided into labeled and unlabeled subsets.*

validation scheme on both views. In other words, we work with the same splits in each view. Figure 5.5 shows the flow of information from original data to the splits which can be used as input to the co-training algorithm, shown schematically in Figure 2.8. We use the labeled and unlabeled data for the training phase to learn a model using co-training. We use the model generated in the training phase to predict the test data. This process is repeated for all 5 folds. The final accuracy is the average of all 5 accuracies corresponding to the 5-folds.

5-fold Cross Validation for Graph-based Transductive Approach:

For the graph-based transductive learning approach, we use the same splits of data that we use for co-training. However, for graph-based approach, we combined the features of two views into a single view. The main difference between the two approaches is given by the fact that co-training does not generate a model from the knowledge of both labeled and unlabeled. Instead, it only labels all the unlabeled data. We computed the accuracy of graph-based transductive approach by considering the accuracy with respect to the unlabeled data. Figure 5.6 shows the implementation of graph-based transductive learning with 5-fold cross validation scheme.

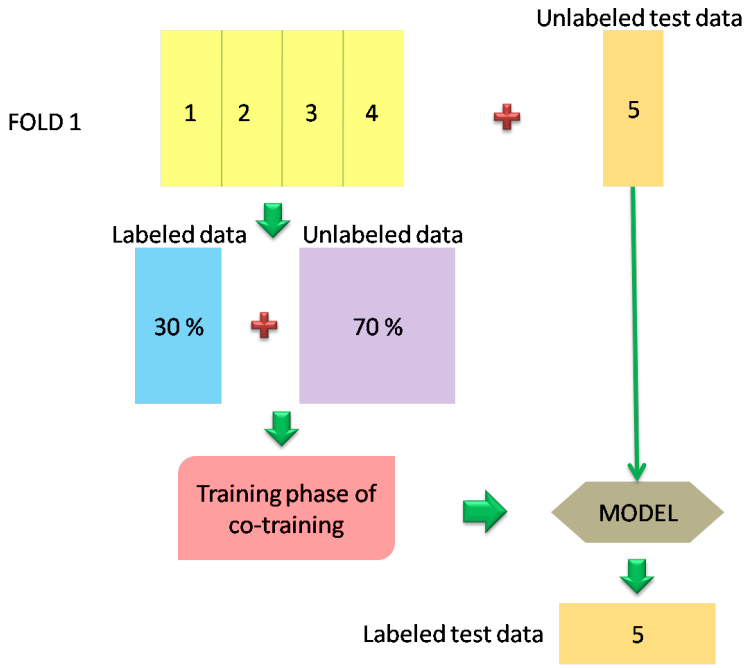


Figure 5.5: *Co-training algorithm with 5-fold cross validation*

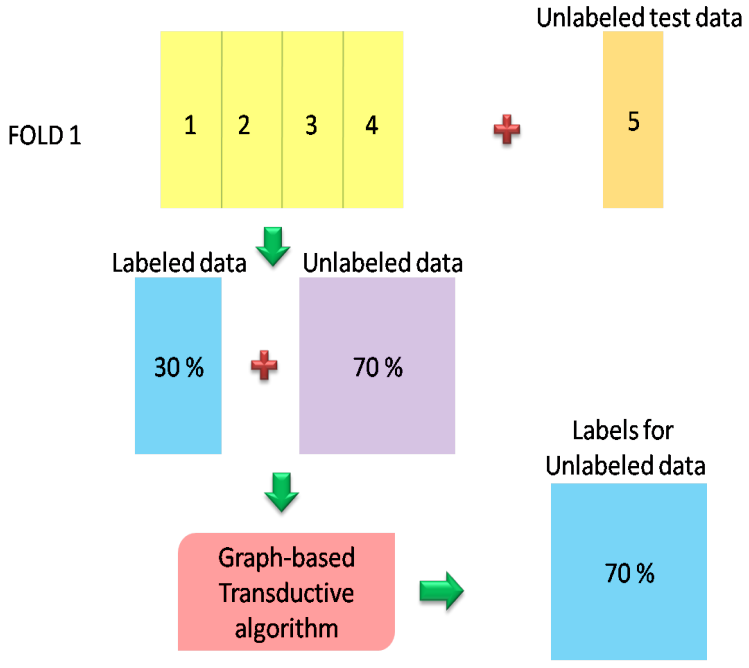


Figure 5.6: *Graph based transductive algorithm with 5-fold cross validation*

5.3 Research Questions

The questions that we address in this work are the following.

- *How does co-training perform on biological data?* Co-training is a semi-supervised learning algorithm, which can make use of both labeled and unlabeled data. It works well when it uses two distinct feature representations which are independent and sufficient to learn the classifier. Given our problem, we believe that the set of features available can be divided into several combinations of views, which might be independent given the class label, and sufficient. For example, exonic splicing enhancers can be used as one view and intronic regulatory motifs as another view. These motifs are presumably independent given the class, as one corresponds to the exonic region and the other corresponds to the intronic region. Our goal is to see if co-training gives acceptable results when compared to supervised approaches for the problem of alternative splicing prediction. Thus, we will compare the co-training results with results obtained in a supervised learning framework, when all training data is used as labeled data.

- *How does the semi-supervised learning approach perform when compared to supervised learning algorithms which have access to a small amount of labeled data?* As it is costly to label large amounts of data, we will compare the co-training results with results obtained in a supervised learning framework, which makes use of a small amount of labeled data.

- *How do the results vary with the amounts of labeled and unlabeled data?* Given the data, we usually divide it into labeled data, unlabeled data and test data for semi-supervised learning. Intuitively, for the same amount of labeled data, increasing the amount of unlabeled data should result in an increase in performance. Similarly, for the same amount of unlabeled data, increasing the amount of labeled data should result in better performance. To study this behavior, in our experiments we vary the

amounts of labeled and unlabeled data, respectively, and observe the variation in the performance.

- *What views of features give the best results?* Given that we are using the ESE motifs, IRS motifs and Length features, there are several combination of views that we can use. Therefore, we are interested in understanding what combinations of features give the best performance in a semi-supervised framework. That involves learning classifiers from various views and comparing their performance to identify the most predictive views.

- *What are the best base classifiers for co-training?* Co-training uses two base classifiers to learn models from the two available views. Given the types of features that are used in this work, NB, NBM and SVM can be used as base classifiers, among others. We investigate NB and NBM, as they are simple probabilistic algorithms that work very well on sequence data [McCallum and Nigam, 1998]. Furthermore, we investigate SVM as a base classifier, as it is known to be one of the best classifiers, especially for the problem of predicting alternative splicing, which is discussed in Section 3.4. We perform experiments with these three classifiers as base classifiers, and compare the results to identify the best combinations of classifiers for the views that we consider.

- *How does the graph based transductive learning approach perform on biological data?* The graph based transductive learning approach has been successful in various domains and for various problems as discussed in Section 3.2 . Our goal is to see if this algorithm gives acceptable results for a biological problem. The working of graph based transductive approach involves the propagation of labels across the graph, based on similarity weights. It uses the basic principle of *cluster assumption*, which states that that two instance belonging to the same cluster should share a common label. For the problem of predicting alternative splicing, instances are sequences of nucleotides around the acceptor and donor regions. The feature vector representations are based

on lengths features and the counts of various motifs that occur in the sequences of an instance (IRS and ESE motifs). We use a Gaussian kernel to compute similarity between feature representations of instances.

As in the case of co-training, for the graph based approaches explored, we are also interested in studying the variation of the performance with the amounts of labeled and unlabeled data, respectively. Furthermore, we investigate several types of feature representations (based on the available features) to identify subsets of features that result in best performance.

5.4 Experiments

5.4.1 Co-Training Evaluation

To address some of the research questions raised above, we consider the combinations of views shown in Table 5.1 for use with the co-training algorithm.

Table 5.1: *Combinations of views used with co-training*

Combination #	View 1	View 2
1	IRS	LENGTH
2	ESE	LENGTH
3	IRS + ESE	LENGTH
4	ESE	IRS

The experiments performed for co-training, including the algorithms that are investigated for each view combination, are shown in Table 5.2.

Table 5.2: *Experiments performed using co-training algorithm.*

Experiment#	Combination# (Table 5.1)	Alg. for View 1	Alg. for view 2
Experiment 1	1, 2, 3	NBM	NB
Experiment 2	1, 2, 3	NBM	SVM
Experiment 3	1, 2, 3	SVM	SVM
Experiment 4	4	NBM	NBM
Experiment 5	4	SVM	SVM

As mentioned above, we evaluated our co-training implementation using 5-fold cross validation. The original dataset is split into training and testing subsets, where 1 out of the 5 folds is used for testing and the remaining 4 folds for training. The training data is further divided into unlabeled and labeled data with various percentage splits, as shown below:

- The amount of labeled data is varied from 5% to 30%.
- The amount of unlabeled data is varied from 15% to 95%.

In our experiments, we also varied the sample size, which is the number of unlabeled instances that we start with, and the number of iterations for which the algorithm is run:

- Sample size is varied from 25 to 250.
- Number of iterations is varied from 25 to 125.

To compare the co-training semi-supervised approach with supervised approaches, we compute *lower* and *upper bounds* as follows: For a given split, and particular percentages of labeled to unlabeled data, we calculate the lower bound by training a supervised classifier on the labeled data and then evaluating this classifier on the test data. Similarly, we calculate the upper bound by training a supervised classifier on both labeled and unlabeled data (which we consider as labeled in this case) and evaluating it on the test data.

5.4.2 Graph-based Approach Evaluation

We also evaluate the graph based transductive learning approach on the combinations described in Table 5.1. In this case, we combine both views into a single representation to compute the similarity matrix as mentioned in Section 2.2.3. We use a Gaussian kernel for computing the matrix. The following values are used for the kernel parameter σ: 0.1, 0.25, 0.5, 0.75, 1.0, 10.0, 50.0 and 100.0, to study how the results vary with this parameter. We also implement the two variants described in [Zhou et al., 2004] and compare them with the original graph-based approach. Given the transductive framework explored, we will not

use the test data separately, although the training labeled and unlabeled data sets as the same used in co-training. Here, we use the labeled and unlabeled data to predict labels for the unlabeled data. The AUC values that we report will be the average of the results of predicting the unlabeled data.

Chapter 6

Results

In this chapter, we discuss the results of the experiments listed in Section 5.4. Section 6.1 presents the co-training results. We first discuss the results when various features and base classifiers are used, in Section 6.1.1. In Section 6.1.2, we discuss the results of co-training when the percentage of labeled data is varied, while the amount of unlabeled data is fixed. Section 6.1.3 deals with the performance of co-training when the percentage of unlabeled data is varied and the amount of labeled data is fixed. Section 6.2 presents the results obtained by performing experiments using the graph-based transductive approach.

6.1 Co-training Results

6.1.1 Study on Features and Base Classifiers

In this section, we report the results of various experiments performed with co-training on predicting alternative splicing events in genes. The goal is to identify the best views to be used with co-training and the best base classifiers corresponding to those views. Table 6.1 shows the results for all the experiments listed in Table 5.2, when 5% of the data is used as labeled data and 95% is used as unlabeled data. The results reported for these experiments are the best AUC values obtained for a particular combination of views and base classifiers.

Table 6.1: *AUC values for co-training applied on various combinations of views (first column) and base classifiers (second column). Column 3 shows the lower bound for view 1 (LBV1), while column 4 shows the upper bound for view 1 (UBV1). Similarly, the lower bound for view 2 (LBV2) and upper bound for view 2 (UBV2) are shown in columns 5 and 6, respectively. Columns 7 and 8 show the lower bound for combined view (LB) and upper bound for combined view (UB). The co-training results (CT) are shown in the last column. The best co-training results for each combination of views and base classifiers are highlighted.*

Features#	Classifers	LBV1	UBV1	LBV2	UBV2	LB	UB	CT
ESE vs LG	NBM + NB	0.709	0.771	0.803	0.826	-	-	**0.826**
ESE vs LG	NBM + SVM	0.709	0.771	0.795	0.835	-	-	0.747
ESE vs LG	SVM + SVM	0.681	0.772	0.795	0.835	0.821	0.862	0.732
IRS vs LG	NBM + NB	0.806	0.907	0.803	0.826	-	-	0.848
IRS vs LG	NBM + SVM	0.806	0.907	0.795	0.835	-	-	0.901
IRS vs LG	SVM + SVM	0.795	0.895	0.795	0.835	0.858	0.907	**0.916**
ESE+IRS vs LG	NBM + NB	0.848	0.93	0.803	0.826	-	-	0.874
ESE+IRS vs LG	NBM + SVM	0.848	0.93	0.795	0.835	-	-	0.892
ESE+IRS vs LG	SVM + SVM	0.794	0.921	0.795	0.835	0.825	0.916	**0.907**
ESE vs IRS	NBM + NBM	0.709	0.771	0.806	0.907	0.848	0.93	**0.889**
ESE vs IRS	SVM + SVM	0.681	0.772	0.795	0.895	0.794	0.921	0.862

By analyzing the results in Table 6.1, we can make the following observations:

- The best co-training performance is obtained when using the views IRS vs LENGTH, and SVM as a base classifier for both views. Specifically, the AUC value is 0.916 in this case. The next best value is 0.907 and is obtained for the views IRS+ESE vs LENGTH, also with SVM as a base classifier for both views. This shows that on our dataset, co-training gives the best results when the views used correspond to motifs and length features, respectively, and SVM is used as a base classifier. Intuitively, these views should be independent given the class variable. Also, the upper bound results show that each view in itself is predictive of the class, given enough data. Thus, co-training assumptions seem to be satisfied for these combinations of views and base classifiers, and the results support this claim.

- When the amount of labeled data is small, it is interesting to note that the combination

of views IRS vs LENGTH gives better results than the combination of views IRS+ESE vs LENGTH. Generally, we would expect that adding more information (ESE) would increase the classifier performance. Instead, a decrease in the AUC value is observed. This result leads us to believe that co-training with SVM as a base classifier cannot handle well ESE features, when small amounts of labeled data are available. However, supervised SVM classifiers can make use of the ESE features, as the upper bound obtained using all the data as labeled data is better for IRS+ESE+LENGTH than it is for IRS+LENGTH.

- The following observations suggest that ESE motifs alone are not predictive when used with SVM. First, we can see that the combination of views ESE vs LENGTH gives the best result when naive Bayes classifiers (NBM and NB, respectively) are used as base classifiers. In this case, the co-training result is similar or better than the upper bounds on the corresponding views. In the other cases, when SVM is used on one or both views, the co-training result is worst than the lower bounds. Furthermore, the combination of views ESE vs IRS, with NBM as base classifier for both views, gives better results than the experiment where SVM is used as base classifier. Therefore, we can conclude that naive Bayes can better capture the information in the ESE motifs than the SVM classifier.

- Another interesting fact to notice is that the combinations of views IRS vs LENGTH and IRS+ESE vs LENGTH, with SVM as base classifier on both views, result in AUC values which are greater than all upper bounds (i.e., UBV1, UBV2 and UB). One possible explanation for this behavior can be that co-training might be able to avoid some noise in the data (mislabeled instances). Co-training learns two classifiers from two different views. If the same instance is classified with high confidence by both classifiers, but the classification labels are different (in other words, the two predictions are conflicting), then that instance is skipped as opposed to being added to the training set of both classifiers. It can be that the instance with conflicting labels is mislabeled

in the original training set. Thus, co-training can ignore possible mislabeling of the data, while supervised learning will use the mislabeled data and can result in a biased classifier. Another possible explanation can be that, the ensemble learning is more beneficial than learning directly from the combined views. In other words, learning separately from two different views and interchanging the best knowledge can be better than learning directly from the combined features.

- We also observe that the best result for the IRS vs ESE combination of views is obtained when NBM is applied on both views. SVM gives worst results in this case. As discussed above, one possible explanation for this is that SVM does not capture well the information in the ESE features. Furthermore, both ESE and IRS motifs are represented using counts and the NBM is known to work very well for count representations (e.g., the count representation can give better results when used with NBM than when used with SVM for text classification problems). As opposed to this, when LENGTH features are used, the results are generally better for SVM.

- At last, the results show that using the same base classifier for both views gives better results than when different classifiers are used for the two views (e.g. NBM and SVM). In other words, classifiers are able to help each other better if they are similar in nature and capture the same type of information.

As a general conclusion of the results, we can claim that IRS vs LENGTH and IRS+ESE vs LENGTH combinations of views, used with SVM, give the best results for our dataset. When NBM is used as a base classifier, the combination IRS vs ESE gives the best results. Therefore, in the next subsections, we will focus on these combinations.

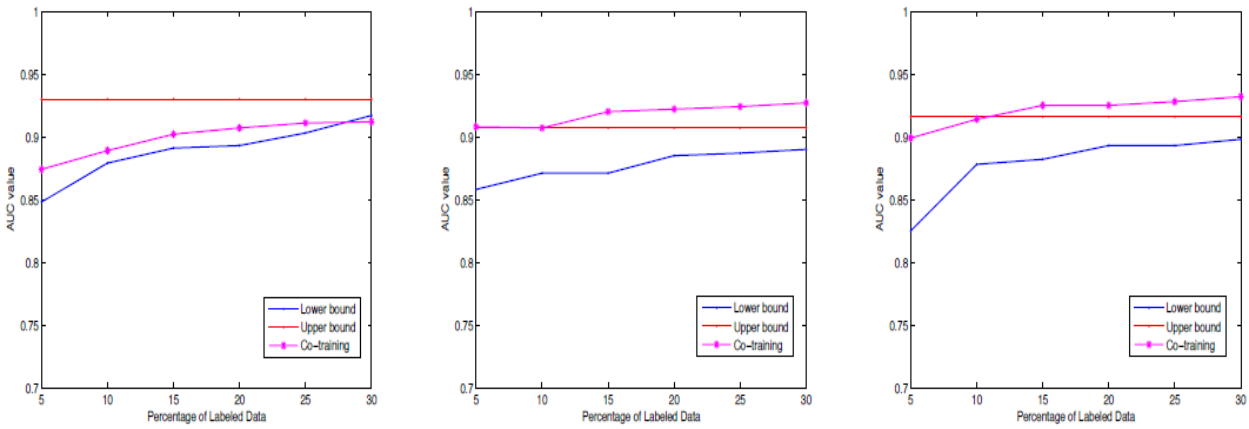
6.1.2 Varying the Amount of Labeled Data

Focusing on the three combinations of views and classifiers mentioned above, in this section we evaluate the performance of co-training when varying the amount of labeled data, while

the amount of unlabeled data is fixed. We first present the results of the experiments performed with NBM as base classifier for both views (IRS vs ESE), followed by results performed with SVM as base classifier for both views IRS vs LENGTH and IRS+ESE vs LENGTH.

NBM results

In this set of experiments, we vary the amount of labeled data from 5% to 30%, while keeping the amount of unlabeled data fixed to 70%. The results are shown in Figure 6.1 (a). As can be seen, increasing the amount of labeled data results in better performance for both co-training and supervised learning from labeled data only (lower bound). However, the difference between the co-training and the lower bound decreases as the amount of labeled data increases, and the lower bound becomes greater than the co-training as the amount of labeled data reaches 30% (although neither of them gets very close to the upper bound). This result shows that co-training works well when the amount of labeled data is small.



(a) ESE vs IRS, vary labeled (b) IRS vs LENGTH, vary labeled (c) IRS + ESE vs LENGTH, vary labeled

Figure 6.1: *Co-training results (AUC values) when varying the amount of labeled data from 5% to 30%, while the amount of unlabeled data is fixed to 70%. (a) IRS vs ESE are used as views, and NBM is used as base classifier for both views; (b) IRS vs LG are used as views, and SVM is used as base classifier; (c) IRS + ESE vs LG are used as views and SVM is used as base classifier.*

SVM results

Similar to the NBM case, in this set of experiments, we vary the amount of labeled data from 5% to 30%, while keeping the amount of unlabeled data fixed to 70%. However, in this case, SVM is used as base classifier for both views, and we experiment with the IRS vs LENGTH and IRS+ESE vs LENGTH combinations of views. The results for the combination IRS vs LENGTH are shown in Figure 6.1 (b). Similar to the NBM results, we notice that the performance of co-training increases with the amount of labeled data. However, the results are much better as compared to the lower bound, and, in fact, they are even better than the supervised upper bound. The difference between co-training and the upper bound increases with the amount of labeled data, reinforcing our observation that co-training is able to ignore examples that might be mislabeled (or inconsistent) in the training data. The results for the combination IRS+ESE vs LENGTH are shown in Figure 6.1 (c) and suggest a similar trend as observed for the IRS vs LENGTH combination. However, the rate at which co-training increases is greater for smaller amounts of labeled data (10% to 15%) as compared to larger amounts of labeled data (25% to 30%).

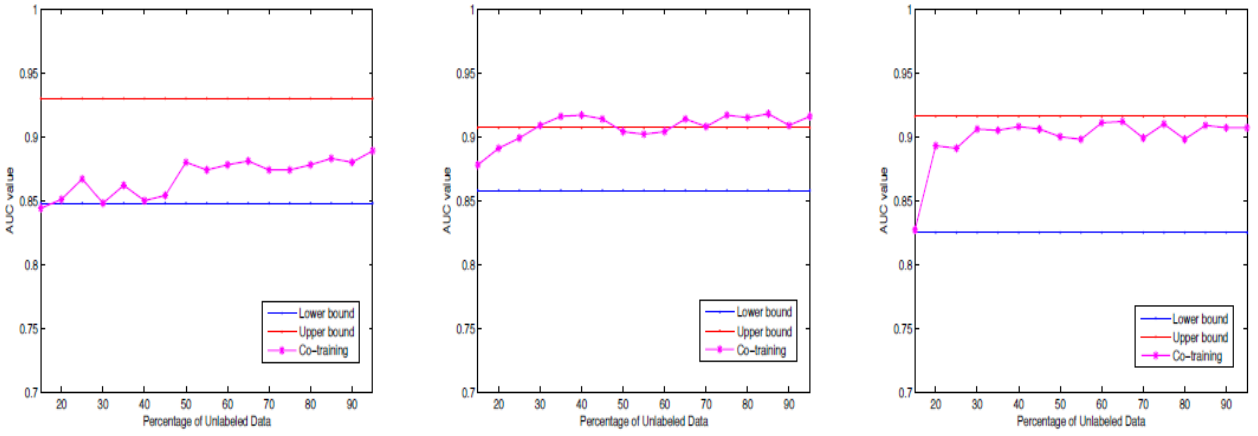
Based on these observations, we can conclude that co-training is effectively using the unlabeled data and gives better results than the supervised learning that uses only the labeled data. While the performance increases with more labeled data, the benefit over the lower bound gets smaller. Thus, co-training should ideally be used when the amount of labeled data is very small.

6.1.3 Varying the Amount of Unlabeled Data

For the combination of features and classifiers described in Section 6.1.2, we are also studying the variation of the co-training performance with the amount of unlabeled data, for a fixed amount of labeled data. As before, we first present the results of the experiments performed with NBM as base classifier for views IRS vs ESE, followed by results performed with SVM as base classifier for views IRS vs LENGTH and IRS+ESE vs LENGTH, respectively.

NBM results

For this set of experiments, we fixed the amount of labeled data to 5% and studied the performance of co-training when we vary the amount of unlabeled data from 15% to 95%. The results for the IRS vs ESE combination of views and NBM as a base classifier are shown in Figure 6.2 (a). While co-training gives better results than the lower bound (showing that unlabeled data helps), the performance increases with the amount of unlabeled data up to a point; adding unlabeled data beyond that point does not help much.



(a) ESE vs IRS, vary unlabeled (b) IRS vs LENGTH, vary unlabeled (c) IRS + ESE vs LENGTH, vary unlabeled

Figure 6.2: *Co-training results (AUC values) when varying the amount of unlabeled data from 15% to 95%, while the amount of labeled data is fixed to 5%. (a) IRS vs ESE are used as views, and NBM is used as base classifier for both views; (b) IRS vs LG are used as views, and SVM is used as base classifier; (c) IRS + ESE vs LG are used as views and SVM is used as base classifier.*

SVM results

Similar to the NBM experiments, for this set of experiments, we fixed the amount of labeled data to 5% and studied the performance of co-training when we vary the amount of unlabeled data from 15% to 95%. Here, SVM is used as a base classifier. The results for the IRS vs LENGTH combination of views are presented in Figure 6.2 (b) and the results for IRS+ESE vs LENGTH are presented in 6.2 (c). As in the case of NBM, the performance of co-training is better than the lower bound, which shows that the unlabeled data helps. When

we vary the amount of unlabeled data from 15%, initially the performance of co-training increases up to a point. However, adding more unlabeled data beyond that point does not result in a consistent increase in the performance of co-training, when the IRS vs LENGTH combination is used. However, for the IRS+ESE vs LENGTH combination of views, no consistent performance increase can be observed across the sequence of percentages from 15% to 95% for unlabeled data. One observation that can be made is that the results of the experiments when SVM is used as a base classifier are better than those obtained with NBM, in the sense that the co-training results are closer to the upper bound for SVM.

Based on the above observations, we can conclude that the unlabeled data results in better classifiers as compared to those learned from labeled data only (results better than the lower bound and close or even better than the upper bound). However, adding unlabeled data beyond a certain percentage does not result in significant improvements, as can be seen in Figure 6.2. To find good combinations of features, in Figure 6.3 we plot the co-training results for all combinations described in Table 5.1, when SVM is used as a base classifier for both views, and the amount of unlabeled data is varied from 50% to 95%. As can be seen from the figure, the results corresponding to IRS vs LENGTH and IRS+ESE vs LENGTH are comparable, while the results for IRS vs ESE and ESE vs LENGTH are much worst. However, the results of ESE vs LENGTH are the worst, as has also been seen from Table 6.1.

6.2 Graph-Based Approach Results

In this section, we report the results of the graph-based transductive approach [Zhou et al., 2004] for predicting alternative splicing events in genes. We used a Gaussian kernel to compute the weight matrix, which captures the similarity between every two instances. We used the following values for the parameter σ in the Gaussian kernel: 0.1, 0.5, 0.75, 1.0, 10, 50 and 100. However, the value 0.1 and values greater than 1.0 are not giving satisfactory results, as the elements in the matrix tend either to infinity or to zero upon transforming the

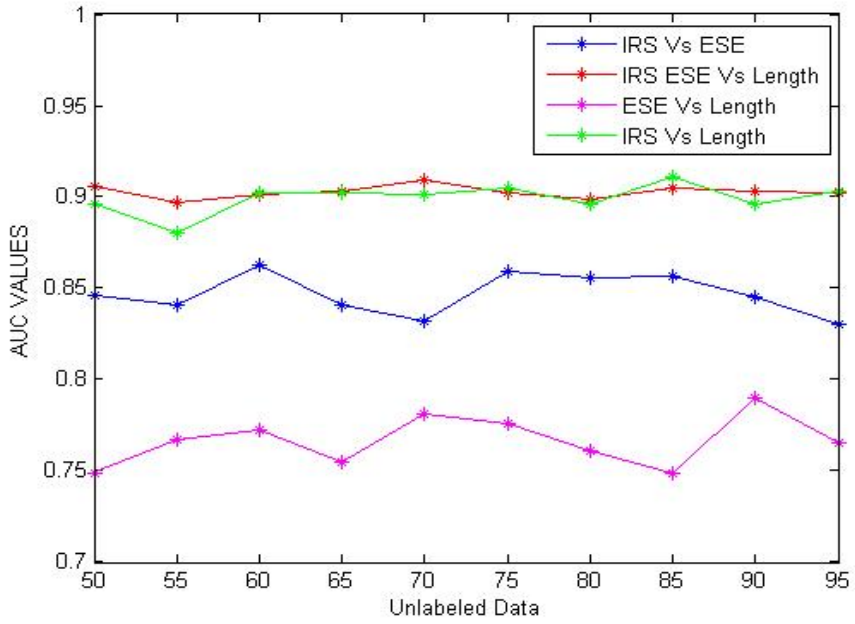


Figure 6.3: *Co-training results for various combination of features, when SVM is used as base classifier and the amount of unlabeled data is varied from 50% to 95% (while the amount of labeled data is fixed to 5%).*

matrices as described in [Zhou et al., 2004]. Therefore, the results corresponding to those values will not be shown. We implemented the main converging function along with its two variants introduced in [Zhou et al., 2004]. The results show that the variants perform better than the main converging function. In particular, the first variant gives the best results. Table 6.2 shows the results of this variant when the algorithm used 30% of the data as labeled data and 70% of the data as unlabeled data.

Table 6.2: *AUC values for the graph-based transductive approach applied to the alternative splicing data set with 30% of data as labeled data and 70% of data as unlabeled data. Column 1 shows the "sigma" value used. The subsequent columns show the combination of features used to represent instances. Variant 1 is used for training.*

Sigma#	IRS+ESE	IRS+LG	IRS+ESE+LG	ESE+LG
0.5	0.793	0.823	0.837	0.774
0.75	0.799	0.831	0.84	0.783
1	0.814	0.849	0.849	0.804

As can be seen in the Table, using all the features IRS+ESE+LG gives the best performance results for the graph-based algorithm. The IRS+LG combination follows closely, while the results obtained with the other variants are significantly worst. Furthermore, we can see that the performance increases with the value of sigma - the best results are obtained for sigma = 1.

Figure 6.4 (a) shows the performance of the graph-based transductive approach for various combinations of feature sets, when $\sigma = 1$ and the amount of labeled data varies from 5% to 30%. While all feature combinations result in increased performance when more labeled data is added, we can observe that the IRS + LG feature combination is consistently performing better than all other combinations. The IRS+ESE+LG feature combination follows closely. Given that IRS+LG combination gives the best results, in the next sets of experiments we will used these features and sigma = 1.

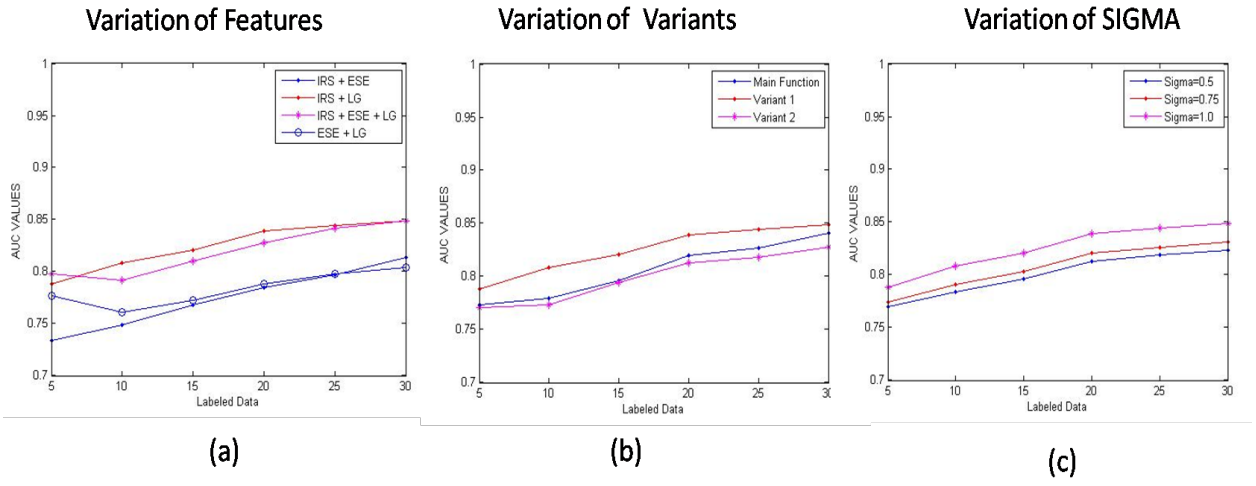


Figure 6.4: *Graph-based transductive learning results for various combinations of sigma values, functions and features. The amount of labeled data is varied form 5% to 30%, while the amount of unlabeled data is fixed to 70%. (a) AUC values vs amount of labeled data, when various feature combinations are used (sigma=1 and variant 1); (b) AUC values vs amount of labeled data, for various sigma values (IRS+LG and variant 1); (c) AUC values vs amount of labeled data, for the three converging functions used (sigma=1and IRS+LG).*

Figure 6.4 (b) shows the variation of the AUC values with the amount of labeled data, when the three converging functions (main function, variant 1 and variant 2) are used.

We used IRS + LG features and sigma = 1, in these experiments. As can be seen, the variant 1 consistently outperforms the other two functions for all the amounts of labeled data considered. The main function is better than variant 1 for smaller amounts of labeled data, while variant 2 is better for larger amounts of labeled data. Furthermore, as expected, the performance increases with the amount of labeled data for all variants.

Figure 6.4 (c) shows the variation of the AUC values with the amount of labeled data, when the parameter sigma of the Gaussian kernel is varied. We used IRS + LG features and variant 1, in these experiments. The graph compares three values of sigma: 0.5, 0.75, 1.0. We can observe that the performance of graph-based transductive learning is increasing with an increase in the parameter sigma. However, for sigma greater than 1.0, the performance starts degrading (results not shown). As before, in this case also, the results improve when the amount of labeled data increases.

Given that we have observed that the best results are obtained using sigma=1, IRS+LG features and variant 1, in Figure 6.5, we use these parameters to show the variation of the performance with both the the amount of labeled data and the amount of unlabeled data. From Figure 6.5 (a), we can observe that there is a continuous increase in the performance of the graph-based transductive approach with the amount of labeled data. This is similar to what we have observed for co-training. However, the graph-based approach gives worst results than co-training. Figure 6.5 (b) shows the variation of graph-based approach with the amount of unlabeled data. As has been seen, there isn't much variation in performance when increasing the amount of unlabeled data beyond a certain percentage. A similar observation was made for co-training.

Based on the above observations, we conclude that the performance of the graph-based transductive approach improves with the amount of labeled data, but does not change much with the amount of unlabeled data (when at least 50% unlabeled data is used). The IRS+LG feature combination is the most predictive. Furthermore, variant 1 give the best AUC values, among the three variants explored. However, overall, co-training performs better than the

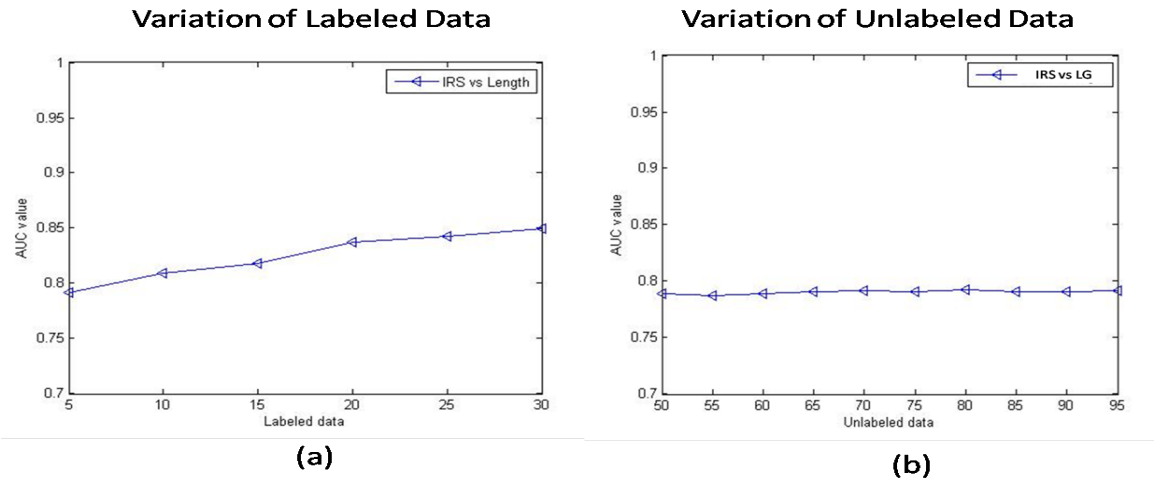


Figure 6.5: *Graph based results when the labeled and unlabeled data percentages are varied. Varian 1 is used in these experiments, with the IRS+LG feature combination and sigma =1: (a) AUC values vs amount of labeled data; (b) AUC values vs amount of labeled data.*

graph-based transductive approach.

Chapter 7

Conclusions and Future Work

In this chapter, we first draw some conclusions for the work presented in this thesis and discuss limitations of our co-training and graph-based approaches, in Section 7.1. Some improvements and future directions for this work are proposed in Section 7.2.

7.1 Conclusions

7.1.1 Co-Training

The following main conclusions can be drawn for co-training, when applied to predict alternatively spliced exons:

- Experimental results have shown that co-training performs well on biological data when the sets of features used as views are sufficient and independent (e.g., IRS vs LENGTH and IRS+ESE vs LENGTH).

- As expected, the performance of co-training increases with the amount of labeled data. When we varied the amount of unlabeled data, the performance of co-training increases initially but does not show a consistent pattern for larger percentages of unlabeled data. This suggests that, while the unlabeled data can help, it does not help much beyond a certain point.

- We can also conclude that SVM works best as a base classifier when motifs versus

length features are used as views. However, NBM gives the best results for the combination of IRS vs ESE (although the corresponding AUC value is smaller than the best value obtained in SVM).Using different classifiers for the two views was found to be ineffective.

Based on the above observations, co-training is found to be effective in predicting alternative splicing events in genes. Unlabeled data plays a crucial role in improving the performance. This suggests that co-training can also be effective for various other biological problems where we have large amounts of unlabeled data.

7.1.2 Graph-Based Transductive Learning Approach

The following main conclusions can be drawn for the graph-based approach:

- Experimental results have shown that the graph-based transductive learning approach can effectively use the knowledge of labeled and unlabeled data to produce accurate labels for the unlabeled data. The best AUC value obtained on the unlabeled data is 0.85. We should note that this result cannot be directly compared with co-training, which was evaluated on the test data. While there might be room for improvement (given the highest AUC on the test data, using co-training), the current results are nevertheless very promising.

- Similar to co-training, the performance of the graph based approach increases with the amount of labeled data. The unlabeled data helps, but adding unlabeled data beyond a point does not increase the performance anymore.

- Consistent with the co-training findings, the feature combination IRS+LG gives the best results also for the graph-based approach, followed closely by the IRS+ESE+LG combination.

- We have evaluated one main function and two variants in this work. The results have shown that the second variant is the most appropriate for our dataset. For all variants,

the similarity matrix has been calculated using a Gaussian kernel. The best results have been obtained for sigma = 1.

Although our approach of using semi-supervised and transductive learning for alternative splicing predictions proved to be effective, it has some limitations as well. These algorithms work well only with certain feature set combinations. The best combinations and some other parameters of the algorithms could be difficult to identify. An automated validation schema would be useful. A limitation specific to co-training is that it performs well only if the two views are sufficient and independent. Failure in identifying views that satisfy the two assumptions may lead to sub-optimal performance. A limitation specific to the graph-based transductive learning algorithm is that this approach is not very efficient, generally, the required matrix operations are time consuming.

7.2 Future Work

As part of the future work, we would like to test our approach on different datasets from different organisms. We would also like to collect more unlabeled data and run experiments where all the available labeled data is used as training, to see if the results can be improved further. Another idea for future work is to construct a validation scheme which can help identify the best set of feature and also for tuning various other parameters (such as number of iterations, sigma). We have used the graph-based approach only in a transductive setting. In future work, we would like to explore ways in which the labeled data produced by the graph-based approach could be used to produce labels for new test data. In other words, we would like to explore ways in which the transductive learning can be transformed into inductive learning. We would also like to investigate different kernels for SVM (when used in co-training) and for calculating similarity (in the graph-based approach). A Gaussian kernel was used for both in the current work, but we would like to explore biological kernels such as the weighted degree kernel in the future. At last, it would be interesting to explore an ensemble co-training with multiple views of features.

Bibliography

D. L. Black. Mechanisms of alternative pre-messenger RNA splicing. *Annual Review of Biochemistry*, 72:291–336, 2003.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 1998.

P. Bonizzoni, R. Dondi, R. Rizzi, and G. Pesole. Aspic: a novel method to predict alternative splicing. *BMC Bioinformatics*, 6, 2008.

M. Collins and Y. Singer. Unsupervised models for named entity classification. In *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110, 1999.

C. Corinna and V. Vladimir. Support-vector networks. *Machine Learning*, 20:273–297, 1995. ISSN 0885-6125. 10.1007/BF00994018.

W. Dai, G. Xue, Q. Yang, and Y. Yu. Transferring naive bayes classifiers for text classification. In *In Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 540–545, 2007.

G. Dror, R. Sorek, and R. Shamir. Accurate identification of alternatively spliced exons using support vector machine. *Bioinformatics*, 21:897–901, 2005.

W. Fei and Z. Changshui. Label propagation through linear neighborhoods. *IEEE Trans. on Knowl. and Data Eng.*, 20:55–67, January 2008. ISSN 1041-4347.

A. Goldberg and X. Zhu. Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In *Proceedings of the First Workshop*

on Graph Based Methods for Natural Language Processing, TextGraphs-1, pages 45–52, 2006.

R. Gupta and L. Ratinov. Text categorization with knowledge transfer from heterogeneous data sources. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 2*, pages 842–847, 2008.

L. Hongchao, L. Lan, S. Seiko, X. Yi, and L. J. Christopher. Predicting functional alternative splicing by measuring rna selection pressure from multigenome alignments. *PLoS Comput Biol*, 5, 12 2009.

T. Jebara and S. Chang. Graph construction and bmatching for semi-supervised learning. In *In International Conference on Machine Learning*, 2009.

T. Joachims. Transductive inference for text classification using support vector machines. In *ICML '99 Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209. Morgan Kaufmann, 1999.

L. Kall, J. Canterbury, J. Weston, W. Noble, and M. MacCoss. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nat Meth*, 4:923–925, 2007.

P. Larraaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inzai, J. Lozano, R. Armaanzas, G. Santaf, A. Prez, and V. Robles. Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7(1):86–112, 2006.

A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *In AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48. AAAI Press, 1998.

G. J. Mclachlan and T. Krishnan. The em algorithm and extensions. *Biometrics*, 65(3): 1000–1000, 2009. ISSN 1541-0420.

M. T. Mitchell. *Machine learning*. McGraw-Hill Companies Inc., international edition, 1997.

K. Nigam and G. Rayid. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, CIKM '00, pages 86–93, 2000.

K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Mach. Learn.*, 39:103–134, May 2000. ISSN 0885-6125.

Q. Pan, O. Sha, LJ. Lee, B. Frey, and B. Blencowe. Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nature Genetics*, 40:1413–1415, 2008.

G. Ratsch, S. Sonnenburg, and B. Scholkopf. Rase: recognition of alternatively spliced exons in c.elegans. In *ISMB 2005 Proceedings. Thirteenth International Conference on Intelligent Systems for Molecular Biology*, volume 21, pages 369–377, 2005.

K. Svetlana and S. Matwin. Email classification with co-training. In *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*, CASCON '01, pages 8–. IBM Press, 2001.

J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. Noble. Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21:3241–3247, 2005.

J. Weston, R. Kuang, C. Leslie, and W. Noble. Protein ranking by semi-supervised network propagation. *BMC Bioinformatics*, 7(Suppl 1):S10, 2006.

J. Xia, D. Caragea, and S. J. Brown. Prediction of alternatively spliced exons using support vector machines. *International Journal on Data Mining and Bioinformatics (IJDMB)*, 4: 411–430, 2010.

Z. Xiaojin. Semi-supervised learning literature survey. In *Technical Conference*, 2006.

Q. Xu, D. Hu, H. Xue, W. Yu, and Q. Yang. Semi-supervised protein subcellular localization. *BMC Bioinformatics*, 10(Suppl 1):S47, 2009.

D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schlkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328. MIT Press, 2004.